Detection of Bacterial Signatures in Genomic DNA Sequences using Hybrid ML-DL Pipeline

Gowtham D

III M.Sc SS

Department of Software Systems And Aiml

Sri Krishna Arts And Science College

Coimbatore, India

gowthamd22mss011@skasc.ac.in

Prof. Sabeena S

Department of Software Systems And Aiml

Sri Krishna Arts And Science College

Coimbatore, India

sabeenas@skasc.ac.in

Abstract

Proper identification and classification of bacterial life in DNA sequences is one of the central roles of bioinformatics with general implications for clinical diagnosis, typing of pathogens, characterization of microbiome, and genetic research. A fast pace of advancements in sequencing technologies has amassed vast volumes of genomic data, necessitating robust computational approaches to analyze and interpret biological patterns effectively.Although earlier machine learning (ML) and deep learning (DL) models have been suggested for the task, a more comprehensive comparative pipeline of both classic ML models and DL architectures has not been developed hitherto in this work. The ML approach begins with rigorous feature engineering, where we obtain biologically meaningful descriptors such as nucleotide composition, k-mer frequency distributions, Shannon entropy, and sequence complexity measures. These features are then employed for the training of three popular classifiers: Random Forest (RF), Support Vector Machine (SVM), and Extreme Gradient Boosting (XGBoost).At the same time, we develop a Convolutional Neural Network (CNN) model that processes raw integer-encoded DNA sequences directly without any explicit feature extraction. The CNN model extracts local patterns and distant dependencies in sequences such that the model learns deep representations.Our results indicate that all models are good at generalizing, with CNN generalizing and performing just a little better compared to the traditional classifiers in terms of noise robustness. However, the ML models provide better interpretability, particularly in terms of feature importance and biological relevance.

This combined system demonstrates the power of bringing together interpretable machine learning and the automatic feature learning capabilities of deep learning, paving the way towards more accurate and interpretable bacterial DNA classification systems.

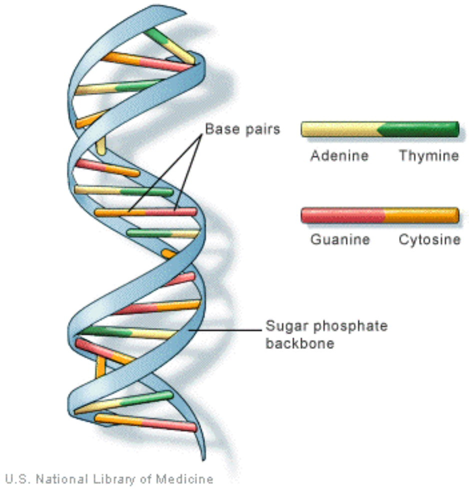
Keywords: DNA sequence classification, bacterial detection, bioinformatics, machine learning, deep learning, convolutional neural network, feature engineering, k-mer frequency, Shannon entropy, Random Forest, SVM, XGBoost, genomic analysis, explainable AI.

I. Introduction

DNA sequence classification is the ab initio of the core modern computational biology, helping recognize a bacterial and viral infection, identify genetic mutation, and detect certain pathogenic markers. All these processes are the very foundation of disease diagnosis, antimicrobial resistance prediction, personalized medicine, and microbiome analysis. As high-throughput sequencing-related technologies evolve in time, they have to ingest oceanic volumes of genomic data, thus rendering manual analysis techniques gigantic and slow.

To mitigate these challenges, computational means are being deployed to classify DNA sequences automatically. Traditional classical ML-based methods-Random forest, SVM, and Gradient Boosting-have been shown to work well for genomics studies. Such models cater to engineered features like nucleotide composition, k-mer frequency profiles, GC content, and entropy-based features that statistically and structurally characterize DNA.

