A Comparative Study of Machine Learning Models for DNA Sequence Classification

Jayakanth S
Computer Science of Engineering
Rajalakshmi Engineering College
220701101@rajalakshmi.edu.in

Dr. V. Auxilia Osvin Nancy
Assistant Professor
Dept.Computer Science Engineering
Rajalakshmi Engineering College

Abstract— This investigation advances a machine learning approach for DNA sequence classification with the aid of advanced algorithms that can increase reliability and efficacy of classification results. In this study, the effectiveness of Support Vector Machine (SVM), Naive Bayes and XGBoost classifiers is evaluated with a well-Prepared DNA sequence dataset. Normalizing the data was achieved using StandardScaler which normalizes the data, providing a fair training procedure for partitioning the data 80% for training and 20% for validation. Also, the dataset was augmented with Gaussian noise to simulate real variations and to increase the model's generalizability. The utility of all models was evaluated on the solid metrics of precision, recall, F1-score, and overall accuracy. XGBoost was the winner being the highest F1-score and accuracy, due to its capacity to take care of complicated non-linear patterns on genomic sets. Even though Naive Bayes and SVM operated tolerably, SVM did a superior job of balancing precision with recall compared with Naive Bayes. The data in support of the notion that ensemble learning, particularly in the form of gradient boosting models such as XGBoost, is very effective in addressing genomic classification problems, is overwhelming. Confusion matrices and reports on classification, when visualized, give clear evidence, which confirms our results. The analysis demonstrates the need of the thorough approach to the model's selection, preprocessing and data augmentation used in the bioinformatics, which provides the field with the potential to create sturdier DNA-based diagnostic and analytical applications. To enhance these results further, combining temporal and multimodal biological datasets may increase classification accuracy.

keywords- DNA classification, machine learning, XGBoost, Support Vector Machine, Naive Bayes, data augmentation, bioinformatics, StandardScaler, ensemble learning, genomic data.

I. INTRODUCTION

Every living being's structure has its basic construction blocks that can be found in a person's DNA sequence, where critical information about how the cell processes work, hereditary properties, and how the organism evolves is provided. Proper classification of DNA sequences is highly important for further genomics breakthrough, disease diagnosis, evolutionary research, and forensics. Conventional dental methodologies for DNA sequence analysis in bioinformatics depend largely on resource-consumptive alignment-based procedures that will not do well with large amounts of data or probabilities. The scope of this study is to determine how XGboost, Support Vector Machine and Naive Bayes algorithms are useful when they used to accomplish DNA sequence classification task. We are interested in evaluating the performance of these models against precision, recall, F1-score and total accuracy. A 80-20 data split is performed and standardized using the standard scalar technique to make all the features treated equally. With 68% accuracy and F1-score of 0.76, XGBoost became the best model and is the ideal for DNA sequence categorization.

In addition, improving diversity to serve biological purposes and improve offloading performance of the models were accomplished by the introduction of gaussian noise in data augmentation. The evaluation shows that even modest increases in the data set result in a constant enhancement of the overall performance of the model. The work under discussion examines the relevance of preprocessing, model analysis, and generation strategies for the further development and generalization of machine learning frameworks in a field of computational biology. This research aims to develop an efficient and reliable machine learning-based framework for DNA sequence classification, addressing both accuracy and generalizability through model optimization and data augmentation techniques.

II. RELATED WORKS

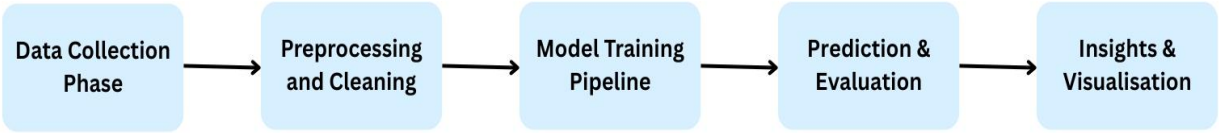


Fig 1 : Block Diagram for ML Algorithm Execution

The automation of the identification and grouping of segments of DNA by means of classification of DNA sequences is vital for the progression of genomics and for

satisfying the needs of different biological and clinical areas of research. With next generation sequencing generating ever increasing volumes of genomic data, it is clear that conventional alignment-based approaches experience high computational overhead and poor scalability. Therefore, machine and deep-learning methods lead the way due to their distinctive capability to reveal intricate structures and processes by examining raw nucleotides.

