Cancer Prediction Model using Dimensionality Reduction techniques and RNN-CNN Classifier

Anuhya Shashi
221IT010
Dept. of Information Technology
NITK Surathkal
Mangalore, India
anuhya.shashi23@gmail.com

Prajna B Shettigar
221IT051
Dept. of Information Technology
NITK Surathkal
Mangalore, India
prajnabshettigar@gmail.com

Sonali Kannojiya
221IT065
Dept. of Information Technology
NITK Surathkal
Mangalore, India
sonali82990@gmail.com

***Abstract*—Cancer remains one of the deadliest illnesses, necessitating early detection and treatment for effective prevention. Medical laboratories employ various tests, including genetic-level detection methods, facilitated by advanced machine learning and deep learning techniques. This paper proposes a classification model leveraging Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) architectures for predicting multiple cancer types—Breast, Lung, Uterine, Kidney, Prostate, and Colon—based on gene expression data. The study incorporates metaheuristic feature selection methods such as Cuckoo Search, Genetic Algorithm, and Artificial Bee Colony, alongside feature extraction techniques including one-dimensional SSB and PCA.**

***Index Terms*—Cancer, Feature Selection, Gene Expression Data, Neural Networks, Deep Learning**

I. INTRODUCTION

Cancer is widely recognized as one of the most significant threats to human health, ranking among the leading causes of mortality globally—second only to infectious, cardiovascular, and cerebrovascular diseases [1–3]. A key factor in cancer classification lies in analyzing gene expression, which offers a snapshot of the biological system's current state by capturing active gene activity at the transcriptome level [4]. The transcriptome encompasses all RNA molecules, particularly messenger RNA (mRNA), that act as intermediaries in transferring genetic instructions from DNA to ribosomes for protein synthesis [5]. RNA sequencing (RNA-Seq) enables the assessment of gene transcription levels, allowing comparisons across tissue types and revealing genes associated with specific phenotypes. Such analyses are invaluable for identifying disease-related genes, improving diagnostic precision, and supporting therapeutic development. By contrasting gene expression profiles from healthy and cancerous tissues, researchers can uncover genetic mechanisms underlying various pathologies [6–9].

Gene Expression Data (GED) has proven particularly useful in enhancing cancer classification due to the granular biological insights it provides [49]. One of the primary computational challenges in working with GED is identifying differentially expressed genes that distinguish tumor cells from normal ones. The high dimensionality of gene features, combined with a relatively small number of samples, complicates traditional computational analysis. To address this, a wide range of supervised and unsupervised learning algorithms have been developed for GED-based cancer classification [9,10]. However, conventional machine learning (ML) models often struggle with feature selection and dimensionality reduction, limiting their effectiveness. These models tend to produce redundant classifiers that fail to adapt well to the complexity of GED, resulting in consistently suboptimal performance regardless of input variation [63].

In contrast, deep learning (DL) approaches have demonstrated significant improvements in handling GED's complexities. DL models are better suited to capture nonlinear relationships and extract high-level features from high-dimensional datasets, making them more effective in both feature selection and classification tasks [9,19].

A. *Machine Learning*

Machine Learning (ML), a vital subfield of Artificial Intelligence (AI), enables systems to learn patterns from data and make decisions with minimal human intervention. In the era of big data, where massive amounts of information are generated across sectors, ML has become essential for extracting meaningful insights. It supports tasks such as classification, regression, and clustering through a variety of algorithms.

Although ML systems can identify key features within large datasets, human expertise is still required to interpret results and guide decision-making [22]. The widespread accessibility of data has expanded ML applications in domains like healthcare, finance, education, and transportation [23], [24]. Common use cases include spam filtering, disease gene identification, customer segmentation, fraud detection, and predictive analytics [22].

B. *Deep Learning*

Deep Learning (DL), a specialized branch of ML, offers enhanced performance in complex tasks such as image recognition and natural language processing. DL models, often based on Artificial Neural Networks (ANNs), consist of mul-

Identify applicable funding agency here. If none, delete this.

tiple layers that learn hierarchical data representations through iterative training with backpropagation.

Expanding ANNs into Deep Neural Networks (DNNs) improves generalization and feature extraction, especially in high-dimensional data contexts. Frameworks like TensorFlow and PyTorch have facilitated DL model development [27]. Supervised DL models include Convolutional Neural Networks (CNNs), suited for spatial data [29], and Recurrent Neural Networks (RNNs), effective for sequence modeling [30]. Unsupervised models like Autoencoders (AEs) and Restricted Boltzmann Machines (RBMs) learn data representations without labeled input [31].

C. *Metaheuristics*

Feature selection and extraction are critical steps in the development of robust disease detection systems, particularly when dealing with high-dimensional medical datasets such as those derived from imaging or genomics. **Feature selection**involves identifying the most relevant subset of features from the original dataset that contribute significantly to classification performance, while **feature extraction** focuses on transforming the input data into a lower-dimensional space that preserves essential patterns. However, traditional feature selection techniques often struggle to cope with the complex, nonlinear, and high-dimensional nature of medical data. This is where **metaheuristic algorithms**—such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Cuckoo Search (CS)—play a pivotal role.

Metaheuristic algorithms are nature-inspired optimization techniques that mimic natural processes like evolution, swarm intelligence, or foraging behavior. These algorithms are particularly well-suited for navigating large search spaces to find near-optimal solutions with acceptable computational cost. In the context of disease detection, metaheuristics help to efficiently explore possible feature subsets or transformation strategies, often leading to improved classification accuracy, reduced computational complexity, and better generalization across datasets. By eliminating redundant or irrelevant features, they not only enhance the interpretability of diagnostic models but also reduce the risk of overfitting. Furthermore, when combined with powerful classifiers such as Support Vector Machines (SVM) or Convolutional Neural Networks (CNN), metaheuristic-based feature optimization can significantly boost early and accurate detection of diseases, including cancer, cardiovascular conditions, and neurological disorders. As such, their integration into computer-aided diagnosis systems is increasingly recognized as a best practice for building intelligent, scalable, and high-performance diagnostic tools.

D. *Motivation and Contribution*

With the rapid growth of high-throughput genomic technologies, vast gene expression datasets (GED) have become increasingly available for solving critical clinical challenges like cancer classification. However, GED presents several inherent difficulties such as high dimensionality with relatively few samples, noise, and variability. These challenges can hinder the performance and generalization of machine learning models, especially in biomedical domains.

Our work is motivated by two core questions:

- How can we effectively reduce the dimensionality of GED while preserving important biological patterns?
- What hybrid deep learning architectures can enhance classification accuracy for complex diseases like cancer?

To address these, we explore and evaluate a range of metaheuristic feature selection techniques—**Artificial Bee Colony**, **Cuckoo Search**, and **Genetic Algorithm**—to reduce irrelevant or redundant features. In parallel, we apply advanced feature extraction using **ConvNet-1D inspired by SSB(sandwich stacked bottleneck) feature EXtractor in base paper [13]**, and **Principal Component Analysis (PCA)** to explore deeper, compressed representations of the data.

For classification, we propose a **hybrid RNN-CNN model** that leverages the temporal modeling capabilities of RNNs and spatial feature detection of CNNs, designed to better handle the complex structure of gene expression data.

*a) **Our main contributions include**:*

- A comparative analysis of dimensionality reduction techniques, combining filter and wrapper-based feature selection methods with evolutionary algorithms (ABC, CS, GA).
- An enhanced feature extraction pipeline **ConvNet-1D** and **PCA** for robust pattern representation.
- A novel hybrid **RNN-CNN architecture** tailored for high-dimensional genomic data classification.
- Extensive evaluation of all approaches using multiple performance metrics against existing benchmark methods.

II. RELATED WORK

Early detection of cancer using computer-aided diagnosis (CAD) systems has garnered significant attention in recent years, owing to the high mortality associated with late-stage detection. Numerous approaches have been proposed that leverage advancements in medical image processing, machine learning, and bio-inspired optimization algorithms to enhance diagnostic accuracy, reduce false positives, and assist radiologists in clinical decision-making.

Initial segmentation methods primarily relied on traditional thresholding techniques due to their simplicity and computational efficiency. Armato et al. [1] employed gray-level thresholding followed by a rolling ball algorithm to extract juxta-pleural nodules from CT images. While effective for distinct nodules, this method was less robust for complex anatomical regions. Hu et al. [2] improved upon this by applying 3D dynamic programming, which offered enhanced structural awareness, although it remained sensitive to noise and tissue variability. To overcome these limitations, unsupervised clustering methods such as Fuzzy C-Means (FCM) were introduced by Gomathi and Thangaraj [3], achieving better segmentation accuracy for diverse nodule morphologies. Shen et al. [4] further proposed a parameter-independent

segmentation method using bidirectional chain coding, which demonstrated improved performance, particularly for irregular and boundary-adjacent nodules. This was coupled with support vector machine (SVM) classification for better generalization.

