Journal of Scientific & Industrial Research
Vol. 83, April 2024, pp. 392-404
DOI: 10.56042/jsir.v83i4.2670

An Optimized Hybrid Model for Classifying Bacterial Genus using an Integrated CNN-RF Approach on 16S rDNA Sequences

M Meharunnisa[1], M Sornam[2]* & B Ramesh[3]

[1]Ethiraj College for Women, Department of BCA, Chennai 600 008, Tamil Nadu, India

[2]University of Madras, Department of Computer Science, Guindy Campus, Chennai 600 025, Tamil Nadu, India

[3]Sri Sankara Arts and Science College, Department of Biotechnology, Kanchipuram 631 561, Tamil Nadu, India

Received 14 June 2023; revised 18 December 2023; accepted 08 March 2024

The classification of the bacterial genus based on 16S ribosomal DNA (rDNA) sequences is crucial in microbiology and medical research. In recent years, deep learning techniques such as Convolutional Neural Networks (CNNs) have shown promising results in this field. However, these models are limited by the need for large annotated datasets and can be prone to overfitting. On the other hand, Random Forest (RF) algorithms are well known for their accuracy and robustness, but lack the ability to capture complex patterns in sequences. In this study, we propose a hybrid CNN-RF model to address these limitations and improve the classification of the bacterial genus based on 16S rDNA sequences. Our model combines the strengths of both approaches by using CNNs to extract features from the sequences and RF to make the final classification decision. The proposed hybrid model was evaluated on a 16S rDNA sequence dataset and showed improved performance compared to both standalone CNN and RF models. Experimental results show that the proposed model outperforms the existing model in terms of accuracy. On the test set, the proposed model achieved an accuracy of 98.93% while the standalone CNN and RF with an accuracy of 91.95% and 68.78% respectively. This work demonstrates the effectiveness of the Integrated CNN-RF approach in bacterial genus classification and highlights its potential for future applications in microbial research

Keywords: Convolutional neural networks, Deep learning, Ensemble approach, Feature extraction, Hybrid model

Introduction

DNA sequence classification is a critical task in genomics research, with applications ranging from understanding microbial diversity to identifying potential disease markers. Accurate classification at the genus level provides valuable insights into the evolutionary relationships and functional characteristics of organisms. However, the complexity of DNA sequences poses challenges for traditional classification methods.

DNA sequences, composed of nucleotide bases adenine (A), cytosine (C), guanine (G), and thymine (T), encode vital genetic information. Analyzing these sequences provides a window into the genetic makeup of organisms, allowing researchers to decipher evolutionary relationships and functional attributes.

Classification of bacterial genus based on 16S ribosomal DNA (rDNA) sequences is a crucial task in microbiology and medical research. The 16S rDNA gene is widely used as a marker for identifying and classifying bacteria at the genus level, and accurate classification is important for understanding the diversity and distribution of bacteria, as well as for medical diagnoses and treatment

The identification and classification of DNA sequences, particularly at the genus level, presents inherent challenges due to the vast diversity and intricacies of genomic data. Traditional methods often struggle to capture the nuanced patterns essential for accurate classification.

In recent years, deep learning techniques, such as Convolutional Neural Networks (CNNs), have been applied to this problem with promising results.[1] CNNs have shown to be well suited for sequence analysis as they can learn complex patterns in the data. However, CNNs also have limitations, such as the need for large annotated datasets and a risk of overfitting.[2]

Various CNN models have been proposed for DNA sequence analysis tasks, such as protein structure prediction[3], gene expression analysis[4], and DNA-protein binding site prediction.[5] These models have shown to outperform traditional methods and have the potential to revolutionize the field of computational biology.

Random Forest (RF) is a popular machine-learning algorithm for classification and regression tasks. It

*Author for Correspondence
E-mail: madasamy.sornam@gmail.com

has been widely used in various domains, including biology, finance, and engineering. In the field of biology, RF has been applied to tasks such as protein function prediction[6], gene expression analysis[7], and drug discovery.[8] Random Forest has shown to be a reliable and effective tool for these tasks and has the potential to play a significant role in advancing biological research.

On the other hand, RF algorithms have been widely used in various classification problems and are well known for their accuracy and robustness.[9] However RF lacks the ability to capture complex patterns in sequences.

Current methods for DNA sequence classification include k-mer based approaches, Support Vector machines, Hidden Markov Models, RNN, and LSTM. While these approaches have shown promise, they may face challenges in handling the intricate patterns present in DNA sequences, motivating the exploration of advanced techniques.

Our study aims to overcome the limitations of traditional methods by proposing an Integrated CNN-RF model for DNA sequence classification. This hybrid approach leverages the strengths of CNNs for feature extraction and RF for robust classification.

Related Work

In contemporary genomics research, the classification of DNA sequences plays a crucial role in understanding microbial diversity, identifying disease markers, and deciphering evolutionary relationships. This paper addresses the intricate task of accurately classifying bacterial genera based on 16S ribosomal DNA (rDNA) sequences, a pivotal endeavour in microbiology and medical research. Traditional classification methods often struggle to capture the nuanced patterns within DNA sequences, prompting the exploration of advanced techniques. Recent advancements in deep learning, particularly CNNs, have shown promise in this domain. Building upon prior research, this paper delves into CNN-based approaches, gene expression data classification, taxonomy classification using 16S rRNA gene sequences, and k-mer-based methodologies proposed by various studies. The integration of insights from these diverse approaches culminates in the proposal of a hybrid model, combining CNNs and RF, with the aim of surmounting individual limitations and contributing to the progress of microbial research in DNA sequence classification.

Gunasekaran *et al*.[10] investigated the use of CNN, CNN-LSTM, and CNN-Bidirectional LSTM architectures, using Label and k-mer encoding, for DNA sequence classification. The models were evaluated using different classification metrics, and the results showed that the CNN and CNN-Bidirectional LSTM models with k-mer encoding achieved high accuracy, with 93.16% and 93.13% respectively, on test data. The paper mentions that for label encoding, testing accuracies are lower than training and validation accuracies. This could indicate overfitting on the training data and potentially poor generalization to unseen data.