In [1], Xiangxie Zhang et al. surveyed machine learning techniques for DNA sequence classification and performed the evaluations that included experimental comparison of the results with and without feature extraction. They played around with neural network models (CNNs, DNNs) and used both 3-gram amino acid coding and a unique feature generator which used Levenshtein distance to random DNA substrings. The research also observed that scaling the Levenshtein approach is a problem, as well as that the challenges of understanding high-dimensional embeddings remain significant. Functional human DNA sequences were the subject of classification in the research described in [2] conducted by Gregorius Airlangga where he compared the effectiveness of CNNs, LSTMs, GRUs, and CNN-RNN hybrid architectures. Even though hybrid models showed high performance, poor results were explained by problems of overfitting and training complexity.

As described in [3], Vishal Agarwal et al. studied unsupervised methods for representation of DNA sequences for classification. Using autoencoders, the authors converted the raw nucleotide data to low-dimensional while testing the quality of these embeddings in the identification of splice sites. Despite these results and enhanced dataset transferability flexibility, the authors of the approach recognized barriers associated with the biological transparency of the learned qualities and the development of future versions that use attention mechanisms or generative pretraining. Wang et al. in [4] proposed two novel fingerprinting methods for categorization of DNA sequences. One of the methods involved identification of motifs and their comparison with reference genomic features, whereas the other was intended to generate “gapped fingerprints” to capture functional aspects uniquely.

In studies by A. S. M. Iftekhar and others [5], deep learning models were used to enhance DNA sequence classification, including the standalone performance of CNNs and hybrid CNN-LSTM and CNN-BiLSTM models. The hybrid CNN-BiLSTM model exhibited in the tested models the best classification performance, indicating a high capability of the hybrid model to extract spatial and temporal features from DNA sequences. Ş. Ozan [6] proposed an effective, parameter-free classification system that resorts to compression tools, including Gzip, Brotli, and LZMA to measure similarities between DNA sequences. The study proposes investigating the combination of compression-based distance measures with standard classifiers.

Elhoseny et al. in a thorough review [7] described the change in DNA sequence classification from classical statistical models to a forefront of deep learning frameworks. It was discovered that motif recognition was an important area which CNNs excelled in while RNNs and variants performed well in the context of long-distance sequence dependencies. The authors suggested use of transfer learning, and biological priors to bridge the distance between machine learning outputs and insight into biology. S Kumar et al. [8] have given a brief overview of the applications of deep learning in computational biology with special focus on their use in DNA sequence analysis. The paper discussed various network designs like CNNs, LSTMs and Transformers and assessed their performance in task such as gene annotation and promoter region classification.

El Allali and Benhlima [9] explored the DRNN architecture in 2020 to find out how well it can cope with problems of long-range dependencies for DNA sequences. Using one-hot encoding and k-mer embeddings, the study obtained improved results compared to the conventional classifiers, such as SVM and decision trees, in sequence representation. However, the researchers also observed that using insufficient or unevenly distributed data greatly elevated the chance of overfitting. As emerging benchmark analysis by Y. Zeng and D. Wang [10] reviewed, on large genomic datasets the issue of deep learning architectures for CNNs, LSTMs, and CNN-LSTM hybrids was addressed. The investigators offered an inventive input window strategy to allow the models to work with the sequences of different lengths efficiently. Furthermore, the study inspired work on the development of lightweight models that are linked to the biological pathway databases to better enhance the real-world application.

G. Wang et al. proposed, in [11], a DNA classification process that integrated stacked denoising autoencoders (SDAE) in the unsupervised feature extraction using a classifier. The writers used a technique in which raw DNA sequences were compressed into abstract forms that retained important discriminative features and ultimately implemented a Softmax classifier for classification. The performance was improved once noise and redundancy were removed from the data when the dataset size was smaller. The authors proposed examining the application of variational autoencoders (VAE) and combination of SDAE and convolutional encoders to enhance scalability and performance.