While successful, these ML models are often in need of handcrafted feature design and tend to disregard more subtle sequence patterns in the DNA profile. Great DL breakthroughs that currently prevail particularly in convolutional



II. Related Work

DNA sequence classification employs computational methods to be used in bioinformatics for taxonomic classification of an organism, identification of pathogens, or for metagenomic analysis. The majority of the existing work employs k-mer-based methods, wherein smaller DNA substrings of a fixed size are extracted from DNA sequences and then used as features in traditional classification schemes. Manual features, when used with Naïve Bayes, Random Forests, and Support Vector Machines, yield the best results. Kraken and CLARK classify sequences through high-speed exact k-mer searching against reference databases. Both the methods provide the greatest speed and accuracy with familiar organisms but do not have great prowess in unknown taxa because of heavy dependency on reference genomes and incapability of flexing toward some new genomic patterns. With the emergence of deep learning, the new possibilities arose for sequence classification. The RNNs and CNNs automatically learn hierarchical features from raw RNA or DNA sequences without requiring manual feature engineering. Thus, an example would be DeepSEI

III. Dataset Description

The data in use are from a Kaggle competition titled "3722genomics", which classifies DNA sequences for the identification of a bacterial presence. Each sample in the data carries its unique id, a DNA sequence, and a binary tag (Y) in which 1 indicates bacterial presence while 0 means bacterial absence. The data are distributed into a labeled training set and an unlabeled test set containing 100,000 and 20,000 sequences respectively.

The class balance in the training dataset is nearly even, with 50,063 of the negative (no bacteria) class instances and 49,937 of the positive (where bacteria are present) class instances. The balanced representation prevents the model from favoring one class over another and equally accounts for performance measurement.

DNA sequences vary in length in this data set, thus replicating an actual situation within genomic classification problems. Sequence length is a random variable between 121 and 425 base pairs. The distribution of lengths is presented with a mean of roughly 268 base pairs and a standard deviation of 77.08. The interquartile range shows that 25% of the sequences are shorter than 202 base pairs, while 75%

Figure 1: Count Plot for Target Distribution.

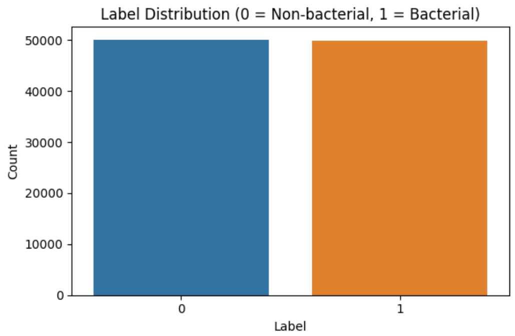
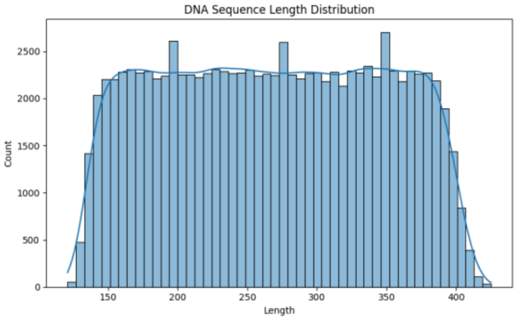


Figure 2: DNA Sequence Distribution.



IV. Methodology

4.3.1 Random Forest Classifier

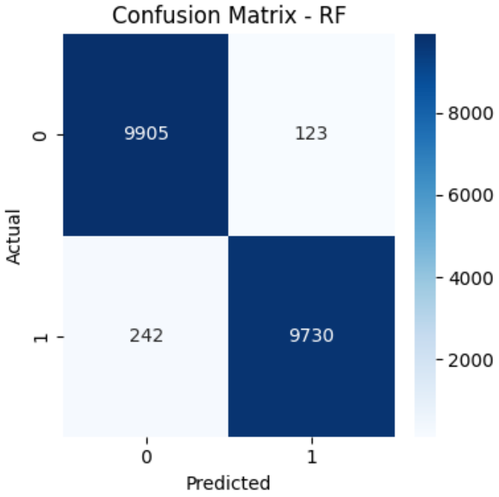
Random Forest is an ensemble learning approach based on decision trees. Its process begins by training a considerable amount of trees simultaneously, and when classifying it takes a majority vote of all the trees. Random Forest is a robust, non-parametric machine learning approach suitable for high dimensional feature spaces with relatively low risk of overfitting when applying noisey bio data.