A comprehensive three-stage detection model encompassing preprocessing, segmentation, and classification was presented in [5]. This model utilized Contrast Limited Adaptive Histogram Equalization (CLAHE) to enhance contrast, followed by Otsu thresholding and region growing for effective segmentation. Subsequently, statistical texture features were extracted from the segmented lung nodules and classified using an artificial neural network (ANN), achieving a commendable accuracy of 93.3%. However, the model's performance was notably reduced near lung boundaries, where nodules often exhibit irregular and indistinct shapes.

In pursuit of more accurate segmentation, Da Silva Sousa et al. [6] introduced a thorax-based region extraction method with refined segmentation logic, yielding 95.21% accuracy. Dehmeshki et al. [7] proposed a genetic template matching technique based on shape index and spherical filters. This method demonstrated a high true positive rate but also suffered from increased false positives, highlighting the challenge of balancing sensitivity and specificity.

Beyond segmentation, significant work has been conducted in feature selection and classification. In [8], statistical texture features such as energy, entropy, contrast, and homogeneity were extracted, and several classifiers—including K-Nearest Neighbors (KNN), Decision Trees, Naïve Bayes, and SVM—were evaluated. Although Decision Trees achieved the highest accuracy at 98.77%, they showed comparatively lower sensitivity, suggesting a trade-off between precision and recall. Parveen and Kavitha [9] utilized Gray-Level Co-occurrence Matrix (GLCM) features to classify lung nodules into benign, malignant, and normal categories using SVM, achieving robust classification performance.

Bio-inspired optimization techniques have also shown considerable promise in feature selection. Kohad et al. [10] integrated Ant Colony Optimization (ACO) with SVM to identify optimal feature subsets, leading to an accuracy of 96.6%. Antony Gnana Singh et al. [11] demonstrated the advantages of combining Particle Swarm Optimization (PSO) with F-score-based feature selection to improve classifier performance across different models. Liu et al. [12] proposed a hybrid framework combining Cuckoo Search (CS), PSO, and SVM, which outperformed standalone GA–SVM and PSO–SVM approaches in terms of classification accuracy.

In a more recent study [13], a hybrid CS-FCM model was proposed for improved segmentation and feature optimization. The system utilized region growing and statistical texture feature extraction, followed by SVM-based classification. This hybrid model achieved a high accuracy of 98.51%, significantly outperforming traditional methods. Additionally, the study highlighted the integration of fog computing for image storage and retrieval, offering scalable and real-time access in distributed systems, which is particularly useful for telemedicine and rural diagnostic centers.

Collectively, these studies underscore the importance of integrating advanced segmentation methods, optimized feature selection, and high-performance classifiers in lung cancer CAD systems. The proposed system in this study builds upon this foundation by combining CS-based feature selection with FCM clustering and SVM classification, while leveraging fog computing to ensure scalable, low-latency deployment suitable for real-time diagnostic applications.

III. PROPOSED METHODOLOGY

The proposed methodology in Fig.1 outlines a comprehensive framework for the classification of gene expression data, leveraging dimensionality reduction and deep learning techniques. The process begins with the import of necessary libraries and data preprocessing, aimed at ensuring data consistency and quality. Given the high-dimensional nature of gene expression datasets, dimensionality reduction is a critical step, implemented through two complementary approaches: feature selection and feature extraction.

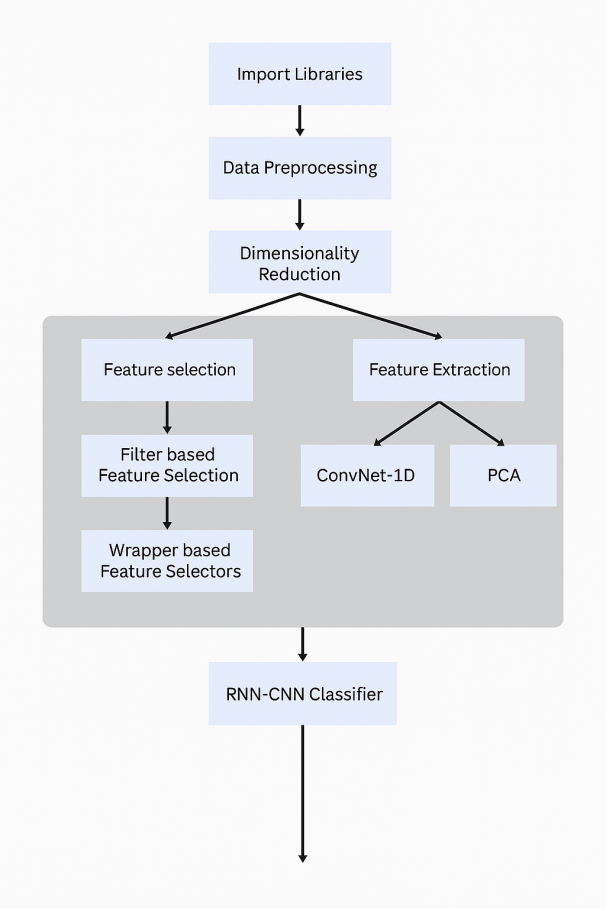

Fig. 1. Flowchart of Proposed Methodology

In the feature selection branch, irrelevant and redundant genes are removed through filter-based methods, such as Variance Thresholding, followed by wrapper-based selectors that iteratively refine the feature set based on model performance. Simultaneously, the feature extraction path employs

techniques such as ConvNet-1D and Principal Component Analysis (PCA) to project the original high-dimensional data into a compact and informative feature space.

The outputs from these dimensionality reduction strategies are subsequently used as inputs to an RNN-CNN hybrid classifier, which is designed to capture both sequential dependencies and spatial patterns within the transformed data, thereby enhancing the model's ability to accurately classify various cancer types from gene expression profiles.

A. Dataset Description

1) Dataset 1: The dataset used in this study consists of RNA-Sequence gene expression profiles collected from 2086 patients, each diagnosed with one of five distinct cancer types: breast invasive carcinoma (BRCA), kidney renal cell carcinoma (KIRC), lung adenocarcinoma (LUAD), lung squamous cell carcinoma (LUSC), and uterine corpus endometrial carcinoma (UCEC). The data includes expression values for 971 genes per patient, with each row representing an individual sample. These gene expression levels serve as features that can be used to analyze or classify the cancer types based on molecular patterns.

The cancer type for each patient is encoded in the final column of the dataset, using numeric labels: 1 for BRCA (878 samples), 2 for KIRC (537 samples), 3 for LUAD (162 samples), 4 for LUSC (240 samples), and 5 for UCEC (269 samples). This structured format enables the dataset to be directly used for supervised learning tasks, particularly multi-class classification problems. It provides a valuable resource for developing and evaluating machine learning models aimed at cancer diagnosis or understanding gene-level distinctions between different cancer types.

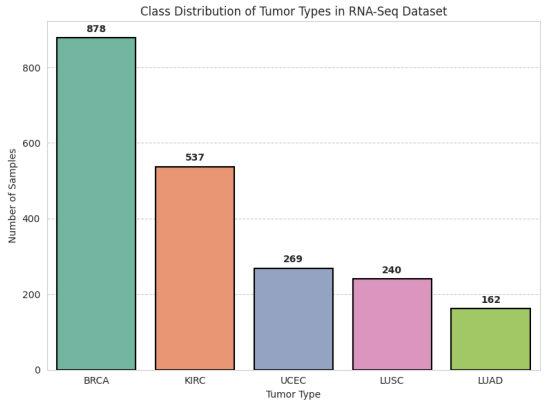

Fig. 2. Class Distribution of Dataset 1.

2) Dataset 2: The dataset is a curated subset of the RNA-Seq (HiSeq) Pan-Cancer (PANCAN) collection derived from The Cancer Genome Atlas (TCGA) project. It contains gene expression profiles for 801 tumor samples, each representing a patient diagnosed with one of five different types of cancer: Breast Invasive Carcinoma (BRCA), Kidney Renal Clear Cell Carcinoma (KIRC), Colon Adenocarcinoma (COAD), Lung Adenocarcinoma (LUAD), and Prostate Adenocarcinoma (PRAD). Each sample is characterized by 20,531 features, corresponding to RNA-Seq gene expression levels measured using the Illumina HiSeq sequencing platform. These expression values are continuous, real-valued measurements that reflect the transcriptional activity of individual genes within the tumor cells.

The dataset is multivariate in nature and is commonly used for classification and clustering tasks in computational biology and bioinformatics. It does not contain any missing values, ensuring consistency and reliability for downstream analysis. The samples are organized row-wise, with each row representing a single patient and each column representing the expression level of a specific gene.