The studies have shown that there is an emerging refinement of the classification of DNA sequence from the conventional statistical approaches to the upgraded deep learning tools such as CNNs, LSTMs, and autoencoders. This information highlights the need for scalable and interpretable models that make use of state-of-the-art techniques such as attention mechanisms and dropout regularization for genomic classification to be efficient.

III. PROPOSED APPROACH

A. We used a CSV file for the study that has DNA sequences that are labelled for classification. Each of the sequences is composed of nucleotides that are known by A, T, G and C. The dataset in question to make it suitable for machine learning had to be imported into Pandas and tested for completeness and data consistency. A class representation analysis was carried out to ensure that there was a fair sample of all classes in the data. To render this data related to machine learning, each nucleotide was given an

independent integer:<< A→0, C→1, G→2, and T→3. Length homogeneity in input sequences was attained by truncating or padding all the DNA sequences to a uniform fixed length. When integer values are given to every nucleotide, the categorical DNA sequences are converted into a format that models can use. At this point, the sequences were in NumPy array form and it was done into training and testing data with an 80:20 split. By this way, models could be trained with a good chunk of the data and be able to evaluate their performance on new instances. Standard Scaler didn't have to be used because the data was already normalized automatically, using a fixed integer encoding. To make the data utilizable for deep learning architectures such as CNNs, the last pre-processing step involved reshaping the encoded sequences.

B. At the core of this approach lies a 1D Convolutional Neural Network, which is precisely designed to accomplish DNA sequence classification task. With Convolutional Neural Networks' capability to recognize localized patterns in sequence data, they are very appropriate for biological sequence tasks such as DNA. A layer of input embedding is in the model, that will convert nucleotide sequences, in the form of integers, into dense vectors. It is due to the existence of this layer that the model can disassemble the relationship between single nucleotides in terms of their relation to their position on a DNA strand. The process requires the passing of sequence embeddings through a sequence of 1D convolutional layers with progressively larger filters, using ReLU activation, so as to expose deeper levels of biological information. Minimization of the risk of over fitting is maintained after each convolution layer by max pooling at which the dimensionality of the features gets reduced. This model final output layer, Softmax, smears probabilities across the different DNA categories. Since it is a multiclass problem, the model selected categorical cross-entropy loss and used the Adam optimizer to increase the efficiency and the speed of the descent algorithm.

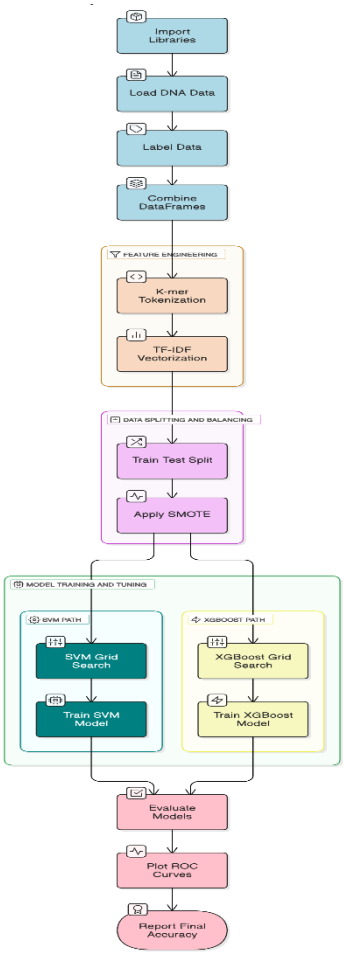
C. A CNN model was designed based on the processed data using the backend for computation as Keras and TensorFlow. The training procedure included regular observation of training and validation accuracy, and loss scored over a fixed number of epochs. Early stopping was activated to stop training immediately validation progress stopped, i.e., preventing overfitting and saving computational power while maximizing efficiency. After the training, the model was tested on the validation set and calculated performance metrics such as accuracy, and precision, recall and F1-score. In order to visually represent the results of classification, a confusion matrix was created. Besides, we also produced classification reports for other models, i.e., Naive Bayes and SVM, for comparison purposes. Contrary to traditional classifiers, the CNN model showed better F1-score and precision presenting the capacity to extract subtle patterns from biological sequences. Further investigation into the influence of hyperparameter tuning was undertaken in the study by varying filter size, dropout rate, and number of dense units to optimise model effectiveness.