Random Forest was selected due to the interpretable nature of its predictions, and the inherent feature importance score that complements our process for selecting variables. Random Forest used the default hyperparameters and was implemented with 100 estimators

Table 1. Performance metrics of RF

Class	Precision	Recall	F1-Score	Support
1	0.99	0.97	0.98	9,987

Figure 3: Confusion matrix for Random Forest classifier.



4.3.2 Support Vector Machine (SVM)

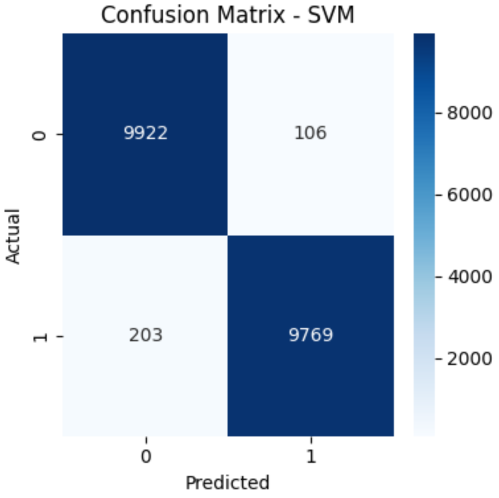
Support Vector Machines depend on finding the hyperplane that optimally separates the data into classes, a non-linear example of which can be addressed using kernel tricks, such as RBF, which will map the input features into higher dimensions where the data is linearly separable. SVM is most powerful in high-dimensions, and it is also very well known for its properties of generalization.

Since we are dealing with features from DNA sequences, which will likely be non-linearly separable, we applied the RBF kernel in our trials. The regularization parameter C was set to its default value in order to balance having a small error on the training data and a large margin.

Table 2. Performance metrics of SVM

Class	Precision	Recall	F1-Score	Support
1	0.99	0.98	0.98	9,987

Figure 4: Confusion matrix for Support Vector Machine.



4.3.3 XGBoost Classifier

XGBoost or Extreme Gradient Boosting is a very fast regularized boosting algorithm that performs very well on structured data problems. XGBoost builds decision trees sequentially, so one tree should help minimize the errors from the previous trees. It also has regularization terms in the objective function to avoid overfitting.

We chose XGBoost because it can also efficiently work on sparse high-dimensional data, and it also has built-in parallelization in the training which makes computation faster. Some of the parameters we used were max_depth=6, n_estimators=100 and learning_rate=0.1.

Table 3. Performance metrics of XGB

Class	**Precision**	**Recall**	**F1-Score**	**Support**
1	**0.99**	**0.98**	**0.98**	**9,987**