The dataset exhibits a notable class imbalance across the five cancer types represented. Breast Invasive Carcinoma (BRCA) accounts for the largest proportion, comprising 300 out of the 801 total samples. This is followed by Kidney Renal Clear Cell Carcinoma (KIRC) with 146 samples, Lung Adenocarcinoma (LUAD) with 141 samples, Prostate Adenocarcinoma (PRAD) with 136 samples, and Colon Adenocarcinoma (COAD) with the fewest at 78 samples.

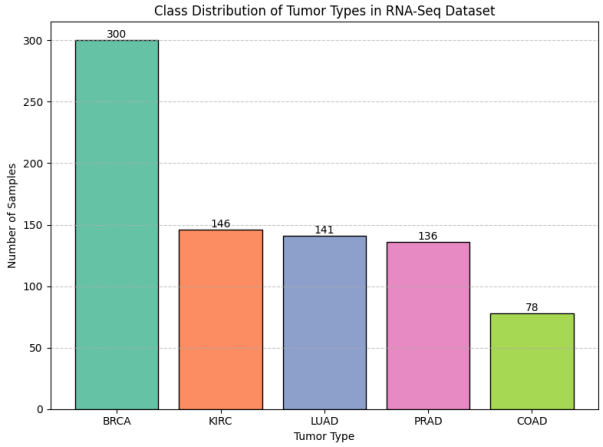

Fig. 3. Class Distribution of Dataset 2.

B. Data Preprocessing:

1) Dataset 1: The preprocessing phase starts by loading the RNA-Seq dataset into a pandas DataFrame and performing an initial inspection to understand its shape and check for missing values. Any columns containing missing values are dropped to maintain data integrity. The dataset is then separated into input features ('X') and target labels ('y'). The target labels represent different cancer types and are initially in categorical numeric form.

To prepare the labels for training in a deep learning model, they are first encoded into integers using LabelEncoder, and then transformed into one-hot encoded format using TensorFlow's tocategorical() function. The dataset is then split into training and testing subsets using an 80:20 ratio to allow proper evaluation of model performance. The resulting sample

size for the training set is 1668, and for the testing set, it is 418.

Finally, the input features are reshaped into a 2D structure with a single channel. Since different feature selection methods may result in different numbers of features, random reshaping is performed to make the data compatible with the CNN-RNN model architecture.

2) Dataset 2: The preprocessing of the RNA-Seq gene expression dataset [14] was meticulously executed to ensure its suitability to carry out further processes. The initial step involved importing two separate CSV files: one containing the gene expression data (data.csv) and the other comprising the corresponding cancer type labels (labels.csv). A verification process confirmed that both files contained an equal number of rows, ensuring a one-to-one correspondence between each sample's gene expression profile and its associated label. Subsequently, the labels were appended as a new column to the gene expression data, culminating in a unified dataset saved as dataset2.csv.

To refine the dataset further, the first column that contained sample identifiers was removed, as it did not contribute meaningful information to the classification task. The categorical labels denoting cancer types were then transformed into numerical format through label encoding, facilitating their compatibility with machine learning algorithms that require numerical input. Specifically, the cancer types were mapped as follows: Prostate Adenocarcinoma (PRAD) to 0, Lung Adenocarcinoma (LUAD) to 1, Breast Invasive Carcinoma (BRCA) to 2, Kidney Renal Clear Cell Carcinoma (KIRC) to 3, and Colon Adenocarcinoma (COAD) to 4.

Following these transformations, the dataset was partitioned into features and labels. The feature matrix X encompassed 20,531 gene expression features per sample, while the label vector y contained the encoded cancer type labels. This structured and cleaned dataset provided a robust foundation for subsequent analyses, ensuring that the models trained on this data would be both reliable and effective in classifying the various cancer types based on gene expression profiles.

C. Dimentionality Reduction:

In gene expression analysis, datasets often contain a vast number of gene features relative to the number of samples, leading to high-dimensional data challenges. This imbalance can result in overfitting, increased computational complexity, and difficulties in model interpretation. Dimensionality reduction techniques address these issues by transforming the original high-dimensional data into a lower-dimensional space, retaining the most informative features while discarding redundant or irrelevant ones. This process not only enhances the performance and efficiency of machine learning models but also aids in uncovering underlying biological patterns by focusing on the most significant gene expressions.In our proposed work, we implemented two distinct dimensionality reduction techniques—feature selection and feature extraction—to address the challenges posed by high-dimensional gene expression data with a limited number of samples.

1) Feature Extraction:

- **PCA** Principal Component Analysis (PCA) is a statistical technique used to reduce the dimensionality of large datasets while preserving as much variability as possible. It achieves this by transforming the original correlated variables into a new set of uncorrelated variables called principal components, which are ordered by the amount of variance they capture from the data.
  For Dataset 2,Principal Component Analysis (PCA) is first applied to reduce the feature space from 19,967 to 801 dimensions.This step effectively condenses the most significant linear patterns in the data, facilitating more efficient processing.
  Following PCA, an autoencoder—a type of neural network designed for unsupervised learning—is utilized to further reduce and encode the data. The autoencoder compresses the 801-dimensional input into a 1,024-dimensional representation through a series of nonlinear transformations. This process captures complex, nonlinear relationships within the data that PCA might not fully encapsulate. The resulting encoded data is then reshaped into a 32×32×1 format, effectively converting the gene expression profiles into grayscale image representations which could be fed into RNN-CNN classifier proposed in the paper [13].
  By combining PCA and autoencoder techniques, this approach aims to harness the strengths of both linear and nonlinear dimensionality reduction methods, enhancing the model's ability to learn meaningful representations from complex gene expression data.
- **ConvNet-1D** In the paper [13], the authors proposed a sandwich stacked bottleneck (SSB) feature extractor to effectively reduce the dimensionality of high-dimensional gene expression data. The SSB mechanism applied a 2D convolutional neural network (CNN) inspired by VGG16 and VGG19 blocks.The method was named as sandwich stacked because VGG19 was stacked between two VGG16 models, which is like a structure of a sandwich. These blocks were used in sequence to extract hierarchical features from subsets of the gene data, where max pooling was repeatedly applied to capture abstract representations. The final bottleneck features were then passed to a classifier for prediction. Although effective, the approach relied on reshaping gene expression data into a 2D format, originally designed for image-based input.
  In order to effectively capture the hierarchical structure of gene expression data while preserving its inherent 1D nature, a custom convolutional neural network architecture was designed using Conv1D layers. The model begins with an input layer that accepts 1D gene expression data with 1024 features. In the first block, two Conv1D layers with 64 filters each extract low-level patterns, followed by a MaxPooling1D layer to reduce dimensionality. The output is stored for a skip connection. The second block

deepens the representation using two Conv1D layers with 128 filters, again followed by pooling. Before merging, the saved output from the first block is processed through a 1x1 Conv1D layer and downsampled to match the dimensions, and is then added to the output of the second block via a skip connection. This helps retain earlier learned information and reinforces feature propagation. The third block continues the hierarchical feature extraction using Conv1D layers with 256 filters, again followed by pooling. Finally, the feature map is flattened and passed through a dense layer to learn complex interactions, and is then reshaped into a 2D format (32×32×1) to align with RNN-CNN classifier requirements.

We chose to implement SSB-1D with Conv1D as a response to limitations in the original paper's use of Conv2D for gene expression. While 2D CNNs are powerful for spatial data like images, applying them to 1D biological sequences could introduce structural artifacts and unnecessary complexity. Our 1D design keeps the data structure biologically intuitive, simplifies the model, and allows for effective feature learning tailored to genomics.

2) Feature Selection: The feature selection process in this project was conducted in two stages: a filter-based method followed by a wrapper-based approach.

1)Filter-Based Feature Selection: Variance Threshold Filter-based feature selection methods independently assess each feature's relevance using statistical metrics, such as correlation with the target variable. This approach is computationally efficient and suitable for initial data preprocessing, particularly in high-dimensional datasets like gene expression profiles. The first stage involved the application of the Variance Threshold technique, a filterbased method that eliminates features exhibiting low variance across samples. Features with minimal variance are considered to carry limited information and may not contribute significantly to the predictive power of a model. By setting a threshold of 0.01, features with variance below this value were removed, reducing the feature set from 20,531 to 19,967. This step effectively discarded features that were nearly constant across all samples, streamlining the dataset and potentially enhancing model performance.

2)Wrapper based Feature Selection

Following the filter-based reduction, wrapper-based feature selection methods were employed to further refine the feature set. Wrapper methods evaluate subsets of features by training and testing a specific machine learning model, selecting the combination that yields the best performance. This approach considers the interaction between features and the model, allowing for the identification of feature subsets that are most predictive for the specific task. While computationally more intensive than filter methods, wrapper methods can lead to improved model accuracy by selecting features that are most relevant to the predictive task.