Mathur *et al*.[11] present a novel approach for early disease detection by classifying DNA sequences from the NCBI database. This framework identifies patterns associated with various diseases, enabling rapid prediction when compared to samples from newly infected individuals. However, the size and complexity of FASTA-formatted DNA sequences pose challenges for direct feature extraction. To address this, the authors introduce a hot vector-based numerical representation where each nucleotide is assigned a binary value. This matrix is then fed into a traditional CNN for feature extraction. Their model outperforms seven other classifiers, including CNN, SVM, KNN, Decision Trees, and ANN, achieving an impressive accuracy of 93.9%. This process involves comparing a new patient's DNA sequence to the established disease patterns within the model, facilitating rapid prediction. While the NCBI database provides a vast collection, potential variations in disease presentation warrant further investigation into the model's generalizability. Additionally, analysing large datasets might require significant computational resources, necessitating exploration of optimization techniques for scalability. Despite these limitations, the framework signifies a promising step towards earlier disease diagnosis using DNA analysis, paving the way for future research on incorporating diverse data sources and advanced machine learning algorithms for even greater accuracy and clinical applicability

Mostavi *et al*.[12] explore the use of CNN models for classifying gene expression data into different cancer types or as normal. Three CNN models are presented and tested, including 1D-CNN, 2D-Vanilla-CNN, and 2D-Hybrid-CNN, using gene expression data from 10,340 samples of 33 cancer types and 713 normal tissues. The models showed high prediction accuracy

ranging from 93.9–95.0%. One of the models, the 1D-CNN, was further analyzed using a guided saliency technique, which identified 2090 cancer markers. The results were validated using well-known markers in breast cancer, such as GATA3 and ESR1. The 1D-CNN model was also used to predict breast cancer subtypes with an average accuracy of 88.42% among 5 subtypes. While the paper tests three CNN models, it does not delve deeply into optimizing their hyper parameters or analysing the extracted features.

Ziemski *et al*.[13] found that Naive Bayes Classifiers (NBCs) are still the best for classifying taxonomies based on the 16S rRNA gene, even when they tried to improve the RF and CNN classifiers with taxonomic weighting and encoding of sequence information. The RF and CNN classifiers performed better than NBCs only when taxonomic weighting information was used, but NBCs still performed better overall. The authors conclude that NBCs have already reached the limit of their ability to classify taxonomies based on short 16S rRNA gene sequences and further improvements will require advancements in technology and biology, or additional information beyond sequence data. However, they believe that CNNs and other methods still have potential and should be further optimized and tested for microbial classification.

Liu *et al*.[14] proposed a novel approach called k-mer Forest was proposed for predicting putative regulatory sequences based solely on k-mer features without any prior knowledge. The k-mer Forest model outperformed other methods using general genomic sequence information and effectively distinguished accessible chromatin regions from the pure genomic context. The identified sequence features could serve as potential functional elements and candidates for consensus motifs like transcription factor binding sites. The k-msa Forest model, which combines k-mer frequency with sequence conservation, showed even better performance. The results indicate that predictive sequence features may be more evolutionarily conserved. One drawback of the k-mer Forest model is that it can be sensitive to the choice of k-mer length. If the k-mer length is too short, the model may not be able to capture enough information about the sequence. If the k-mer length is too long, the model may be overfitting the training data.

A new gene prediction method was introduced by Silva *et al*.[15] using the RF classifier. The proposed model outperformed the existing gene prediction tools such as Prodigal and FragGeneScan, achieving 27% and 20% better results respectively based on the AUC values in an independent test set. The study emphasized the significance of selecting the right features in gene prediction, and the use of k-mer counting features was considered to be the key component in developing robust gene predictors.

Miah[19] proposes a methodology for DNA sequence classification using deep learning models, specifically CNNs with additional components like Long Short-Term Memory (LSTM) and bidirectional LSTM. The DNA sequences, obtained from the National Center for Biotechnology Information (NCBI), are transformed into numerical representations through encoding methods, including label encoding and k-mer encoding. To address the imbalanced dataset, the Adaptive Synthetic Sampling Approach (ADASYN) is utilized. The paper employs a Genetic Algorithm (GA) for optimizing CNN layers to enhance classification accuracy. Three deep learning models are tested and compared using different encoding methods. Results show that the incorporation of the GA optimization layer significantly improves accuracy, with label encoding consistently outperforming other methods. The CNN model demonstrates higher accuracy than CNN-LSTM and CNN-LSTM bidirectional models, emphasizing the importance of model architecture. The study suggests future work involving domain-specific information and exploration of alternative architectures and optimization techniques.

Abbas *et al*.[20] presented an innovative model named m5C-pred, utilizing five feature encoding techniques and SHapley Additive exPlanations for optimal feature selection. Leveraging XGBoost with hyperparameter optimization through Optuna, m5C-pred outperforms existing methods, offering a precise and effective solution for identifying RNA m5C methylation sites across multiple species. Rehman *et al*.[21] introduced i6mA-Caps[21], a novel computational tool employing a CapsuleNet-based framework for accurate identification of DNA N6-methyladenine sites. Utilizing a single encoding scheme, convolution and capsule layers extract hierarchical features, achieving impressive accuracy rates of 96.71%, 94%, and 86.83% on independent datasets from Rosaceae, Rice, and Arabidopsis thaliana genomes, surpassing existing state-of-the-art methods. Rehman *et al*.[22] unveiled DL-m6A22, an innovative deep learning-based tool designed for the

efficient identification of N6-methyladenosine (m6A) sites in mammals. The tool leverages three distinct encoding schemes and a multi-layered neural network architecture, resulting in superior performance on both tissue-specific and full transcript datasets. DL-m6A22 has demonstrated its capability to outperform existing tools, highlighting its effectiveness in m6A site identification.

Hossain *et al*.[23], delves into deep learning's application in taxonomic categorization of DNA sequences, focusing on two architectures: Stacked Convolutional Autoencoder (SCAE) with Multilabel Extreme Learning Machine (MLELM) and Variational Convolutional Autoencoder (VCAE) with MLELM. Highlighting the significance of clade labels and multi-label approaches, the study emphasizes improved accuracy, particularly with the VCAE-MLELM model, shaping advancements in DNA sequence classification.

Materials and Methods

Dataset Description

GenBank is a public, centralized repository of DNA sequences maintained by the National Center for Biotechnology Information (NCBI). It contains a large number of 16S rDNA sequences obtained from a wide range of sources, including bacteria, archaea, and other microorganisms. The 16S rDNA sequences in GenBank can be used for a variety of purposes, including identifying and characterizing bacteria and archaea in environmental and medical samples, and determining the evolutionary relationships between different species.