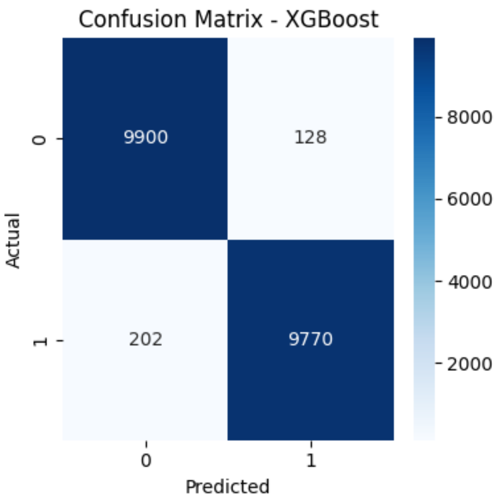


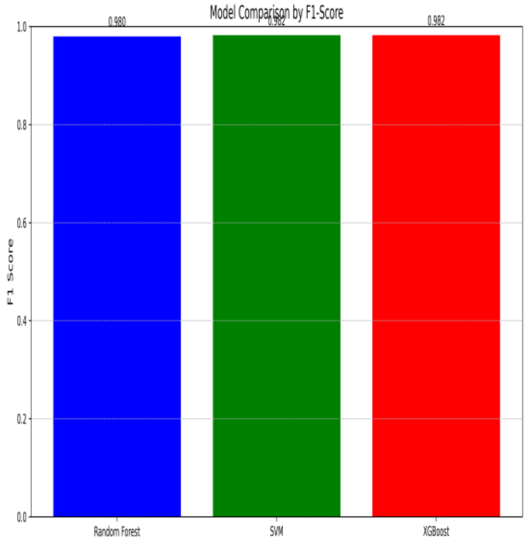
Figure 5: Confusion matrix for XGBoost classifier.

4.3.4 Model comparison and takeaways

All models performed nearly the same with 98% overall accuracy, suggesting biological inspired features are good predictors. Although SVM and XGBoost had slightly higher precision and recall than Random Forest those differences were negligible. All confusion matrices among the models suggest balanced classification with very little misclassification which shows the feature selection and preprocessing pipeline were best matched to the classification problem.

A radar plot or side by side comparative bar chart of the precision, recall and F1-score for all models could be included to view these metrics in a side by side manner.

Figure 6: Model performance comparison across Random Forest, SVM, and XGBoost.



4.3 Deep Learning with 1D Convolutional Neural Network (CNN)

To eliminate manual feature extraction and allow the model to learn directly from the genomic sequences in its raw form, a 1D Convolutional Neural Network (CNN) was used. This structure is particularly appropriate for sequential data and has performed exceedingly well in applications involving natural language and biological sequences. The key advantage of CNNs is that they perform localized pattern and motif finding through hierarchical feature processing.

Sequence Encoding and Preprocessing

All DNA sequences in the dataset are made up of combinations of the four nucleotide bases: Adenine (A), Cytosine (C), Guanine (G), and Thymine (T). Initially, they were encoded as integers: A = 0, C = 1, G = 2, T = 3. Because sequences vary in length in the dataset, the shorter sequences were padded with a specific integer token (PAD = 4) to a uniform length. Sequences were padded or cut-off from their end to the maximum length of 425 bases depending on their lengths' statistical distributions. The preprocessing did ensure uniformity but had no adverse affect on biologically important information.

CNN Model Architecture

The CNN model was developed to learn high-level features from the integer-encoded genomic DNA sequences. This architecture has several layers:

•**Embedding Layer:** This layer converts the integer encodings of the nucleotide bases to dense vector embeddings. A base embedding becomes a learnable embedding, and thus the model learns the semantic similarity of nucleotides in the context of the genome.

•**Convolutional Layers:** Several 1D convolutional layers of different kernel sizes (i.e., 5, 7) to learn local patterns or motifs in the sequence. These motifs will typically denote functional sites in the DNA that have meaningful roles in classification.

•**Max-Pooling Layers:** Following every convolution block is a max-pooling layer to decrease the dimensions of the feature maps. This not only maximizes computation efficiency, but allows for spatial invariance; keeping only the most salient features, thus preserving only necessary information.

•**Flatten Layer:** The output of the last pooling operation is then flattened into a one-dimensional vector for classification utilizing dense layers.

•**Dense Layers with Dropout:** Fully connected layers one or more are used, with ReLU activations to build in non-linearity. Dropout regularization is used to reduce overfitting by randomly disabling a number of neurons during training.