By integrating both filter and wrapper-based methods, the feature selection process aimed to balance computational efficiency with model performance, ensuring that the final feature set was both manageable in size and rich in predictive information.The methods implemented in the work are as follows:

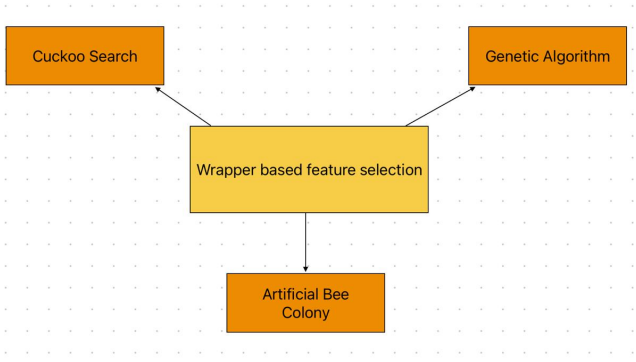

Fig. 4. Wrapper based Feature Selection Methods implemented

- **Cuckoo Search(CS) :** The Cuckoo Search (CS) algorithm is a metaheuristic inspired by the brood parasitism of certain cuckoo species, used here for feature selection to enhance classification accuracy. It starts with a population of random binary solutions (cuckoos), where each gene indicates whether a feature is selected. In each generation, new solutions are generated via Levy flights, mimicking the exploratory behavior of cuckoos, and these are used to replace existing ones if they yield higher classification accuracy using a Random Forest classifier. A fraction of poor solutions (nests) are periodically abandoned and replaced by new random solutions to maintain diversity. Over successive generations, the population evolves toward feature subsets that yield better classification performance. The best-performing solution is selected as the optimal feature set for model training and evaluation.

Algorithm 1 Cuckoo Search Algorithm for Feature Selection

```
Initialize cuckoo population (binary vectors)
for each generation do
    for each cuckoo do
        Generate a new solution via Lévy flight
        if new solution is better than current then
            Replace current solution with new one
        end if
    end for
    Sort cuckoos by fitness
    Replace worst nests with new random solutions (abandonment)
end for
return Best feature subset =0
```

- **Artificial Bee Colony (ABC):** The Artificial Bee Colony (ABC) algorithm is a swarm intelligence-based optimization technique inspired by the foraging behavior of honey bees, applied here for feature selection. The algorithm

starts with a population of binary vectors representing feature subsets, where each bit indicates whether a feature is selected. In each iteration, employed bees explore new solutions by modifying their current ones using information from randomly chosen peers. If a new solution improves the classification fitness (evaluated via a custom fitness function), it replaces the old one. During the onlooker bee phase, solutions are probabilistically selected based on their fitness and refined similarly. If a solution fails to improve over a set number of trials, the scout bee phase replaces it with a new random solution to introduce diversity. This process iterates until a stopping criterion is met, and the best-performing solution is selected as the optimal subset of features, balancing dimensionality reduction with predictive performance.

Algorithm 2 Artificial Bee Colony (ABC) Algorithm for Feature Selection

```
Initialize population of feature subsets (solutions)
for each generation do
    Employed Bee Phase:
    for each employed bee do
        Generate a neighbor solution
        Evaluate its fitness (e.g., F1-score)
        Replace current solution if neighbor is better
    end for
    Onlooker Bee Phase:
    Compute selection probabilities based on fitness
    for each onlooker bee do
        Select a solution probabilistically
        Generate and evaluate a new neighbor
        Replace if improved
    end for
    Scout Bee Phase:
    for solutions that haven't improved for a limit do
        Replace with a new random solution
    end for
end for
return Best feature subset found =0
```

- **Genetic Algorithm(GA):** The Genetic Algorithm (GA) is a bio-inspired optimization technique used here for feature selection to improve model performance by identifying the most relevant features. It begins with a population of randomly generated binary individuals, where each gene represents whether a feature is selected (1) or not (0). The fitness of each individual is evaluated using the F1 score of an XGBoost classifier trained and tested on the selected features. Through a process mimicking natural evolution—selection (tournament), crossover (two-point), and mutation (bit flipping)—the population evolves over multiple generations to maximize the fitness score. The best-performing individual is stored in a Hall of Fame, and its selected features are used for model training and testing. This approach efficiently navigates the vast search space of possible feature subsets to find a near-optimal solution.

Algorithm 3 Genetic Algorithm for Feature Selection

```
Initialize population of binary feature subsets
for each generation do
    Evaluate fitness of each individual (e.g., F1-score)
    Select parents using tournament selection
    Apply crossover and mutation to generate offspring
    Replace population with offspring
    Update Hall of Fame with best individual
end for
return Best feature subset from Hall of Fame =0
```

D. RNN-CNN classifier:

The proposed hybrid RNN-CNN classifier in the base paper [13] was designed to effectively perform multi-class classification of cancer types using high-dimensional gene expression data. This approach combined the feature extraction capabilities of Convolutional Neural Networks (CNNs) with the sequential modeling strength of Recurrent Neural Networks (RNNs), offering a comprehensive solution for analyzing gene expression patterns.

To facilitate CNN processing, the feature vector is reshaped from one dimension into a pseudo-image format.For Dataset-1, the 961-dimensional vector is reshaped into a (31 × 31) matrix and further reshaped to (32 × 32 × 1), which serves as the input to the CNN module. For Dataset-2, the 20531-dimensional vector is reshaped into a (143 × 143) matrix and further reshaped to (32 × 32 × 1), which serves as the input to the CNN module.

*1) **CNN-Based Feature Extraction**:* The CNN module comprises three consecutive convolutional blocks. Each block includes a 2D convolution layer followed by ReLU activation, batch normalization, and dropout (0.25) for regularization. The configuration of each block is as follows:

Block 1: Applies 32 filters of size 2 × 2.

Block 2: Applies 64 filters of size 2 × 2.

Block 3: Applies 128 filters of size 2 × 2 and includes a dropout layer.

The output of the third block, having the shape (4 × 4 × 128), is reshaped into (4 × 512) to serve as input to the recurrent layers.

*2) **RNN-Based Sequence Modeling**:* The reshaped feature map is fed into a sequence of recurrent layers, which includes a simple RNN layer with 128 units, followed by an LSTM (Long Short-Term Memory) layer with 128 units, a Dropout layer (0.25) applied after the LSTM layer to prevent overfitting. These layers enable the model to capture temporal dependencies and sequential patterns in gene expression that may be indicative of cancer types.

*3) **Final Classification**:* The output of the RNN layers is flattened into a 1D vector and passed through another dropout layer (0.25) to enhance generalization. This vector is finally connected to a fully connected dense layer with five output neurons corresponding to the number of classes.

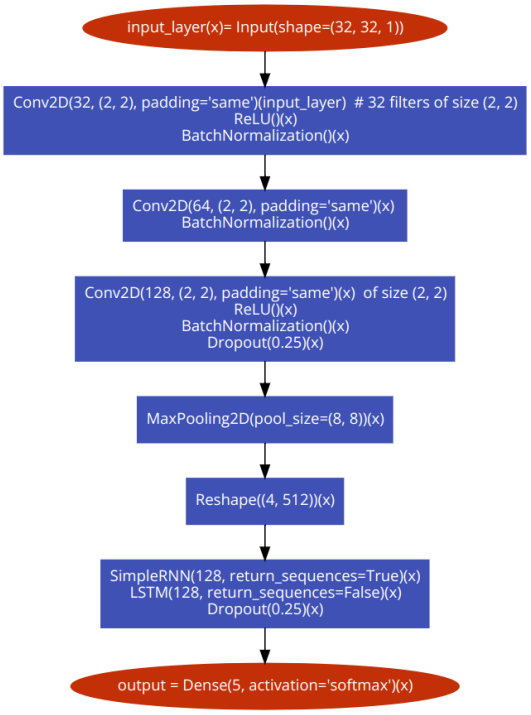

Fig. 5. Flowchart of RNN-CNN Classifier

A Softmax activation function is used to obtain the final class probabilities. This proposed model is trained using the following setup: ReLU (hidden layers), Softmax (output layer) activation functions, Adam optimizer, Poisson Loss Function.

The hybrid RNN-CNN architecture effectively addresses the challenges posed by high-dimensional and noisy gene expression data. CNN layers efficiently extract spatial gene patterns, while RNN and LSTM layers model the sequential dependencies among gene expression levels. This integration improves the model's ability to discern subtle and complex variations associated with different cancer types. The layered use of dropout and batch normalization further enhances robustness and reduces overfitting, making the model suitable for practical deployment in clinical settings. This RNN-CNN based approach demonstrated improved performance in classifying gene expression data across multiple cancer types, validating its efficacy as a reliable computational tool in bioinformatics and precision oncology.

IV. Results and Analysis

This section presents the comparative performance analysis of different deep learning architectures with and without the proposed Stacked Sandwich Bottleneck (SSB) model, as well as feature extraction performance on Dataset 1 and Dataset 2.