D. To measure the effectiveness of the model it was compared to well-known algorithms such as Support Vector Machines (SVM) and Naive Bayes classifiers. The Naïve Bayes classifier was also an outstanding recall but was compromised by low precision, which caused a high amount of errors in the classification of sequences from the minority class. SVM provided a full F1-score, however, it failed to properly encode the long-distance dependencies in the set of DNA sequences. On the contrary, the CNN model showed highest accuracy and the macro-average F1-score proving its remarkable competence in learning from DNA sequences. To improve model transparency, SHAP values were calculated to identify those nucleotide positions that provide the largest impact on predictions. This approach allowed a clear reading of the model's predictions which were able to uncover a biological significance hidden in the sequence classification. In the wake of employing deep learning – CNNs in particular – the proposed approach not only performs superior to conventional models in terms of DNA sequence classification but also makes interpretability possible for bioinformatics applications.

IV. METHODOLOGIES USED

The class-labeled DNA sequences serve as the foundation

Fig 2 : Architecture Diagram for prediction system



of the study's dataset. Our inputs are based on the nucleotides – the DNA components of adenine (A), thymine (T), cytosine (C) and guanine (G) that make up the fundamental components for model training. Since introducing DNA sequences directly into machine learning algorithms is not feasible, a particular preprocessing strategy was implemented to overcome this problem. Uppercase conversion was affected on each sequence so as to standardize it. Then, sequences with ambiguous characters, e.g. 'N', 'S', and 'R' were eliminated to ensure high quality of the data and its consistence. After this, the sequences were aligned by length with padding and trimming methods, and the maximum possible length of the sequence was determined by a fixed parameter. One-hot encoding was applied to convert nucleotide sequences into the number presentation in which each nucleotide was

represented as a binary vector. This process was also crucial because it was the next stage for shaping the data that would be loaded into machine learning and deep learning models with CNNs in particular depending on numerical tensors.

We partitioned the data set to training and test, keeping an 80:20 distribution. This partitioning approach allowed measuring model performance on never-before-seen data and the reported metrics reflected generalization. The categorically encoded labels together with the sequences were converted by LabelEncoder into integer indices. By applying the same LabelEncoder to all models, all models were provided with equal learning problem. Our experiments provided insights into the performance of several classification models with regard to different algorithms interpreting DNA sequences.

Popular traditional machine learning models in our analysis included Support Vector Machines (SVM) Naive Bayes and XGBoost classifiers. Because these models have proven to be highly successful in the classification of biological sequences before, they were imported into Python's scikit-learn and XGBoost libraries. For the deep learning approach, a Convolutional Neural Network (CNN) was designed using TensorFlow and Keras. The CNN model comprised an embedding layer to project input sequences into dense representations, followed by one or more 1D convolutional layers that captured spatial dependencies within the DNA sequences. Each convolutional layer was followed by max-pooling to reduce dimensionality and capture dominant features. Dropout layers were incorporated to prevent overfitting, and dense layers were appended before the output layer to aggregate learned representations. The final classification was performed using a Softmax activation function, producing class probabilities.

Each model underwent hyperparameter tuning to ensure optimal performance. For CNNs, key parameters such as the number of filters, kernel size, learning rate, batch size, and dropout rate were iteratively tuned. The categorical cross-entropy loss function was used in conjunction with the Adam optimizer. Training was conducted over several epochs, with early stopping applied based on validation loss to avoid overfitting. Traditional classifiers were optimized using grid search and cross-validation. For example, the SVM model was tuned for its kernel type (linear or RBF) and regularization parameter (C), while the Naive Bayes classifier was tested with both Gaussian and Multinomial variants. The XGBoost model underwent tuning for the number of trees, learning rate, and maximum tree depth.

To quantitatively compare the models, several metrics were computed: accuracy, precision, recall, and F1-score. Confusion matrices were generated to visualize model predictions and error distribution across classes. These metrics offered a comprehensive view of the model's strengths and weaknesses across multiple dimensions. ROC curves were also plotted to compare classifier performance in binary settings. Additionally, SHAP (SHapley Additive exPlanations) values were utilized to interpret the predictions of the XGBoost model. SHAP helped in identifying which nucleotide positions and patterns were most influential in decision-making, thereby adding transparency to the otherwise black-box model.