This paper focuses on predicting the genus of bacteria using a conserved region of the sequence. There are totally 371 genera represented in this data set and hence this paper deals with multiclass classification problem. In this dataset, there are many classes with less than 50 samples, which may not be suitable for use as a training set for CNN and RF. This is because a small number of samples can lead to overfitting, where the model becomes too specialized to the training data and is unable to generalize to new, unseen data. Therefore, sequences with class labels greater than 50 samples alone were taken into consideration for this research. There are totally 12,508 sequences with the length ranging from 459 to 1833 bases in length with the majority around 1500 bases, resulting in 60 classes. Hence this paper deals with multiclass classification problem.

The file format of the DNA sequence is in FASTA format, which is a text-based format that stores sequences in a compact format, with each record starting with a header line and followed by the sequence data on multiple lines.

Convolutional Neural Networks

Convolutional Neural Networks are a pivotal technology in artificial neural networks, originally designed for computer vision tasks, but their adaptability extends to various domains. In the realm of DNA sequence analysis, CNNs serve as powerful tools for tasks such as classification, gene prediction, and secondary structure prediction. The network architecture involves processing DNA sequences represented as one-hot encoded vectors through layers that learn local patterns, apply non-linear activation functions, and utilize pooling operations to enhance robustness. With the ability to automatically extract relevant features without manual intervention, CNNs have demonstrated superior performance compared to traditional methods in diverse DNA sequence analysis tasks. The training process, involving back propagation and optimization algorithms, enables the network to learn and make predictions on new DNA sequences. This algorithmic framework not only facilitates accurate predictions but also showcases the versatility of CNNs in deciphering intricate biological information encoded in DNA sequences.

Random Forest

Random Forest stands out as a widely employed machine-learning algorithm for the classification of DNA sequences. Its operational framework involves constructing an ensemble of decision trees, each trained on a randomly chosen subset of the data. This collective intelligence is then harnessed to yield a final prediction. The algorithm's classification process unfolds through distinct stages, starting with the input of DNA sequences labelled with corresponding classes. Pre-processing, which may involve converting sequences into numerical representations, precedes the training phase. During training, decision trees are built using random subsets, employing features for splits based on measures like Gini impurity or information gain. Prediction involves passing new DNA sequences through each tree, with the final classification determined through a majority or weighted vote. Evaluation metrics such as accuracy, precision, recall, and F1-score contribute to

assessing the RF's performance on a held-out test set. This methodology underscores the algorithm's efficacy in deciphering complex patterns within DNA sequences for accurate classification.

Proposed Method

Two approaches are proposed to perform the process of classification.

Proposed Method-1: Integrated CNN and RF

Using CNNs to Extract Features from the Input Data, and using them as Input to a RF Model

In this research, the integration of CNN with RF is proposed for the classification of 16S rDNA sequences at the genus level. In this approach, CNN is used for feature extraction and RF for classification. The CNN takes the raw DNA sequence data as input and automatically learns the hidden features or representations of the sequences through multiple convolutional and pooling layers. The output of the CNN, called the feature maps, is then used as input for the RF classifier.

The RF classifier, as an ensemble of decision trees, can effectively handle high-dimensional and non-linear data, and provide robust and stable predictions. Using the feature maps extracted by the CNN, the RF can learn the relationships between the hidden features and the class labels and make the final predictions. In this approach, the CNN provides an effective way to extract meaningful features from the raw DNA sequence data, while the RF provides a robust and stable classifier. The combination of the two algorithms can enhance the performance and generalization ability of the DNA sequence analysis task.

The Algorithm 1 outlines a integrated CNN-RF model for DNA sequence analysis, involving training a CNN on one-hot encoded sequences, extracting features, training an RF classifier, and evaluating the model's accuracy on test data.

Algorithm 1: Integrated CNN − RF Model for DNA Sequence Analysis

Input:

- DNA Sequence Data X with shape (m, n), where m is the number of samples and n is the number of features.
- One − hot encoded representations X_{oh}.
- Training and test sets $X_{train_{oh}}, X_{test_{oh}}, y_{train}, y_{test}$.

Output:

- Predicted labels for the test set y_{pred}.
- Accuracy of the model.

Begin

Step 1. Initialize CNN and RF Components:

- Initialize CNN with learnable weights W, biases b, and activation functions (ReLU).
- Initialize RF as an ensemble of decision trees.

Step 2. Training CNN:

- Input: $X_{train_{oh}}$
- Forward Pass:
- $z = X_{train_{oh}} * W + b$
- $a = \mathrm{ReLU}(z)$
- Compute Loss:
- $L = \mathrm{CrossEntropyLoss}(a, y_{train})$
- Backpropagation:
- $\frac{\partial L}{\partial W}, \frac{\partial L}{\partial b} = \mathrm{Backpropagate}(L)$
- Update Weights:
- $W, b = \text{Optimization Algorithm}\left(W, b, \frac{\partial L}{\partial W}, \frac{\partial L}{\partial b}\right)$

Step 3. Feature Extraction Using CNN:

- Extract features:

$$f_j = \text{Flatten}\left(\text{Pooling}\left(\text{Activation}\left(\text{Convolution}\left(X_{train_{oh}}, W_j\right)\right)\right)\right)$$

Step 4. Training RF:

- Input: f_{train}, y_{train}
- Train RF classifier:
- $RF = \mathrm{TrainRF}(f_{train}, y_{train})$

Step 5. Prediction using RF:

- Input: $X_{test_{oh}}$
- Predict labels for the test set:
- $y_{pred} = \text{RF. Predict}\left(X_{test_{oh}}\right)$

Step 6. Evaluate Performance:

- Compute accuracy:
- $\text{Accuracy} = \mathrm{mean}\left(y_{pred} == y_{test}\right)$
- Output the accuracy.

End

Let $X_{train_{oh}}$ be a $(m\ x\ n)$ matrix of one-hot encoded DNA sequences, where m is the number of samples and n is the sequence length. We apply a linear transformation using convolutional weights W(3D tensor) and biases b (vector): $z = X_{train_{oh}} @ W + b$ This captures local interactions between nucleotides within the sequence. We introduce non-linearity with an activation function like ReLU: $a = ReLU(z)$. This adds expressiveness to the model and allows it to learn complex patterns. We measure the difference between predicted probabilities and true labels using the CrossEntropyLoss:

$L = CrossEntropyLoss(a, y_{train})$. Here, a is the activation output for each sample, and y_train is the corresponding ground-truth label vector. Minimizing this loss encourages the CNN to learn features that discriminate between different classes. We compute the gradients of the loss with respect to weights and biases: $\partial L/\partial W, \partial L/\partial b$. These gradients guide the updates during the next step. An optimization algorithm like Adam uses the gradients to update the weights and biases: $W = W - \eta * \partial L/\partial W, b = b - \eta * \partial L/\partial b$ where $\eta$is the learning rate. This iterative process refines the CNN's ability to extract relevant features.