A. Dataset 1

*1) **Comparison of Different Feature Extractors**:* Table I presents the performance comparison of five deep learning classifiers—SSB, VGG16, VGG19, InceptionV3, and ResNet50—on the gene expression data from Dataset-1. The evaluation is based on accuracy and mean squared error (MSE).

Among the models, ResNet50 achieved the highest performance with an accuracy of 0.9378 and a relatively low MSE of 0.2153. VGG16 and VGG19 also showed strong results, with accuracies of 0.9330 and 0.9139, and corresponding MSE values of 0.1651 and 0.3469, respectively. InceptionV3 performed moderately with an accuracy of 0.8206 and an MSE of 0.4067.

In contrast, the SSB model showed the lowest performance, with an accuracy of 0.6411 and the highest MSE of 1.8182, suggesting that the SSB feature extractor may not be suitable for Dataset-1 in isolation.

TABLE I
ACCURACY AND MSE OF SELECTED FEATURE EXTRACTORS ON DATASET 1

Model	Accuracy	MSE
SSB	0.641	1.818
VGG16	0.933	0.165
VGG19	0.914	0.347
InceptionV3	0.821	0.407
ResNet50	0.938	0.215

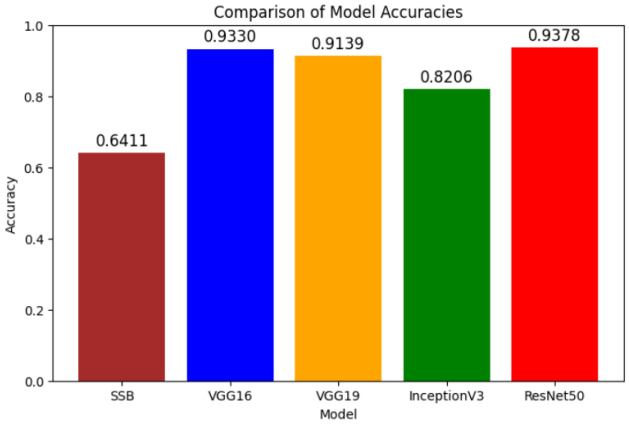

Fig. 6. Comparison of Accuracy of feature extractors

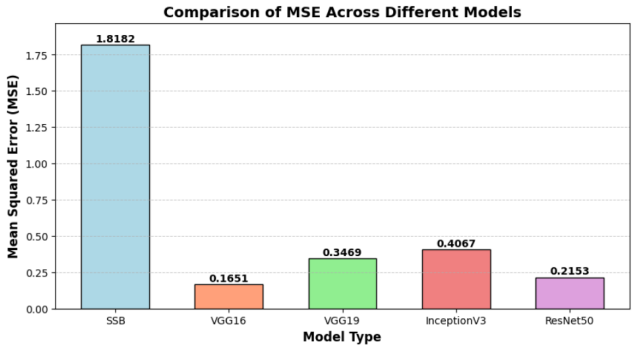

Fig. 7. Comparison of MSE of feature extractors

*2) **Comparison of Different Deep Learning Classifiers**:* Table II presents a comparative analysis of multiple deep learning models on Dataset-1 using gene expression data. The RNN-CNN model without SSB achieved the highest overall performance with an accuracy of 0.97, a low MSE of 0.07, and strong precision, recall, and F1-score values ($\geq$ 0.94). In contrast, the RNN-CNN+SSB model performed the worst, indicating that the stacking strategy may not be effective for Dataset-1. VGG16 and VGG19 also performed consistently well, each achieving 0.95 accuracy and F1-scores around 0.92, whereas models like InceptionV3 and MobileNet showed comparatively lower metrics.

TABLE II
COMPARISON OF MODELS ON DATASET 1

Model	Accuracy	MSE	Precision	Recall	F1-Score
VGG16	0.95	0.13	0.92	0.93	0.92
VGG19	0.95	0.11	0.92	0.92	0.92
InceptionV3	0.75	1.38	0.70	0.74	0.71
ResNet50	0.84	0.71	0.82	0.78	0.79
MobileNet	0.78	1.02	0.71	0.70	0.70
RNN-CNN	0.97	0.07	0.95	0.94	0.94
RNN-CNN+SSB	0.58	2.24	0.44	0.42	0.39

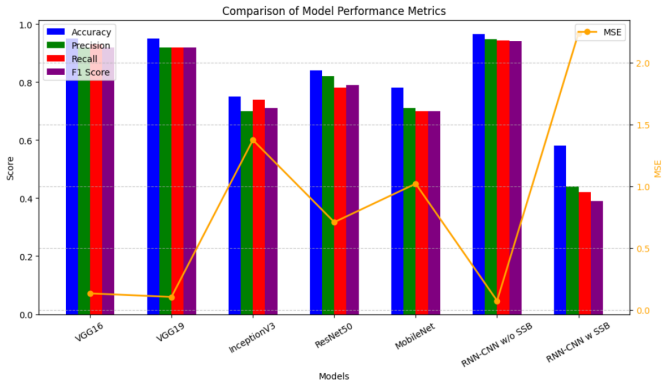

Fig. 8. Comparison of Accuracy of feature extractors

The results indicate that even without the use of the SSB stacking strategy, a well-structured hybrid model like RNN-CNN can surpass deeper CNN-based architectures by effectively capturing contextual patterns in gene expression data. Owing to its strong and consistent performance across all metrics, the RNN-CNN classifier was chosen for further experiments involving various feature selection and extraction techniques, such as PCA, filter-based selection, and custom convolutional embedding layers.

The confusion matrix in Fig. 10, generated by the RNN-CNN classifier without applying any feature selection on Dataset-1, shows generally strong classification performance, though minor misclassifications are evident in the BRCA, LUAD, LUSC, and UCEC classes. This outcome reflects the baseline capabilities of the model when using the full feature set. To further improve accuracy and minimize these

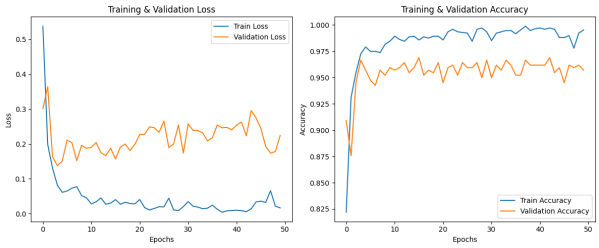

Fig. 9. Graphs demonstrating Training and validation accuracy and loss for RNN-CNN classifier without SSB feature Extractor for Datset-1

classification errors, multiple feature selection strategies were explored in the subsequent experiments.

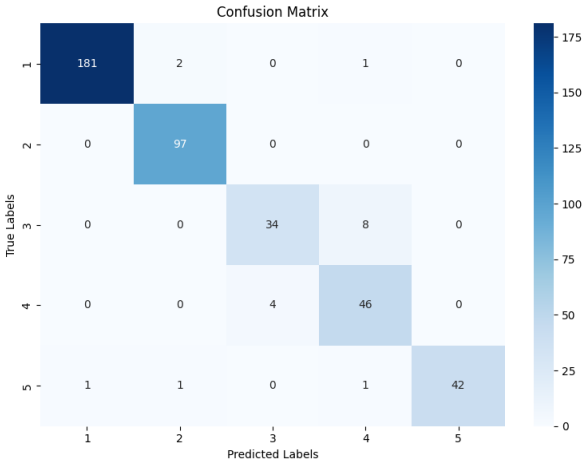

Fig. 10. Confusion Matrix for RNN-CNN classifier without using Feature Extractor

*3) **Comparison of Feature Selection/Extraction Methods using RNN-CNN Classifier**:* Features are selected from the dataset using nature-inspired optimization algorithms: Cuckoo Search, Artificial Bee Colony (ABC), and Genetic Algorithm. These methods aim to identify the most relevant features while reducing dimensionality and improving model performance. The Genetic Algorithm selects 479 features, Cuckoo Search selects 610 features, and the ABC algorithm selects 496 features. Additionally, Principal Component Analysis (PCA) is employed as a statistical dimensionality reduction method, where 961 components are retained to preserve maximum variance in the data.

The selected features are then used to train deep learning models, and their performance is evaluated using metrics such as accuracy and Mean Squared Error (MSE). The dataset is divided using an 80:20 training-testing ratio, where 0.8 is used for training and 0.2 for testing.

The accuracy comparison of models using features selected by the three optimization algorithms and PCA. The Cuckoo Search-based selection achieves the highest accuracy of 0.9737, followed by the Genetic Algorithm with 0.9713, and ABC with 0.9689. PCA also demonstrates strong performance with an overall test accuracy of 0.9522. Its classification

report reveals a precision of 0.9545, recall of 0.9522, and F1-score of 0.9529. The MSE comparison further supports the effectiveness of these methods, with Cuckoo Search achieving the lowest MSE of 0.0091, followed by Genetic Algorithm at 0.0096, PCA at 0.0161, and ABC yielding a slightly higher MSE of 0.0117.