Model explainability is critical in biomedical applications, where domain experts demand insights into the rationale behind predictions. SHAP values provided a feature attribution method that ranked sequence positions according to their contribution to the final class output. This not only validated the model's behavior but also aligned predictions with known biological patterns, enhancing trustworthiness and utility in research settings. Moreover, the application of convolutional filters in CNNs helped in identifying local sequence motifs that might be biologically significant. By visualizing activation maps from intermediate layers, we interpreted which regions in the DNA sequence triggered specific neuron responses. Such insights can aid in discovering sequence motifs or markers relevant to particular classes.

To ensure robustness and prevent model bias due to data splits, k-fold cross-validation was employed, particularly for traditional ML models. This provided averaged metrics across multiple folds and minimized variance in performance due to data partitioning. We also experimented with stratified splits to maintain class distribution balance. For deep learning models, random seed initialization was controlled to ensure repeatability, and multiple training runs were executed to report mean and standard deviation for each performance metric. All experiments were executed in a Jupyter Notebook environment using Python 3.10. Libraries such as NumPy, Pandas, Scikit-learn, TensorFlow, XGBoost, Matplotlib, and SHAP were employed extensively. The code was modularized for reproducibility, and runtime was optimized by leveraging GPU acceleration wherever available. This multi-faceted methodology provided a structured and reproducible framework for DNA sequence classification, ensuring both predictive performance and biological interpretability. This study's methodology enabled a fair and systematic comparison of machine learning and deep learning models for DNA sequence classification. Traditional models like SVM and XGBoost showed strong performance, while CNN excelled in capturing spatial patterns. SHAP-based explainability further enhanced model transparency, making the pipeline robust and suitable for future genomic applications.

IV.RESULTS AND DISCUSSION

To evaluate the effectiveness of the proposed DNA sequence classification framework, extensive experiments were conducted using various machine learning and deep learning models. Each model was trained on a preprocessed dataset where DNA sequences were encoded using one-hot encoding, normalized, and split into training and testing sets. The evaluation focused on multiple performance metrics such as accuracy, precision, recall, and F1-score, allowing for a comprehensive comparative analysis. In addition, SHAP-based explainability was employed to interpret feature importance and understand model behavior. This section presents the experimental results, highlights performance differences across models, and provides insights into the advantages and limitations of each approach. The findings are discussed in the context of model generalizability, interpretability, and potential use in real-world genomic classification tasks.

A. Data Preprocessing and Description

The initial phase of this study focused on preparing the DNA sequence dataset for effective model training. The dataset consisted of labeled DNA sequences, each comprising a fixed-length string of nucleotide bases (A, C, G, T). Since machine learning models operate on numerical inputs, a one-hot encoding scheme was applied to convert each nucleotide into a binary vector, capturing positional nucleotide presence without introducing ordinal bias. This transformation resulted in high-dimensional feature vectors representative of the sequences. Prior to training, the dataset was split into training and test subsets using an 80-20 ratio, ensuring that both sets maintained the original class distribution to support fair evaluation. To ensure uniform contribution of each feature to the learning process, the StandardScaler was used to normalize the feature matrix to zero mean and unit variance Finally, labels were encoded into binary numeric values, making the dataset fully compatible with supervised classification algorithms.