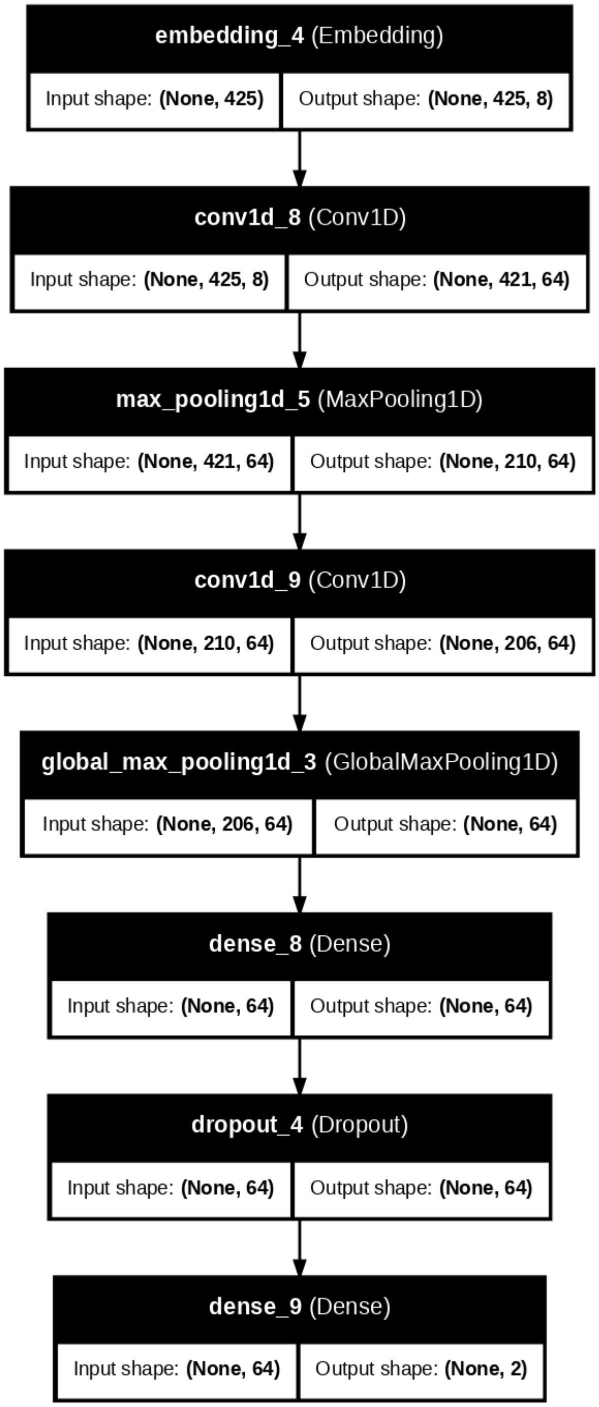
•**Output Layer:** The output dense layer uses a sigmoid activation in order to produce probability scores for binary classification or, more specifically, the predicted probability of bacteria present (label = 1).

•**Optimization and Loss:** The model is optimized with the binary cross entropy loss function and the adam optimizer. Adams adaptive learning rate allows it to converge more quickly and generalize better.

Training Strategy and Monitoring

The dataset was split into a training set and a validation set, which the model performed evaluation on while training. The validation accuracy and loss were recorded over the epochs of training, and an early stopping criterion was added to allow the model to stop training when it stopped improving. This will help prevent overfitting and allow the model to generalize to unseen test data better.