Across all training-testing ratios, the proposed methods show that Cuckoo Search consistently provides the highest accuracy and lowest MSE among the optimization-based techniques, while PCA also proves to be a robust alternative for dimensionality reduction in gene expression classification tasks.

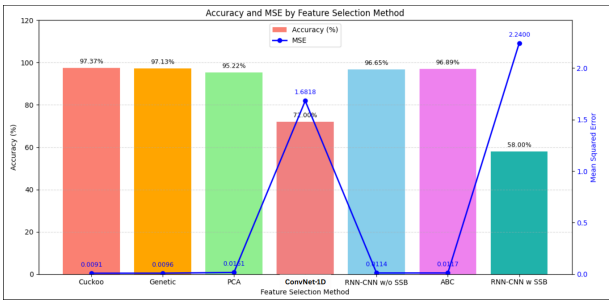

Fig. 11. Comparison of Accuracy and MSE scores of Classifiers using different feature selection methods

Table III shows across all training-testing ratios, the proposed methods show that Cuckoo Search consistently provides the highest accuracy and lowest MSE among the optimization-based techniques, while PCA also proves to be a robust alternative for dimensionality reduction in gene expression classification tasks.

TABLE III
COMPARISON OF FEATURE SELECTION AND EXTRACTION METHODS ON DATASET 1

Method	Accuracy	MSE	Precision	Recall	F1-Score
CS	0.974	0.009	0.975	0.974	0.974
GA	0.971	0.010	0.972	0.971	0.972
PCA	0.952	0.016	0.955	0.952	0.953
ConvNet-1D	0.720	1.682	0.708	0.584	0.604
RNN-CNN	0.967	0.011	0.968	0.967	0.966
ABC	0.969	0.012	0.969	0.969	0.969
SSB	0.580	2.240	0.440	0.420	0.390

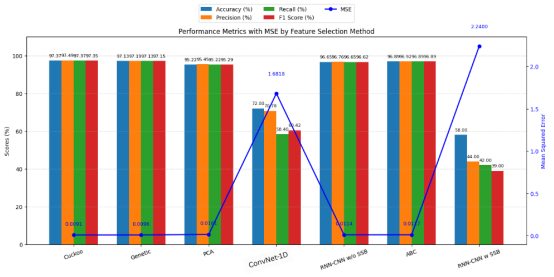

Fig. 12. Comparison of Precision, Recall, and F1-scores of classifiers using different feature selection methods

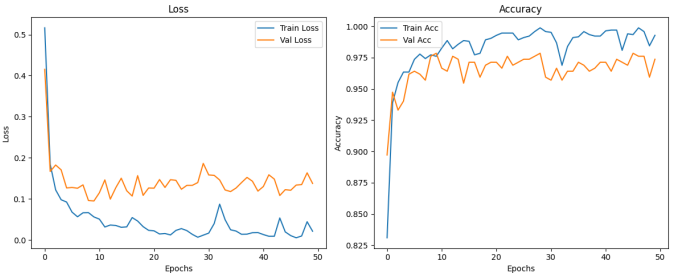

Fig. 13. Graphs demonstrating Training and validation accuracy and loss for RNN-CNN classifier using Cuckoo Search Feature selection for Dataset-1

The confusion matrix presented in Fig.14 reflects the classification results of the RNN-CNN model integrated with Cuckoo Search-based feature selection for Dataset-1. Achieving a test accuracy of 97.37%, the model demonstrates excellent predictive performance across all five cancer types—BRCA, KIRC, LUAD, LUSC, and UCEC. The few misclassifications observed indicate that the selected features effectively capture key discriminative information, while the RNN-CNN architecture exhibits strong generalization capabilities in analyzing complex biomedical data.

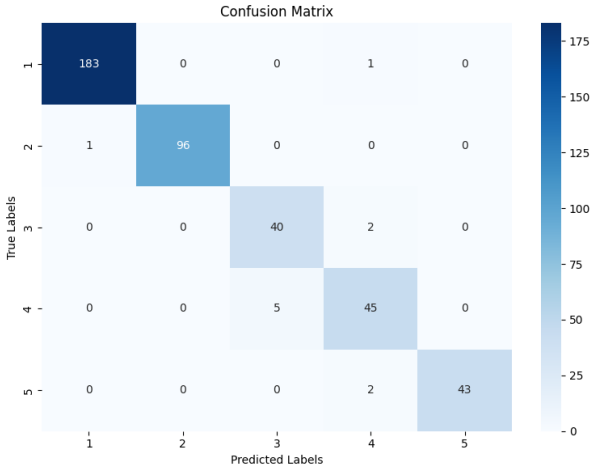

Fig. 14. Confusion Matrix for RNN-CNN classifier using Cuckoo Search Feature selection

B. Dataset 2

1) ***Comparison of Different Feature Extractors****:* Table V compares the performance of six deep learning classifiers—VGG16, VGG19, InceptionV3, ResNet50, MobileNet, and RNN-CNN—on gene expression data enhanced with the SSB feature Extractor, which follows a VGG16-VGG19-VGG16 stacking design.

The results in Table IV show that VGG16 with SSB achieved the highest accuracy of 0.61, followed closely by RNN-CNN (0.60). Both models reported F1-scores of 0.60, indicating consistent and balanced classification performance. In contrast, lightweight architectures like MobileNet performed poorly, achieving an accuracy of 0.38 and F1-score of 0.11,

likely due to insufficient representational power for complex gene patterns. Similarly, deeper models like InceptionV3 and ResNet50 underperformed, with F1-scores of 0.31 and 0.46, respectively, possibly due to overfitting on limited training data or architectural incompatibility with the SSB-extracted features.

TABLE IV
COMPARISON OF FEATURE EXTRACTORS ON DATASET 2

Model	Accuracy	MSE	Precision	Recall	F1-Score
VGG16	0.7516	0.7205	0.7738	0.7736	0.7570
VGG19	0.7578	0.9876	0.7645	0.8155	0.7584
ResNet50	0.8509	0.5652	0.9079	0.7959	0.8203
InceptionV3	0.6460	1.4161	0.7885	0.5740	0.5198
SSB	0.6211	1.4472	0.7576	0.8621	0.8065

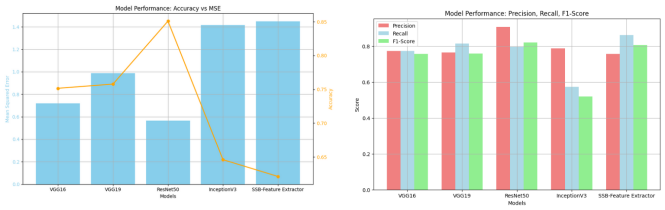

Fig. 15. (A)Comparison of Accuracy and MSE scores of feature extractors (B)Comparison of Precision, Recall, and F1-scores of feature extractors

*2) **Comparison of Deep learning based Classifiers with SSB for Dataset 2**:* Table V compares the performance of six deep learning classifiers—VGG16, VGG19, InceptionV3, ResNet50, MobileNet, and RNN-CNN—on gene expression data enhanced with the proposed SSB module, which follows a VGG16-VGG19-VGG16 stacking design.

The SSB model acts as a bottleneck feature extractor by compressing and refining the high-dimensional gene expression data through multiple convolutional layers in a sandwich structure. This architecture enables deep feature representation learning, particularly beneficial for biological datasets with complex and sparse patterns.