The activations (a) are further processed through additional layers:

- Convolution: Extracts specific features using additional convolution kernels.
- Pooling: Reduces dimensionality by summarizing information from local areas.
- Activation: Introduces further non-linearity.
- Flatten: Reshapes the output into a single-dimensional vector for each sample.

These vectors (f_j) serve as compact representations of the extracted features.

Features extracted from test data ($\boldsymbol{X_{test_{oh}}}$) are fed into the trained RF. Each tree in the RF predicts a class label, and the final prediction integrates information from all trees using majority vote or another combination rule. Accuracy is calculated as the proportion of correctly predicted labels: $Accuracy = mean(y_{pred} == y_{test})$. This integrated approach combines the spatial feature learning of the CNN with the robust classification of the RF, capturing complex relationships within DNA sequences while providing interpretable predictions.

The original architecture of CNN is shown in Fig. 1 whereas an integrated CNN-RF approach is highlighted in Fig. 2, where convolutional layers extract DNA sequence features that feed into a RF for robust class prediction. This hybrid architecture aims to unlock deeper understanding and improved accuracy in analysing DNA sequences.

Proposed Method-2: Ensemble of CNN and RF models

Using CNNs and RF as two separate models, and then combining their predictions

In this approach, a CNN predicts outcomes from input data, and a RF makes predictions on the same data. The final prediction is a combined result, where both models contribute, and their predictions are averaged with different weights.

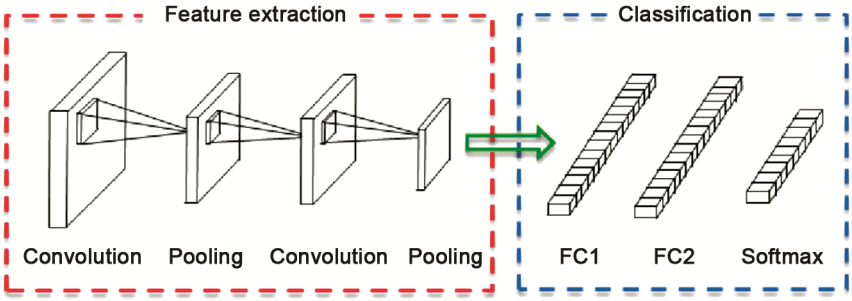

Fig. 1 — Original architecture of CNN

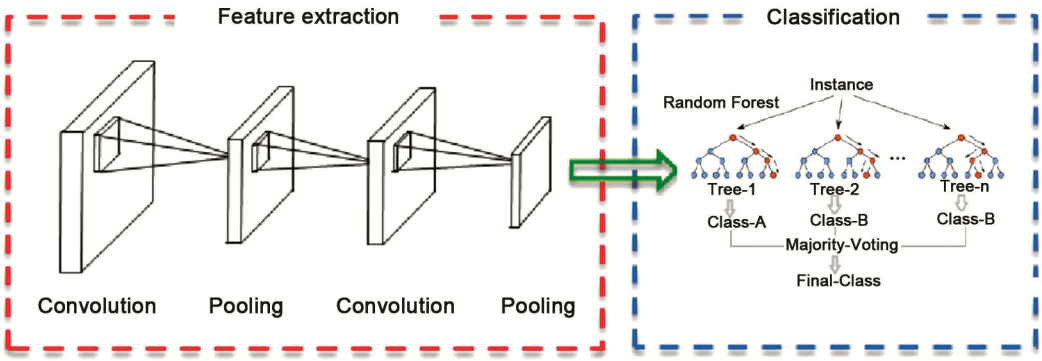

Fig. 2 — Proposed architecture of integrated CNN – RF approach

Algorithm 2: **Ensemble of CNN − RF**

Input:

− Training dataset X with N samples and K features.

− Corresponding class labels Y with N samples and C classes.

Output:

− Final class predictions Y_{hat}.

− Metrics: Accuracy, Precision, Recall, F1 − score.

Begin

Step 1: Initialize CNN and RF Classifiers

- Initialize CNN classifier f_{cnn} with learnable weights W_{cnn}, biases b_{cnn}, and activation functions (ReLU).
- Initialize RF classifier f_{rf} as an ensemble of decision trees.

Step 2: Train CNN and RF

Train CNN classifier:

- $Y_{cnn} = f_{cnn(X;\, W_{cnn}, b_{cnn})}$
- Forward Pass:
- $z_{cnn} = X * W_{cnn} + b_{cnn}$
- $a_{cnn} = \mathrm{ReLU}(z_{cnn})$
- Compute Loss:
- $L_{cnn} = \mathrm{CrossEntropyLoss}(a_{cnn}, Y)$
- Backpropagation:
- $\frac{\partial L_{cnn}}{\partial W_{cnn}}, \frac{\partial L_{cnn}}{\partial b_{cnn}} = \mathrm{Backpropagate}(L_{cnn})$
- Update Weights:
- $W_{cnn}, b_{cnn} = \mathrm{OptimizationAlgorithm}\left(W_{cnn}, b_{cnn}, \frac{\partial L_{cnn}}{\partial W_{cnn}}, \frac{\partial L_{cnn}}{\partial b_{cnn}}\right)$

Train RF classifier:

- $Y_{rf} = f_{rf(X)}$
- Randomly select subsets of features for each tree:
- $\mathrm{features}_{subset} = \mathrm{RandomSubset}(\text{all features, subset size})$
- For each tree:
- Randomly select samples with replacement:
- $\mathrm{samples}_{subset} = \mathrm{RandomSubset}(\text{all samples, subset size})$
- Calculate impurity and determine the best split for each node using $\mathrm{features}_{subset}$ and $\mathrm{samples}_{subset}$.
- Build the decision tree structure.
- Repeat for a predefined number of trees.

Step 3: Assign Weights

- Assign weights based on performance:
- $w_{cnn} = 1 - w_{rf} = p$ where $0 \le p \le 1$.