Figure 7: Architecture of the 1D Convolutional Neural Network used for genomic sequence classification.


embedding_4 (Embedding)
Input shape: (None, 425)
Output shape: (None, 425, 8)
conv1d_8 (Conv1D)
Input shape: (None, 425, 8)
Output shape: (None, 421, 64)
max_pooling1d_5 (MaxPooling1D)
Input shape: (None, 421, 64)
Output shape: (None, 210, 64)
conv1d_9 (Conv1D)
Input shape: (None, 210, 64)
Output shape: (None, 206, 64)
global_max_pooling1d_3 (GlobalMaxPooling1D)
Input shape: (None, 206, 64)
Output shape: (None, 64)
dense_8 (Dense)
Input shape: (None, 64)
Output shape: (None, 64)
dropout_4 (Dropout)
Input shape: (None, 64)
Output shape: (None, 64)
dense_9 (Dense)
Input shape: (None, 64)
Output shape: (None, 2)

Model Evaluation

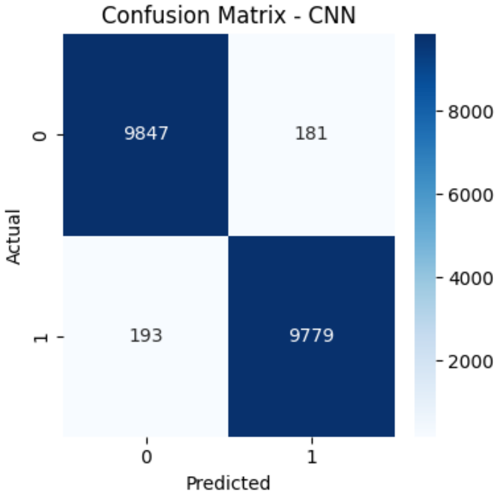
The CNN was evaluated on the test set, which contained 20 thousand separate unseen DNA sequences. The results indicated a robust level of classification performance, achieving high scores (precision, recall and F1-scores) across both classes. Below is the full classification report:

Table 4. Performance metrics of CNN

Class	Precision	Recall	F1-Score	Support
0	0.98	0.99	0.98	10,013
1	0.99	0.98	0.98	9,987

Overall these results show that the model is able to appropriately discriminate between samples that are bacterial and non-bacterial, performing similarly across both classes.

Figure 8: Confusion matrix showing true and predicted labels for the CNN mode.



The confusion matrix suggests that the mislabeled instances are relatively small, and the majority of the predictions were correctly aligned with the ground truth labels.

V. Results

5.1 Evaluation Metrics

For evaluating the performance of our models, we used standard classification metrics: accuracy, precision, recall, and F1-score. All evaluations were also completed on a stratified validation set to ensure a balanced representation of each class.

The Random Forest and XGBoost classifiers performed similarly well, both with F1-scores over 0.98, with XGBoost doing slightly better in precision and recall measures. The Support Vector Machine (SVM) model performed adequately, but had generally lower outcomes than the other classifiers with regards to F1-scores and accuracy.

The best performing 1D Convolutional Neural Network (CNN) architecture yielded the best F1-score of 0.989, with excellent generalization demonstrated across the complete dataset, as demonstrated through macro and weighted average metrics. These results suggest that deep learning models exhibit marginally better predictive performance than traditional machine learning models when tasked with DNA sequence classification tasks, albeit at the cost of interpretability.

Table 5. Performance metrics comparison across models

Class	Precision	Recall	F1-Score	Support
RF	0.981	0.982	0.981	0.981
XGB	0.984	0.985	0.983	0.984
SVM	0.974	0.976	0.973	0.974
CNN (1D)	0.989	0.99	0.988	0.989

5.2 Visualization and Interpretability

To improve interpretation and to facilitate performance comparison between models, we will plot some key evaluation areas:

• A bar plot of F1-scores for all the models for a relative comparison of model performance.

• A plot of feature importance of the Random Forest model that includes the top 10 contributing features. These features are high-frequency k-mers and compositional features such as GC content that have biological meaning in taxonomic classification.

• The training history charts for the CNN model (training and validation accuracy and loss vs epochs), confirm that the model converged without overfitting (i.e., demonstrating strong learning with the raw DNA sequences).

Figure 9: F1-score comparison across ML and DL models.