TABLE V
COMPARISON OF CLASSIFIERS WITH SSB FOR DATASET 2

Model	Accuracy	MSE	Precision	Recall	F1-Score
VGG16	0.61	1.354	0.60	0.61	0.60
VGG19	0.50	1.795	0.55	0.54	0.50
InceptionV3	0.43	1.938	0.41	0.31	0.31
ResNet50	0.46	1.957	0.47	0.46	0.46
MobileNet	0.38	1.478	0.08	0.20	0.11
RNN-CNN	0.60	1.360	0.63	0.60	0.60

The results in Table V show that VGG16 with SSB achieved the highest accuracy of 0.61, followed closely by RNN-CNN (0.60). Both models reported F1-scores of 0.60, indicating consistent and balanced classification performance. In contrast, lightweight architectures like MobileNet performed poorly, achieving an accuracy of 0.38 and F1-score of 0.11, likely due

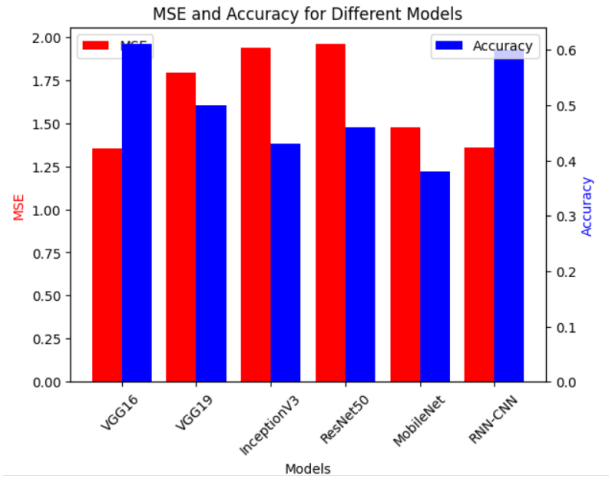

Fig. 16. Comparison of Accuracy and MSE scores of Classifiers with SSB

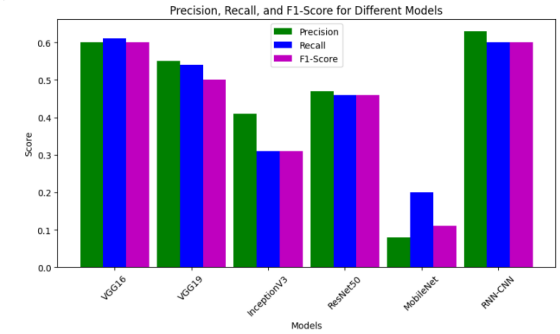

Fig. 17. Comparison of Precision, Recall, and F1-scores of classifiers without SSB

to insufficient representational power for complex gene patterns. Similarly, deeper models like InceptionV3 and ResNet50 underperformed, with F1-scores of 0.31 and 0.46, respectively.

However, it is important to acknowledge that the overall performance across models was relatively limited compared to other datasets, due to two major challenges in Dataset 2:

- Lack of Clarity in Preprocessing: The base paper [11] implemented did not clearly specify which columns (features) needed to be dropped or retained. This ambiguity might have led to the inclusion of noisy, irrelevant, or redundant features, adversely affecting model learning and generalization.
- Presence of Zero-Valued Columns: A significant portion of the dataset contained columns with all values being 0.0, which offer no discriminative information. If not properly filtered, these columns may have introduced bias and sparsity, leading to misleading feature representations and lower classification performance, particularly for deep learning models which rely heavily on informative feature variance.

*3) **Comparison Between Different Classifiers Without SSB**:* Table VI presents the performance of the same classifiers applied directly to reshaped gene expression data, without using SSB. Notably, VGG16 significantly outperformed all other models, achieving the highest accuracy of 0.97, lowest MSE of 0.0683, and an F1-score of 0.98. Similarly, RNN-CNN achieved an accuracy of 0.94 and an F1-score of 0.96. These results demonstrate that these models can effectively learn from raw reshaped gene data, even without additional feature extraction.

In contrast, MobileNet and InceptionV3 failed to perform effectively in this setting, recording F1-scores of 0.55 and 0.59, respectively. These findings suggest that while some deep networks like VGG can robustly extract features from raw data, others like InceptionV3 may require guided feature learning through modules like SSB.

TABLE VI
COMPARISON BETWEEN DIFFERENT CLASSIFIERS WITHOUT SSB FOR DATASET 2

Model	Accuracy	MSE	Precision	Recall	F1-Score
VGG16	0.97	0.0683	0.95	1.00	0.98
VGG19	0.94	0.2236	0.86	0.93	0.95
InceptionV3	0.59	1.7702	0.21	0.14	0.59
ResNet50	0.38	1.4286	0.50	0.55	0.64
MobileNet	0.55	1.4783	0.00	0.00	0.55
RNN-CNN	0.94	0.1429	0.91	1.00	0.96

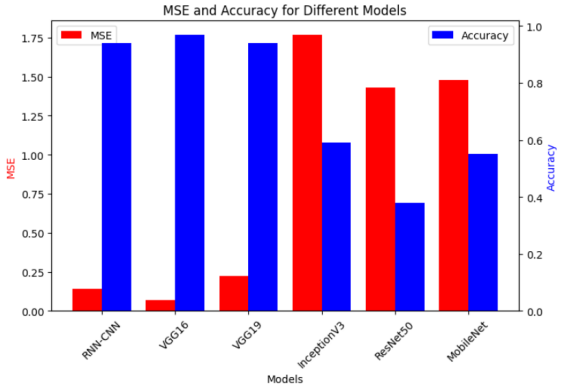

Fig. 18. Comparison of Accuracy and MSE scores of Classifiers without SSB

These strong results suggest that even in the absence of SSB stacking, a well-designed hybrid architecture like RNN-CNN can outperform deeper networks by efficiently learning contextual relationships within the data. Furthermore, due to its consistent performance and generalization capabilities, the RNN-CNN classifier has been selected for all subsequent experiments involving different feature selection and feature extraction methods, including PCA, filter-based selection, and custom convolutional embedding layers.

The confusion matrix in Fig. 21 obtained from the RNN-CNN classifier without any feature selection for Dataset-2 demonstrates accurate classification for most samples, with slight misclassifications observed across BRCA, KIRC, and

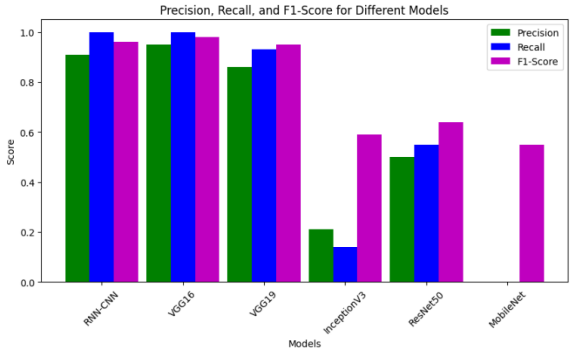

Fig. 19. Comparison of Precision, Recall, and F1-scores of classifiers without SSB

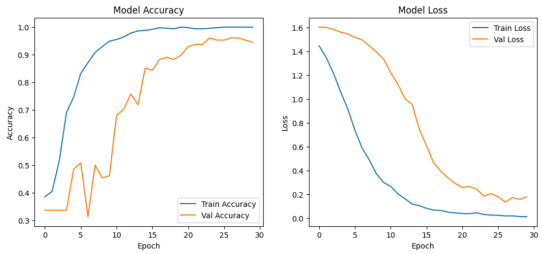

Fig. 20. Graphs demonstrating Training and validation accuracy and loss for RNN-CNN classifier without SSB feature Extractor

COAD classes. These results highlight the model's baseline performance. Thus, to enhance classification accuracy and reduce misclassifications, various feature selection techniques were applied in subsequent experiments.

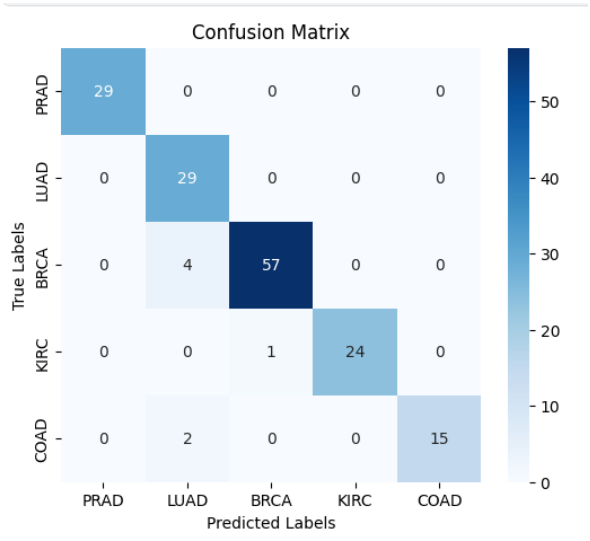

Fig. 21. Confusion Matrix for RNN-CNN classifier without using Feature Extractor

*4) **Comparison of Feature Selection/Extraction Methods using RNN-CNN Classifier**:* Features from the Dataset-2 are

extracted using bio-inspired metaheuristic optimization techniques—**Cuckoo Search, Artificial Bee Colony (ABC), and the Genetic Algorithm**—to effectively reduce dimensionality and enhance model performance. Out of the original 19,967 features, the Cuckoo Search algorithm selected **10,022 features**, while ABC and Genetic Algorithm selected **9,914 and 9,913 features**, respectively. Notably, approximately **4,950 features** were found to be common across all three methods, indicating a shared agreement on the most informative features.

These reduced feature sets served as input to deep learning classifier to evaluate their impact on prediction accuracy. The dataset is split into a training and testing ratio of **80:20**, where 80% of the data is used for model training and the remaining 20% for performance evaluation using metrics such as **Accuracy,Mean Squared Error (MSE),Precision, Recall and F1 score**.