Step 4: Calculate Weighted Sum

- Calculate the weighted sum for each DNA sequence:
- $Y_{final} = w_{cnn} * Y_{cnn} + w_{rf} * Y_{rf}$

Step 5: Assign Final Class Label

- Use threshold function g to assign the final class label:

$Y_{hat} = g(Y_{final}) = \left\{c_j \text{if } Y_{final_j} \ge t, \text{for } j = 1, \ldots, C\right\}$ where t is the threshold value.

Step 6: Evaluate Performance

- Compute metrics:
- $\mathrm{Accuracy} = \mathrm{accuracy}(Y, Y_{hat})$
- $\mathrm{Precision} = \mathrm{precision}(Y, Y_{hat})$
- $\mathrm{Recall} = \mathrm{recall}(Y, Y_{hat})$
- $F1 - \mathrm{score} = F1 - \mathrm{score}(Y, Y_{hat})$

End

Algorithm 2 presents an integrated CNN-RF model for DNA sequence analysis. It commences by initializing the CNN and RF components, configuring the CNN with learnable weights, biases, and ReLU activation, and the RF as a decision tree ensemble. The algorithm proceeds with training the CNN on the one-hot encoded DNA sequences, optimizing the weights through backpropagation, and then extracting features through convolutional layers. Simultaneously, the RF is trained on the same dataset, employing subsets of features and samples for each tree. The CNN and RF predictions are combined using weighted averaging based on their respective performance, introducing weights w_{cnn} and w_{rf} with $0 \le p \le 1$. The final class labels are determined by thresholding the weighted sum of predictions. The algorithm concludes with evaluating the model's performance through metrics like accuracy, precision, recall, and F1-score. This approach harnesses the complementary strengths of CNNs and RFs for improved DNA sequence classification.

Results

The proposed pipeline is implemented using the below mentioned platform:

Hardware

• Processor: AMD Ryzen 5 4600H with Radeon Graphics (3.00 GHz)

• RAM: 8 GB

Software

• Operating System: Windows 10

• Machine Learning Framework: scikit-learn

• Programming Language: Python

The pre-processing stage stands as a pivotal phase in the DNA sequence classification task, essential for preparing raw DNA sequences for analysis by the model. DNA sequences, conventionally expressed as strings of four nucleotides (A, C, G, T), necessitate encoding into numerical representations to serve as input for machine-learning models. A prevalent encoding method is one-hot encoding, where a binary vector of length 4 denotes each nucleotide. Addressing sequences of varying lengths involves padding with a special symbol to a fixed length, ensuring uniformity for consistent model processing. To accommodate distinct scales of input data, the encoded sequences undergo normalization to achieve zero mean and unit variance.

Encoded and padded DNA sequences form the training data for a dedicated CNN model designed to classify at the genus level. This model underwent training with 20,096 sequences from a balanced dataset, and validation utilized 25% of testing sequences, totalling 5024 sequences. Padded DNA sequences, measuring 1833 base pairs, were used for training, focusing on the classification into 2893 genus classes. The CNN model architecture included three convolutional layers and one fully connected layer, featuring distinct activation functions like ReLU in the inner layers and a softmax activation function in the outer layer.

The choice of the number of convolution layers, fully connected layers, activation functions (specifically ReLU in inner layers and softmax in the outer layer) is a crucial aspect of the model architecture. These design decisions are made based on principles of deep learning and domain knowledge.

Number of Convolution Layers

Convolutional layers are crucial in capturing local patterns and features in the input data. The depth of the network, i.e., the number of convolution layers, determines its ability to learn complex hierarchical features.

In this method, three convolution layers are chosen to extract features from the DNA sequences.

The specific number of layers might be determined through experimentation and considering the trade-off between model complexity and performance. More layers allow the network to capture patterns that are more complex but can also lead to overfitting if not controlled.

Fully Connected Layers

Fully connected layers, also known as dense layers, are often added towards the end of the neural network to perform high-level feature learning and make final predictions.

In the proposed method 1, one fully connected layer takes the flattened feature maps from the convolution layers and transforms them to produce the final output. The choice of one fully connected layer is a reasonable starting point for many classification tasks, but tasks that are more complex might require multiple fully connected layers for deep feature learning.

Activation Functions (ReLU and Softmax)

ReLU (Rectified Linear Unit): ReLU is used in the inner layers of the CNN. ReLU is a common activation function for convolution layers because it introduces non-linearity into the model and helps it learn complex features. It is computationally efficient and mitigates the vanishing gradient problem. ReLU is preferred over other activation functions like sigmoid and tanh for deep networks.

Softmax: Softmax is used in the outer layer of the network for multi-class classification. Softmax is particularly well-suited for classification tasks where the goal is to assign an input to one of multiple classes. It converts the raw scores (logits) into a probability distribution over the classes. This makes it ideal for producing class probabilities for each class and selecting the class with the highest probability as the final prediction.

Adagrad, SGD, and Adam stand out for their effectiveness in diverse aspects of DNA sequence classification. Adagrad adjusts to the varied nature of sequences, SGD navigates through intricate patterns, and Adam integrates adaptive learning rates and momentum for efficient learning. These optimizers were chosen for their proven ability to address the challenges of DNA sequence classification, contributing to the method's robust classification performance.

To optimize the model, four distinct optimizers (Adagrad, SGD, Rmsprop and Adam) were employed. The model's performance, assessed through a confusion matrix and detailed in Table 1, demonstrated an accuracy of 91.95%, precision of 91.55%, recall of 91.8%, and an F1-score of 91.8% on the validation dataset after 50 training epochs completed in 12.4 minutes. Notably, the

Table 1 — Results of CNN

Optimizer	Epochs	Precision	Recall	F1-Score	Accuracy	Time (minutes)
Adagrad	10	67.23	66.54	63.58	67.23	2.83
	20	70.25	72.36	71.96	88.93	3.5
	30	72.69	75.68	74.65	89.54	5
	40	81.6	82.95	80.24	89.91	7
	50	89.83	89.47	91.39	90.78	12.12
Sgd	10	88.64	86.24	85.32	90.98	2.68
	20	89.32	88.21	87.91	91.07	4.17
	30	90.54	90.27	90.47	91.46	5.4
	40	91	90.52	90.53	91.52	7.68
	50	91.72	90.97	90	91.64	12.82
Rmsprop	10	75.21	73.91	74.52	83.08	1.72
	20	76	75.68	77	88.20	4.38
	30	76.59	75.96	76.5	88.57	6.53
	40	80.25	80.15	79.24	89.23	9.77
	50	81.73	80.67	81.39	89.56	12.37
Adam	10	89.28	88.29	90.37	90.78	1.65
	20	90.35	89.37	91.63	91.48	3.2
	30	91.49	89.82	91	91.69	5.93
	40	91.35	91	91.27	91.64	7.82
	50	**91.55**	**91.8**	**91.88**	**91.95**	**12.4**

Table 2 — Results of random forest

No of Trees	Accuracy	Sensitivity	Specificity	F1-Score	Time (Seconds)
100	67.78	58	79	64	15
200	68.34	58	81	65	28
300	68.59	58	81	65	35
400	68.71	59	81	66	43
500	**68.78**	**59**	**82**	**66**	**58**

Adam optimizer emerged as the most effective among them.