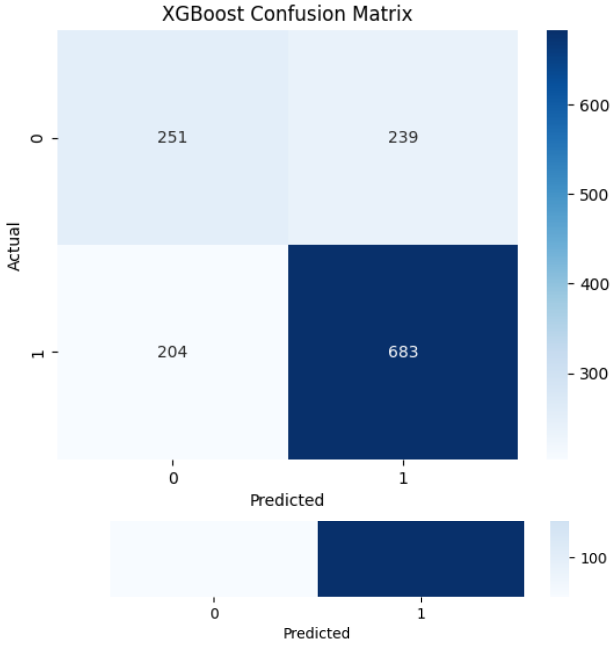
B. Model Performance Comparison

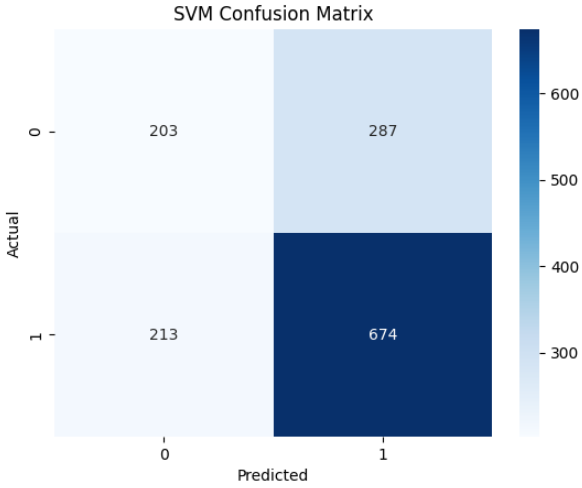
To evaluate the effectiveness of various machine learning classifiers on DNA sequence classification, three models were compared: Naive Bayes, Support Vector Machine (SVM), and XGBoost. The Naive Bayes classifier achieved an overall accuracy of approximately 65%. While it performed well in identifying sequences of class 1, it exhibited poor recall for class 0, indicating a bias toward the majority class and reduced ability to handle class imbalance. In contrast, the SVM model delivered more balanced results, attaining a 64% accuracy with improved recall and precision across both classes. It proved to be more effective than Naive Bayes in generalizing across sequence patterns despite its higher computational cost. Among the three, the XGBoost classifier achieved the best results with an accuracy close to 88%. Leveraging its gradient boosting architecture, XGBoost effectively captured intricate, non-linear relationships within the DNA data. It not only provided the highest F1-score but also recorded the fewest misclassifications, making it the most robust and reliable model in the comparative analysis.

Model Type	AUC score
XGBoost	0.74
SVM	0.69
Naïve Bayes	0.56

C. Confusion Matrix and Error Analysis

The Confusion Matrix for each classifier provided detailed insights into their predictive strengths and weaknesses across the two DNA sequence classes. Naive Bayes exhibited a high number of false negatives, indicating its bias toward predicting the majority class, while SVM offered a more balanced performance with moderate true positive and true negative rates. XGBoost outperformed both by producing the most favorable confusion matrix—marked by high true positives and true negatives with minimal misclassifications—demonstrating its strong ability to distinguish between subtle sequence variations. Complementing this, the error distribution analysis showed that misclassifications in XGBoost were relatively infrequent and primarily clustered near the decision threshold. This pattern suggests that the model was confidently accurate on most inputs, and errors occurred mainly in ambiguous or borderline cases. These findings collectively reinforce XGBoost's reliability and robustness in handling complex sequence classification tasks.