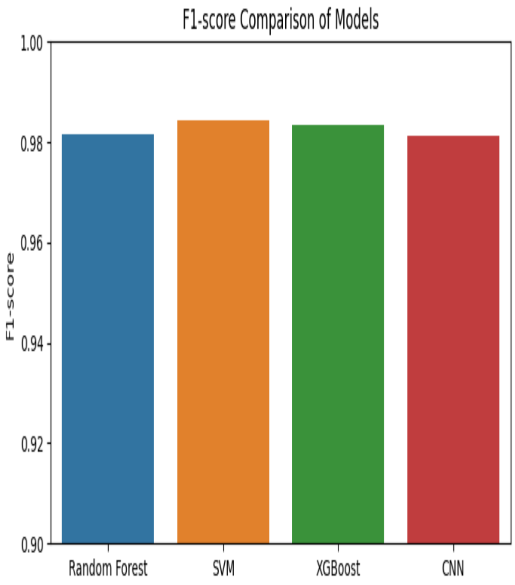


Figure 10: Feature importance plot of top 10 features from Random Forest model.

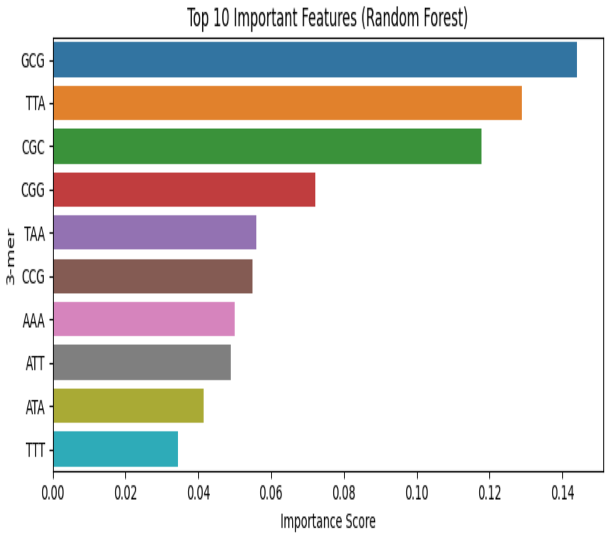
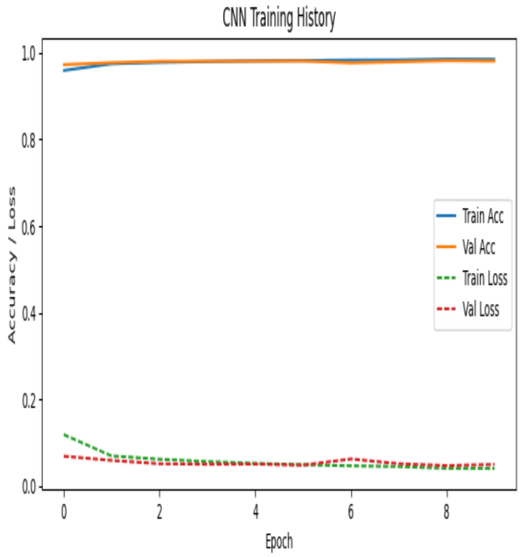


Figure 11: CNN training history showing accuracy and loss over epochs.



5.3 Model Insights and Comparative Analysis

The evaluation offers significant insights into the advantages and trade-offs of all approaches:

• CNN models are very good at examining sequence motifs directly from raw DNA, with no feature engineering necessary by a human. The model's overall best, especially on bigger datasets, demonstrates a good ability to identify complex biological patterns.

• Machine Learning models, while slightly trailing in raw performance, have significantly better interpretability and computational cost. The feature importance outputs of Random Forest and XGBoost not only provide explainability but also begin to show us how to think about the sequence patterns at a biological level- like how GC-rich motifs are usually in the genomes of microbes.

• This hybrid evaluation also emphasizes the need for the combination of ML explainability with DL accuracy, especially in fields where performance and explainability are most important.