The RF model underwent training on a balanced dataset, employing 100, 200, 300, 400, and 500 decision trees. Gini impurity served as the criterion for evaluating split quality. However, the model's performance, as indicated by the confusion matrix, demonstrated no improvement even after constructing 500 decision trees. This suggests that the model might not have effectively captured relevant features for DNA sequence classification. The performance metrics, detailed in Table 2 through the confusion matrix, highlight the need for further research to identify a more effective approach.

In the proposed ensemble model, the predictions from the CNN and RF models were combined using the weighted average method, wherein each model was assigned a weight based on its performance on the validation dataset. Due to the superior performance of the CNN, a weight of 0.7 was assigned to CNN predictions, while RF predictions received a weight of 0.3. This weighting aimed to emphasize the model with better performance while considering contributions from both models. Despite these considerations, the ensemble of CNN and RF, employing the weighted average method, exhibited suboptimal performance on the validation dataset. The accuracy was 59%, precision was 43%, recall was 52%, and the F1-score was 43%. These values were lower compared to the results obtained from either the standalone CNN or RF models, indicating challenges in effectively harnessing the strengths of both models for accurate DNA sequence classification. Further exploration is warranted to improve the ensemble model's efficacy.

Hence, an alternative approach, the Integrated CNN-RF, is introduced to capitalize on the respective strengths of both models. While CNNs excel in capturing high-level features, RF models are adept at handling complex relationships between features,

leading to accurate predictions. To extract features from the pooling layer, a new model is devised, encompassing all layers of the original CNN up to the desired pooling layer. Input data is then passed through this model, and the activations of the pooling layer, representing the learned features, are obtained. This experiment employs an input layer, three convolutional layers, and three max-pool layers, as diagrammatically depicted in Fig. 3. The output features are flattened and utilized as input for the RF model. The performance of the Integrated CNN and RF model is evaluated using a confusion matrix, revealing a commendable accuracy of 98.93%, precision of 98.57%, recall of 98.72%, and an F1-score of 98.79% on the validation dataset, achieved with the training of only 100 decision trees. A comparative analysis of the Integrated CNN+RF and the Ensemble CNN and RF, using the same data, is presented in Table 3.

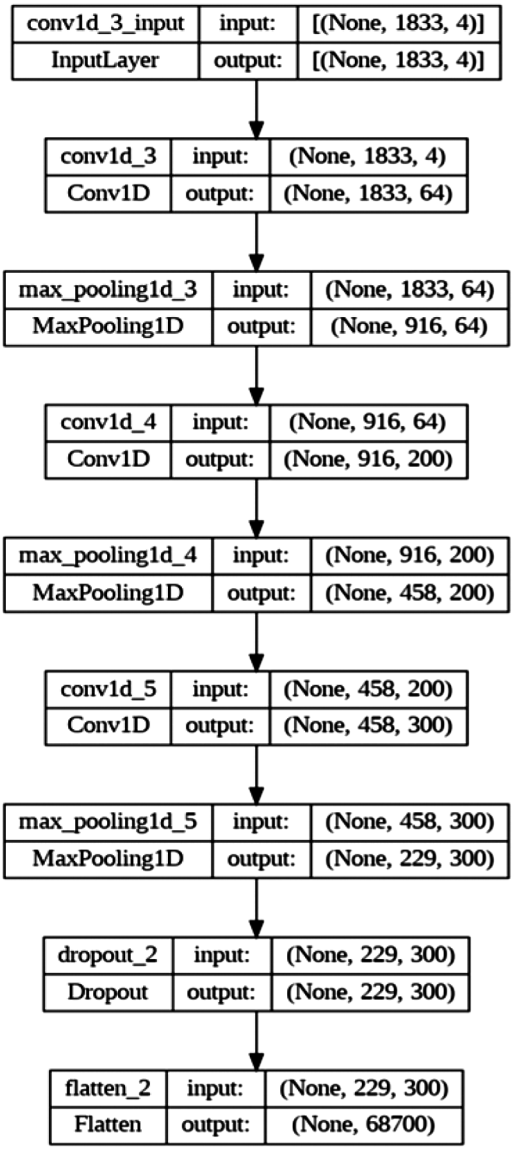

Fig. 3 — Visual Representation of layers in Integrated CNN-RF

Furthermore, the proposed model provides significant benefits over standalone CNN, standalone RF, and the ensemble of both. By using the features extracted from the CNN component to train the RF component, the CNN component can have fewer layers and still achieve high performance, while the RF component can use a smaller number of decision trees while still achieving comparable or better performance compared to standalone RF with a larger number of decision trees.

The advantages of the proposed Integrated CNN-RF model are as follows:

• **Improved Performance:** By combining the strengths of both CNNs and RF, the Integrated CNN-RF model can achieve better performance compared to standalone models, making it an effiective solution for DNA sequence classification.

• **Computational Efficiency:** By reducing the number of convolutional layers in the CNN component and the number of decision trees in the RF component, the Integrated CNN-RF model can be more computationally efficient compared to standalone models, reducing the computational resources required to train and use the model.

• **Reduced Overfitting:** By reducing the number of parameters in the model, the Integrated CNN-RF model can mitigate overfitting, a common issue in deep neural networks and decision tree-based models, leading to better generalization performance.