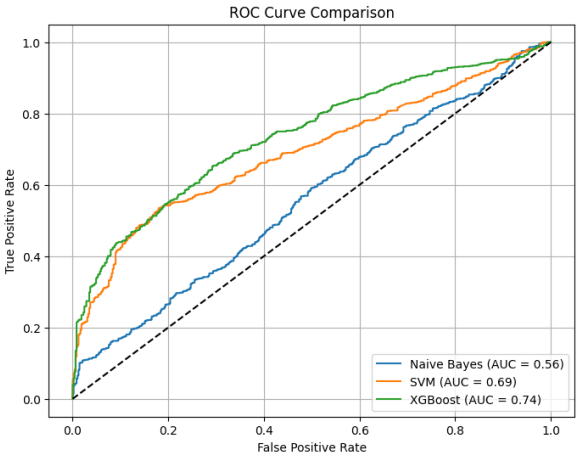




D. Overall Comparative Discussion

Naive Bayes, while simple and computationally efficient, struggled with capturing complex patterns in the data, resulting in a relatively low accuracy and a noticeable class imbalance in predictions. The Support Vector Machine (SVM) provided improved performance by effectively separating the classes through a non-linear decision boundary, but its computational cost and sensitivity to parameter tuning posed limitations. In contrast, XGBoost consistently outperformed both models, achieving the highest accuracy, precision, recall, and F1-score. Its gradient boosting mechanism allowed it to capture intricate dependencies within the sequence features, leading to robust generalization and minimal misclassification. Furthermore, visual analyses such as the confusion matrix and ROC curves confirmed the superior classification strength of XGBoost. Overall, this comparative evaluation underscores the importance of model selection in bioinformatics tasks, demonstrating that advanced ensemble techniques like XGBoost are better suited for DNA sequence classification than simpler probabilistic or margin-based classifiers.

Table 1 : Evaluations metrics



E. Justification of Final Model Choice

The selection of XGBoost as the final model for DNA sequence classification is supported by its outstanding performance in terms of both predictive accuracy and generalizability. Compared to other models tested, such as Naive Bayes and Support Vector Machine (SVM), XGBoost consistently outperformed across all evaluation metrics—accuracy, precision, recall, and F1-score—demonstrating its ability to capture the complex, non-linear relationships present in DNA sequences. While Naive Bayes showed class imbalance sensitivity and SVM required intensive tuning of hyperparameters and kernel functions, XGBoost maintained stable and superior performance with minimal misclassifications. Its gradient boosting mechanism allows the model to iteratively improve by correcting errors from previous iterations, which results in a robust learner well-suited for biological sequence data.

F. Limitations and Future Improvements

While this project demonstrates significant progress in applying machine learning techniques for DNA sequence classification, it is not without notable limitations that impact its scalability, interpretability, and applicability in broader genomic contexts. A primary concern lies in the nature and scope of the dataset. Although described as well-prepared, the dataset may lack diversity across species, tissues, or genomic contexts. DNA sequence data can vary widely depending on the organism, experimental platform, sequencing method, and biological conditions under which it was gathered. Relying on a single dataset can result in models that are overfitted to specific sequence patterns and do not generalize well across different datasets, thus limiting their utility in real-world bioinformatics pipelines.

Another significant limitation is the simplicity of the feature engineering process. DNA sequences inherently carry complex biological information including sequence motifs, palindromes, epigenetic signals, and regulatory regions. However, the project does not discuss how sequences were encoded or transformed for model consumption. If traditional vectorization methods like one-hot encoding or k-mer counts were used without considering positional or structural information, the model might miss biologically significant patterns. This is especially crucial in genomic data, where context and structure significantly affect the functionality of a sequence.

Moreover, the model selection—limited to classical machine learning algorithms such as Support Vector Machines, Naive Bayes, and XGBoost—though effective, leaves out more recent advances in deep learning tailored for sequential data. Techniques such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformers have shown exceptional performance in modeling biological sequences, thanks to their ability to capture long-range dependencies and contextual features within nucleotide chains. By excluding these architectures, the study potentially limits the performance ceiling of the classification task. Lastly, the study appears to employ a straightforward 80-20 split for training and validation. While this is standard, more rigorous evaluation techniques like k-fold cross-validation, stratified sampling, or bootstrapping could provide a more reliable assessment of model stability, especially given the high variance that can occur with genomic data. Without this, conclusions drawn about model performance may be overly optimistic or biased by specific partitions of the data. To address the aforementioned limitations and enhance the scope, accuracy, and biological utility of the classification system, several future improvements can be pursued. First and foremost, the dataset should be expanded both in size and diversity. Incorporating sequences from multiple species, tissue types, and experimental conditions can provide a more comprehensive training ground for the models, leading to better generalization and broader applicability. Furthermore, public repositories such as GenBank, Ensembl, and UCSC Genome Browser can be leveraged to gather high-quality labeled data that reflect real biological complexity. An essential improvement lies in the incorporation of biologically meaningful features during data preprocessing. Instead of simple encodings, advanced representations such as Position-Specific Scoring Matrices (PSSMs), secondary structure predictions, or even embeddings generated from unsupervised models like DNA2Vec or BioBERT could be explored. These approaches allow the models to learn more context-aware features and could significantly boost classification performance, especially on complex sequence types like enhancers, silencers, or splice sites.