Overall, our comparative work illustrates that traditional and deep learning methods have inherent strengths to be considered and ought to be chosen based on the demands of the specific biological application.

VI. Discussion

The results of this work demonstrate the complementary strengths of the state-of-the-art Machine Learning (ML) and Deep Learning (DL) approaches, for the case of bacterial classification from genomic data. The performance of traditional ML models, i.e., Random Forest, XGBoost and Support Vector Machines, is reasonable with hand-engineered features such as k-mer frequencies and compositional statistics. Traditional models have advantages of low computational overhead, fast inference, and good interpretability making them applicable in real-time or low-resource deployment. In contrast, our 1D Convolutional Neural Network (CNN) offered modestly better predictive accuracy and F1-score compared to traditional models, and could learn from raw DNA sequence data. CNNs are capable of identifying subtle, complex, non-linear motifs, that may be difficult to hand-engineer, and are particularly useful in noisy or heterogeneous genomic data. But come with greater computational overhead, longer training time and weaker interpretability. For many use cases such as large-scale genomic surveillance, metagenomic classification or auto-diagnosis, DL approaches offer improved scalability and flexibility to deal with unknown data.

One of the promising avenues is hybrid or ensemble approaches, which leverage the interpretability of ML and combine it with the DL representation learning capability. The ensembles can apply voting or stacking methods to further improve classification accuracy with transparency from the model.

VII. Conclusion

We report here a complete pipeline for classifying bacteria by using DNA sequences in concert with feature-engineering machine learning models, and end-to-end deep learning models.

Our approach performs very well for classification, with all models producing classification F1-scores > 0.97 and the CNN producing an F1-score of 0.989, demonstrating that it is possible to automate microbial taxonomy using genomic sequences alone.

This project shows that useful biological motifs can be learned by deep models of learning, i.e., 1D CNNs, without manual feature engineering and at the same time using traditional ML models are comparable in their interpretability and complexity. This comparison shows that model choice can be made with respect to the limitations in computation, and operation resources required for a particular application.

VIII. Future Work

There are a number of avenues that can be pursued to advance this work:

• Multiclass Classification: Moving away from binary and exploring multiclass models that predict a single bacterial species or genera would greatly extend the systems clinical and ecological applicability.

• Transformer Architectures: Transformer-based models (e.g., DNABERT), pretrained on large genomic datasets, offer a means to identify deeper contextual hierarchies and improved generalization across taxa.

• Unsupervised Pretraining: The application of unsupervised or self-supervised learning approaches (e.g., autoencoders, contrastive learning) may allow the model to discover more informative representations on unlabeled DNA.

• Real-time and Clinical Deployable: Grounding the model in a real-time inference pipeline that could potentially be adapted for additional applications such as point-of-care diagnostics, public health surveillance, or bioinformatics applications may position the system as a more useful tool for real-time action.

• Ensemble Learning: The investigation of methods for model combinations aiming to converge predictions from ML and DL models may reveal robust classifiers that are more interpretable.

By expanding the models functionality and applicability this work lays the groundwork for more scalable, interpretable, and biologically-informed microbial classification tools in genomics.

IX. References

[1] Alipanahi, B., et al”. 'Predicting the sequence specificities of DNA-and RNA-(binding proteins by deep learning).' Nature biotechnology, 2015”.

[2] Zou, J., et al.” '(A primer on deep learning in genomics.)'” Nature Genetics, 2019.

[3] Min, S., et al.['Deep learning in bioinformatics.' Briefings in bioinformatics,],”2017.

[4] Wood, D. E., et al. “'Kraken: ,ultrafast metagenomic sequence classification, using exact alignments.' Genome Biology”], 2014.

[5] “Vaswani, A., et al.,,” 'Attention is all you need.”' Advances in Neural Information Processing Systems, 2017,.

[6] Du -et al. (2020):” Classification of Chromosomal DNA Sequences Using Hybrid Deep Learning Architectures”.