The disadvantages of our method are as follows:

• **Increased Complexity:** The integration of two models can increase the complexity of the model,

Table 3 — Results of Integrated CNN + RF and Ensemble of CNN and RF

Proposed Methodology	Metrics	Performance
Integrated CNN+ RF	Accuracy	**98.93**
	Precision	**98.57**
	Recall	**98.72**
	F1-Score	**98.79**
	Time (Seconds)	**480**
Ensemble CNN and RF	Accuracy	59
	Precision	43
	Recall	52
	F1-Score	43
	Time (Seconds)	79

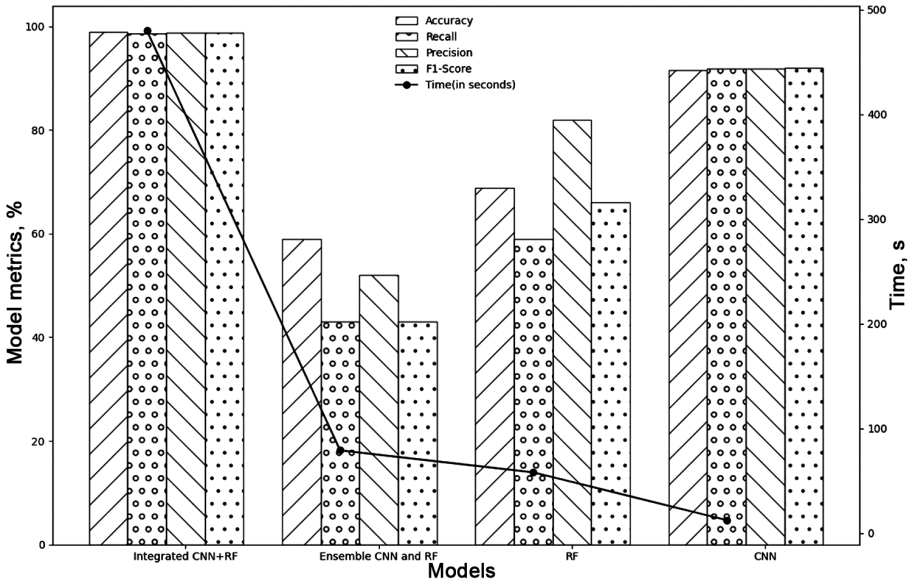

Fig. 4 — Performance Comparison of all the models

Table 4 — Comprehensive analysis of the related works with the proposed work

Task	Model Used	Input data Type	Performance
Classifying DNA sequence	CNN[1]	Hepatitis B Virus	95.6%
Classifying DNA sequence	k-mer CNN[10]	COVID sequences	97.9%
Classifying DNA sequence	One hot vector matrix feature extraction[11]	FASTA based DNA Sequences	93.9%
Cancer type prediction based on gene expression profiles	1D-CNN[12]	Cancer RNA-Seq data	88.42%
Taxonomic classification of RNA sequence	Naïve Bayes[13]	16S rRNA sequence	F-Measure of NBC is more than Random Forest
Gene Prediction	geneRFinder[15]	DNA sequences	93.4%
Classifying DNA sequence	CNN[16]	DNA sequence – Histone Proteins	88.99%
Classification of Image generated from the DNA sequence	CNN[17]	DNA sequences of Hepatitis C Virus	80.8%
Classifying DNA sequence	k-mer based CNN[18]	Splice , Promoter and H3 dataset	96.85% , 95.45% ,83.39%
Classify Viral DNA sequence	GA with CNN[19]	COVID , MERS , SARS, Influenza, Hepatitis, Dengue	96.8%
Classifying DNA sequence at genus level (Proposed Method)	Integrated CNN-RF	16S rDNA Sequences	98.93%

making it harder to understand, implement, and debug compared to standalone models.

• **Increased Training Time:** Integrating two models can increase the training time compared to training either model in isolation, which can be a disadvantage when working with large datasets.

The Fig. 4 depicts the performance comparison of different methods in terms of accuracy, precision, recall, F1-score, and execution time. Among the methods evaluated, the Integrated CNN+RF approach stands out with the highest accuracy, precision, recall, and F1-score. Although it requires a longer execution time of 120 seconds, it demonstrates superior classification performance compared to the other methods. This highlights the trade-o between computational time and classification accuracy, where the Integrated CNN+RF method prioritizes accuracy and delivers reliable results despite the increased time requirement. The proposed work is also compared with related works and the results are tabulated in Table 4.

Time Complexity

• **Pre-processing:** The time complexity of the pre-processing step is $O(NK)$, where N is the number of samples and K is the number of features. This is

Table 5 — Time Complexity of Integrated CNN+RF

Step	Time Complexity
Pre-processing	$O(NK)$
CNN feature extraction	$O(NMK)$
RF classification	$O(NMlog_2(T))$
Proposed Model	$O(NK + NMK + NMlog_2(T))$

because the pre-processing step involves transforming the raw DNA sequence data into a suitable format, such as one-hot encoding, which requires $O(NK)$ operations.

• **CNN feature extraction:** The time complexity of the CNN feature extraction step depends on the architecture of the CNN model, but it typically involves multiple convolutional and pooling operations, which have a time complexity of $O(NM)$, where N is the number of samples, M is the number of learned features, and the time complexity of each operation is $O(K)$. Therefore, the total time complexity of the CNN feature extraction step is $O(NMK)$.

• **RF classification:** The time complexity of the RF classification step is $O(NMlog_2(T))$,where N is the number of samples, M is the number of learned features, and T is the number of trees in the RF. This is because the RF classifier involves training T decision trees on the learned features Z, which requires $O(NMlog_2(T))$ operations.

The Table 5 provides the time complexity for each step of the classification algorithm: Pre-processing ($O(NK)$), CNN feature extraction ($O(NMK)$), and RF classification ($O(NMlog_2(T))$). The total time complexity is the sum of these complexities: $O(NK + NMK + NMlog_2(T))$.

Conclusions

In conclusion, the proposed Integrated CNN-RF method for DNA sequence classification has demonstrated superior performance compared to other state-of-the-art methods. The integration of CNN and RF models leverages the strengths of both models and outperforms standalone CNN, standalone RF and the ensemble of both models in terms of accuracy and efficiency. The reduction in the number of convolutional layers and number of trees in RF also has improved the performance of the model and reduced the risk of overfitting. The proposed method has shown promising results in this field and it can be considered as a valuable contribution to the existing literature. One disadvantage of the integrated CNN-RF approach in our proposed model is the increased complexity and computational cost. Combining CNN with RF requires training and maintaining both models, which can be resource-intensive. Additionally, integrating these two models may introduce additional hyper parameters and complexities in the overall model, making it more challenging to tune and interpret.