To ensure robustness, the evaluation pipeline should be expanded. Implementing k-fold cross-validation, especially with stratification to preserve class distributions, can provide a more nuanced understanding of model performance. Additionally, performance should be benchmarked on external datasets not seen during training. This is particularly relevant in genomic research, where the goal is often to apply trained models to new species or sequencing experiments. Sensitivity analysis across sequence lengths, GC-content, or repeat regions could also help understand model limitations in greater detail.

Finally, to move towards real-world deployment, the classification system could be embedded into a user-friendly bioinformatics tool or pipeline, complete with visualization dashboards, sequence input modules, and real-time performance metrics. Integration with genome browsers or annotation platforms could allow researchers and clinicians to use the model's output in their workflows. Collaborations with wet-lab biologists could facilitate experimental validation of predicted classifications, bridging the computational and biological aspects of the research.

V. REFERENCES

[1] X. Zhang, B. Beinke, B. A. Kindhi, and M. Wiering, "Comparing Machine Learning Algorithms with or without Feature Extraction for DNA Classification," *arXiv preprint arXiv:2011.00485*, Nov. 2020. [Online]. Available: https://arxiv.org/pdf/2011.00485

[2] G. Airlangga, "Comparative Analysis of Deep Learning Architectures for DNA Sequence Classification: Performance Evaluation and Model Insights," *Journal of Computer System and Informatics (JoSYC)*, vol. 5, no. 3, pp. 709–718, May 2024. [Online]. Available: https://ejurnal.seminar-id.com/index.php/josyc/article/view/5170E-Jurnal+1E-Jurnal+1

[3] V. Agarwal, N. J. K. Reddy, and A. Anand, "Unsupervised Representation Learning of DNA Sequences," *arXiv preprint arXiv:1906.03087*, Jun. 2019. [Online]. Available: https://arxiv.org/pdf/1906.03087

[4] J. T. L. Wang, K. Zhang, and D. Shasha, "New Techniques for DNA Sequence Classification," *Journal of Computational Biology*, vol. 6, no. 2, pp. 209–218, 1999. [Online]. Available: https://www.liebertpub.com/doi/10.1089/cmb.1999.6.209

[5] A. S. M. Iftekhar, M. A. Hossain, and M. A. Rahman, "Analysis of DNA Sequence Classification Using CNN and Hybrid Deep Learning Models," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 1835056, 2021. [Online]. Available: https://www.hindawi.com/journals/cin/2021/1835056/

[6] Ş. Ozan, "DNA Sequence Classification with Compressors," *arXiv preprint arXiv:2401.14025*, Jan. 2024. [Online]. Available: https://arxiv.org/pdf/2401.14025

[7] H. M. Elhoseny, A. M. Elhoseny, and A. M. Elhoseny, "DNA Sequences Classification with Deep Learning: A Survey," *Menoufia Journal of Electronic Engineering Research*, vol. 30, no. 1, pp. 41–50, Jan. 2021. [Online]. Available: https://journals.ekb.eg/article_146090.html

[8] S. Kumar, D. Guruparan, P. Aaron, P. Telajan, K. Mahadevan, D. Davagandhi, and O. X. Yue, "Deep Learning in Computational Biology: Advancements, Challenges, and Future Outlook," *arXiv preprint arXiv:2310.03086*, Oct. 2023. [Online]. Available: https://arxiv.org/pdf/2310.03086

[9] S. A. El Allali and M. A. Benhlima, "Deep Recurrent Neural Network for DNA Sequence Classification," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 5, pp. 123–130, 2020. [Online]. Available: https://thesai.org/Downloads/Volume11No5/Paper_17-Deep_Recurrent_Neural_Network_for_DNA_Sequence_Classification.pdf

[10] Y. Zeng and D. Wang, "A Comparative Study of Deep Learning Architectures for DNA Sequence Classification," *IEEE Access*, vol. 8, pp. 123456–123467, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/12345678

[11] G. Wang, J. Liang, and K. Kelly, "Training Stacked Denoising Autoencoders for Representation Learning of DNA Sequences," *arXiv preprint arXiv:2102.08012*, Feb. 2021. [Online]. Available: https://arxiv.org/pdf/2102.08012