TABLE VII
PERFORMANCE COMPARISON OF FEATURE SELECTION/EXTRACTION METHODS

Method	Accuracy	MSE	Precision	Recall	F1-Score
CS	1.00	0.0000	1.00	1.00	1.00
GA	1.00	0.0000	1.00	1.00	1.00
PCA	0.98	0.0186	0.98	0.98	0.98
ConvNet-1D	0.96	0.2981	0.96	0.96	0.96
RNN-CNN	0.96	0.1429	0.96	0.96	0.96
ABC	1.00	0.0000	1.00	1.00	1.00
SSB	0.60	0.4000	0.63	0.60	0.61

Table VII shows the performance comparison of various feature selection and extraction methods used with the RNN-CNN classifier on Dataset-2. The evaluation is based on standard metrics including Accuracy, Mean Squared Error (MSE), Precision, Recall, and F1-Score.

Among all methods evaluated, the Cuckoo Search, Genetic Algorithm, and Artificial Bee Colony (ABC) consistently delivered perfect results across all metrics (Accuracy, Precision, Recall, F1-Score = 1.0 and MSE = 0.0). These optimization-driven approaches likely succeeded in identifying the most relevant and discriminative features, avoiding noise or redundant data.

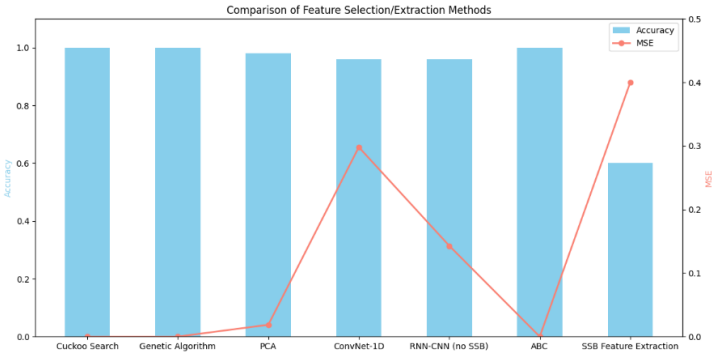

Fig. 22. Comparison of Accuracy and MSE scores of Classifiers using different feature selection methods

PCA also demonstrated strong performance with accuracy of 0.98, and balanced values for all metrics (Precision, Recall, F1-Score = 0.98) while keeping the MSE low at 0.0186. This reaffirms PCA's effectiveness in dimensionality reduction while preserving informative variance in the data.

Deep learning-based extractors like ConvNet-1D and RNN-CNN (without SSB) both achieved strong metrics (accuracy = 0.96, F1-score = 0.96), indicating that automatic hierarchical feature extraction via neural networks is viable, even without optimization-based selection.

Fig. 23. Comparison of Precision, Recall, and F1-scores of classifiers using different feature selection methods

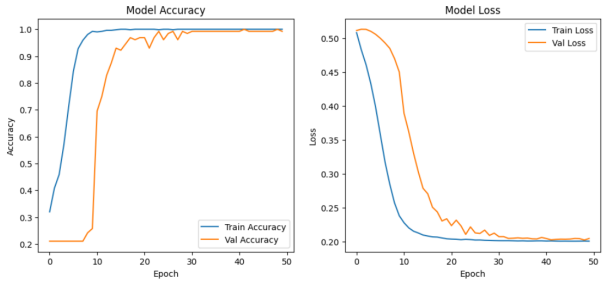

Fig. 24. Graphs demonstrating Training and validation accuracy and loss for RNN-CNN classifier using Cuckoo Search Feature selection

The confusion matrix in Fig.25 obtained using the RNN-CNN classifier in conjunction with Cuckoo Search-based feature selection for Dataset-2 demonstrates perfect classification performance. All samples across the five cancer types—PRAD, LUAD, BRCA, KIRC, and COAD—were correctly predicted, indicating 100% accuracy. This result highlights the effectiveness of the selected features in capturing discriminative patterns and the robustness of the RNN-CNN model in handling complex biomedical data.

V. CONCLUSION

This study explored an end-to-end deep learning pipeline for multi-class cancer classification using Dataset-1 and Dataset-2, a high-dimensional gene expression dataset. The methodology involved three main stages: data reshaping, model evaluation, and feature selection.

In the first stage, gene expression data was reshaped into 2D matrices, enabling the use of CNN-based architectures.

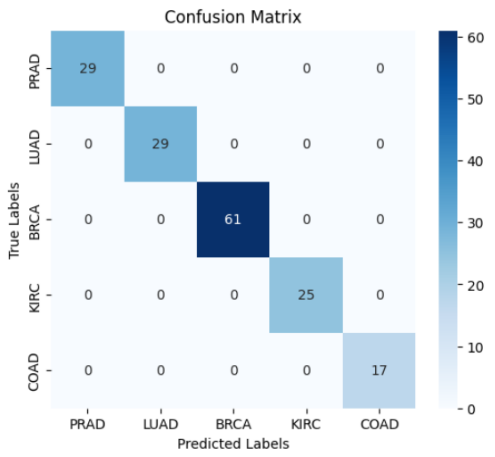

Fig. 25. Confusion Matrix for RNN-CNN classifier using Cuckoo Search Feature selection

Removing the SSB layer and feeding reshaped data directly into deep models like VGG16, VGG19, ResNet50, InceptionV3, MobileNet, and a custom RNN-CNN led to significant accuracy improvements. VGG16 and the RNN-CNN hybrid model performed best, achieving high accuracies.

In the second stage, we applied three metaheuristic feature selection algorithms—Cuckoo Search (CS), Artificial Bee Colony (ABC), and Genetic Algorithm (GA)—to reduce dimensionality and enhance performance. All three yielded perfect classification results, outperforming both PCA and raw-data approaches. Among these, the CS-RNN-CNN combination proved most efficient in balancing computational cost and accuracy.

In conclusion, this study demonstrates that combining deep learning with bio-inspired feature selection can yield highly accurate and robust cancer classification models, offering a promising direction for future biomedical diagnostics. We have successfully applied the proposed methodology for GED, wherein number of features is large and number of samples is rather small.

REFERENCES

[1] S.G. Armato III et al., "Automated lung nodule detection in CT scans: Preliminary results," *Med. Phys.*, vol. 26, no. 11, pp. 2422–2430, 1999.

[2] S. Hu et al., "A 3D model-based method for automated segmentation of lungs with severe interstitial lung disease," *Acad. Radiol.*, vol. 8, no. 12, pp. 1232–1243, 2001.

[3] R. Gomathi and P. Thangaraj, "A novel approach for lung segmentation using FCM with spatial information," *Int. J. Eng. Sci. Technol.*, vol. 2, no. 10, pp. 6020–6024, 2010.

[4] R. Shen et al., "Juxta-pleural pulmonary nodule segmentation based on bidirectional chain coding," *Comp. Med. Imag. Graph.*, vol. 39, pp. 37–46, 2015.

[5] M. Jinsa and R. Gunavathi, "Lung cancer cell classification using ANN classifier," *Proc. Int. Conf. Comput. Commun. Informatics*, 2014.

[6] D. Da Silva Sousa et al., "Automatic detection of pulmonary nodules in CT images: A survey," *Comp. Biol. Med.*, vol. 41, no. 7, pp. 483–497, 2011.

[7] J. Dehmeshki et al., "Automated detection of lung nodules in CT images using shape index," *Pattern Recognit.*, vol. 41, no. 10, pp. 3265–3276, 2008.

[8] R. Aarthy and R. Ragupathy, "Lung cancer detection using CT image," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 2, no. 11, pp. 74–77, 2012.

[9] S. Parveen and A. Kavitha, "Classification of lung cancer stages using SVM," *Int. J. Comput. Appl.*, vol. 98, no. 17, pp. 26–32, 2014.

[10] V. Kohad et al., "Feature selection using ant colony optimization for medical diagnosis," *Int. J. Comput. Appl.*, vol. 89, no. 15, pp. 23–29, 2014.

[11] Kaur, S., Kumar, Y., Koul, A. *et al.* A Systematic Review on Metaheuristic Optimization Techniques for Feature Selections in Disease Diagnosis: Open Issues and Challenges. *Arch Computat Methods Eng*

[12] Y. Liu et al., "An improved hybrid model using cuckoo search and SVM for disease diagnosis," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 9257–9265, 2014.

[13] T. Thakur, I. Batra, A. Malik, D. Ghimire, S. -H. Kim and A. S. M. Sanwar Hosen, "RNN-CNN Based Cancer Prediction Model for Gene Expression," in *IEEE Access*, vol. 11, pp. 131024-131044, 2023, doi: 10.1109/ACCESS.2023.3332479.

[14] Gene Expression Cancer RNA-Seq Dataset, UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/dataset/401/gene+expression+cancer+rna+seq

[15] Cancer Types: RNA Sequencing Values From Tumor Samples/Tissues. Accessed: Jul. 2020. [Online]. Available: https://data.mendeley.com/datasets/sf5n64hydt/1