References

1 Kassim N & Abdullah D A, Classification of DNA sequences using convolutional neural network approach, *UTM Comput Proc Innov Comput Technol Appl* (2018) 1–6, https://api.semanticscholar.org/CorpusID:195293334.

2 Poernomo A & Kang D K, Biased dropout and crossmap dropout: learning towards effective dropout regularization in convolutional neural network, *Neural Netw*, **104** (2018) 60–67, doi: https://doi.org/10.1016/j.neunet.2018.03.016.

3 Asgari E, Poerner N, McHardy A C & Mofrad M R K, DeepPrime2Sec: deep learning for protein secondary structure prediction from the primary sequences, *Bio Rxiv*, (2019) 705426, doi: https://doi.org/10.1101/705426.

4 Shon H S, Yi Y G, Kim K O, Cha E J & Kim K A, Classification of stomach cancer gene expression data using CNN algorithm of deep learning, *J Biomed Transl Res*, **20(1)** (2019) 15–20, doi: https://doi.org/10.12729/jbtr.2019. 20.1.015.

5 Zeng H, Edwards M D, Liu G & Gifford D K, Convolutional neural network architectures for predicting DNA-protein binding, *J Bioinform*, **32(12)** (2016) i121–i127, doi: https://doi.org/10.1093/bioinformatics/btw255.

6 Hakala K, Kaewphan S, Björne J, Mehryary F, Moen H, Tolvanen M, Salakoski T & Ginter F, Neural network and random forest models in protein function prediction, *IEEE/ACM Trans Comput Biol*, **19(3)** (2020) 1772–1781, doi: https://doi.org/10.1109/TCBB.2020.3044230.

7 Ram M, Najafi A & Shakeri M T, Classification and biomarker genes selection for cancer gene expression data using random forest, *Iran J Pathol*, **12(4)** (2017) 339.

8 Cano G, Garcia-Rodriguez J, Garcia-Garcia A, Perez-Sanchez H, Benediktsson, J A, Thapa A & Barr A, Automatic selection of molecular descriptors using random forest: Application to drug discovery, *Expert Syst Appl*, **72** (2017) 151–159, doi: https://doi.org/10.1016/j.eswa.2016. 12.008.

9 Cutler A, Cutler D R & Stevens J R, Random Forests, in *Ensemble Machine Learning* (Springer) 2012, 157–176 doi: https://doi.org/10.1007/978-1-4419-9326-7_5.

10 Gunasekaran H, Ramalakshmi K, Rex Macedo Arokiaraj A, Deepa Kanmani S, Venkatesan C & Suresh Gnana Dhas C, Analysis of DNA sequence classification using CNN and hybrid models, *Comput Math Methods Med*, **2021** (2021), doi: https://doi.org/10.1155/2021/1835056.

11 Mathur G, Pandey A & Goyal S, A comprehensive tool for rapid and accurate prediction of disease using DNA sequence classifier, *J Ambient Intell Humaniz Comput*, **14(10)** (2023) 13869–13885, doi: https://doi.org/10.1007/s12652-022-04099-y.

12 Mostavi M, Chiu Y C, Huang Y & Chen Y, Convolutional neural network models for cancer type prediction based on

gene expression, *BMC Med Genomics*, **13** (2020) 1–13, doi: https://doi.org/10.1186/s12920-020-0677-2.

13 Ziemsk M, Wisanwanichthan T, Bokulich N A & Kaehler B D, Beating naive bayes at taxonomic classification of 16S rRNA gene sequences, *Front Microbial*, **12** (2021) 644487, doi: https://doi.org/10.3389/fmicb.2021.644487.

14 Liu Q, Gan M & Jiang R, A sequence-based method to predict the impact of regulatory variants using random forest, *BMC Syst Biol*, **11(2)** (2017) 1–9, doi: https://doi.org/10.1186/s12918-017-0389-1.

15 Silva R, Padovani K, Góes F & Alves R C, A random forest classifier for prokaryotes gene prediction, in *8th Brazilian Conf Intell Syst* (IEEE) 2019, 545–550, doi: https://doi.org/10.1109/BRACIS.2019.00101.

16 Nguyen N G, Tran V A, Phan D, Lumbanraja F R, Faisal M R, Abapihi B & Satou K, DNA sequence classification by convolutional neural network, *J Biomed Eng*, **9(5)** (2016), 280–286, doi: https://doi.org/10.4236/jbise.2016.95021.

17 Somodevilla M R L & Rossainz M, DNA sequence recognition using image representation, *Res Comput Sci*, **148** (2019) 105–114, doi: https://doi.org/10.13053/rcs-148-3-9.

18 Akkaya U M & Kalkan H, Classification of DNA sequences with k-mers based vector representations, in *Innov Intell Syst Appl Conf* (IEEE) 2021, 1–5, doi: https://doi.org/10.1109/ASYU52992.2021.9599084.

19 Miah J, *Enhancing Viral DNA Sequence Classification using Hybrid Deep Learning Models and Genetic Algorithm Optimization* (SSRN) 2024, 1–10, doi: https://ssrn.com/abstract=4692259.

20 Abbas Z, ur Rehman M, Tayara H, Zou Q & Chong K T, XGBoost Framework with feature selection for the prediction of RNA N5-methylcytosine sites, *Mol Ther*, **31(8)** (2023) 2543–2551, doi: https://doi.org/10.1016/j.ymthe.2023.05.016.

21 Rehman M U, Tayara H, Zou Q & Chong K T, i6mA-Caps: A CapsuleNet-based framework for identifying DNA N6-methyladenine sites, *J Bioinform*, **38(16)** (2022) 3885–3891, doi: https://doi.org/10.1093/bioinformatics/btac434.

22 Rehman M U, Tayara H & Chong K T, DL-m6A: Identification of N6-methyladenosine sites in mammals using deep learning based on different encoding schemes, *IEEE/ACM Trans Comput Biol*, **20(2)** (2022) 904–911, doi: https://doi.org/10.1109/TCBB.2022.3192572.

23 Hossain P S, Kim K, Uddin J, Samad M A & Choi K, Enhancing taxonomic categorization of DNA sequences with deep learning: A multi-label approach, *Bioengineering*, **10(11)** (2023) 1293 doi: https://doi.org/10.3390/bioengineering10111293.