algorithms

MDPI

Article

Enhanced Viral Genome Classification Using Large Language Models

Hemalatha Gunasekaran *, Nesaian Reginal Wilfred Blessing, Umar Sathic and Mohammad Shahid Husain

College of Computing and Information Sciences, University of Technology and Applied Sciences, Ibri 516, Oman; wilfred.blessing@utas.edu.om (N.R.W.B.); umar.sathic@utas.edu.om (U.S.); mohammad.shahid@utas.edu.om (M.S.H.)

* Correspondence: hemalatha.david@utas.edu.om

Abstract: The classification of genomic sequences is a crucial area of research in the field of virology. This is due to the increasing number of outbreaks we have faced in recent times. We have a vast repository of genomic sequences from various species, including humans, animals, plants, bacteria, and viruses, which tend to mutate and form new variants or strains. In the realm of machine learning, several models are employed for genome sequence classification. Among these are traditional algorithms such as Random Forest (RF), K-nearest neighbors (KNNs), Decision Tree (DT), and Naive Bayes (NB), each offering unique advantages in handling genetic data. Additionally, deep learning models like Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Bi-Directional LSTM networks are utilized for their robust capabilities in capturing complex patterns and dependencies within genomic sequences. In this study, we explored the application of Natural Language Processing (NLP) techniques to classify the genomic sequences. The focus of our research involves utilizing advanced large language models (LLMs) such as DNABERT, DNAGPT, and GENA LM, which are fine-tuned explicitly on the language of DNA. In this research, after a detailed analysis, we found that DNAGPT achieved an accuracy of 96%, which exceeds the performance of state-of-the-art machine learning and deep learning models.

Keywords: large language models; DNA sequence classification; DNABERT; DNAGPT; GENA-LM; transformer; embedding; attention mechanism

check for updates

Academic Editors: Carla Piazza, Roberto Pagliarini and Wing Kin Sung

Received: 9 March 2025
Revised: 13 May 2025
Accepted: 15 May 2025
Published: 22 May 2025

Citation: Gunasekaran, H.; Wilfred Blessing, N.R.; Sathic, U.; Husain, M.S. Enhanced Viral Genome Classification Using Large Language Models. *Algorithms* **2025**, *18*, 302. https://doi.org/10.3390/a18060302

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Genomic sequence classification is a fundamental task in bioinformatics that helps understand the taxonomy of viruses. The taxonomic classification of a virus becomes especially important during a pandemic or widespread epidemic. Identifying the virus taxonomy is crucial for tailoring targeted treatments, expediting the development of drugs and vaccines, and enhancing public health response strategies, including outbreak control measures [1]. There are around 10 nonillion (10^{31}) individual viruses existing on the Earth [2]. The genomic sequence frequently undergoes mutations that alter its genetic structure, resulting in a new variant, making it difficult to identify its taxonomy.

Several bioinformatics tools utilize the local alignment method to classify genome sequences. BLAST version 1.30+ (Basic Local Alignment Search Tool) is one of the most widely used tools for comparing a DNA sequence with a database of known DNA sequences and identifying regions of similarity [3]. The classification is based on the top matching sequences in the database. Some tools utilize local and global alignment methods, such as the Smith–Waterman algorithm [3], for classifying DNA sequences. All these

alignment methods can achieve high precision only if the sequence is already known and present in the database. As the genome sequence tends to mutate, many of the strains remain unclassified.

To overcome the above limitation, AI uses alignment-free methods such as extracting features from DNA sequences, including k-mer [3], statistical features [4] such as GC content, dinucleotide frequencies, codon usage bias, and the presence of specific motifs. These extracted features are utilized by machine-learning models, such as Random Forest (RF), K-nearest neighbor (KNN), Naive Bayes (NB), Decision Tree (DT), and Support Vector Machine (SVM) [5–7], to create an accurate classification model for classifying genomic sequences.

Deep learning algorithms are more powerful than machine learning algorithms as they can automatically extract complex features from datasets and learn complex patterns autonomously. In contrast, machine learning algorithms are more versatile and often sufficient for more straightforward tasks. Machine learning algorithms are effective with small datasets. However, as both the dataset size and DNA sequence length increase, deep learning models tend to outperform traditional machine learning approaches. Models such as CNN, LSTM, and hybrid models like CNN-LSTM and CNN-Bidirectional LSTM are used in DNA sequence classification [8–10].

Recently, Transformers have gained significant attention after the release of ChatGPT (3.1). Transformers are primarily used for NLP tasks, such as text generation, summarization, and prediction. These Transformer models can also be used to understand genomic sequences, which are similar to natural language [11]. Given the similarities between genomic sequences and natural language, Transformers can be effectively applied to understanding genomic sequences. Their capabilities encompass a range of tasks, including genomic sequence classification, promoter prediction, gene expression prediction, and other related functions.

In this study, we utilized pre-trained large language models (LLMs) that employ the Transformer architecture as their foundational framework [12–16]. These LLMs are initially trained on vast datasets and subsequently fine-tuned specifically on Human DNA sequences as well as DNA from various other species. This fine-tuning enables the models to effectively understand and interpret the language of DNA. LLMs can handle long DNA sequences, which can be challenging for traditional methods. These LLMs can also be applied to various downstream tasks, such as promoter prediction (identifying regions of DNA that initiate transcription of a specific gene), gene expression prediction (forecasting gene activity levels by analyzing and interpreting complex biological data), TF-DNA binding predictions (predicting the interactions between transcription factors and DNA sequences), and DNA sequence classification task (categorizing DNA sequence into predefined class or category such as identifying the gene family)

Our research focuses on discussing and evaluating the performance of several prominent large language models (LLMs), including DNABERT [13], DNABERT-2 [14], GENA-LM [15] (employing the BigBird architecture), and DNAGPT. Each of these models has been adapted to enhance its capabilities in DNA sequence analysis, enabling advanced classification and understanding of genetic information. By comparing these models, we aim to highlight their respective strengths and weaknesses and assess their utility in genetic research and applications.

2. Related Works

Genomic sequences serve as the blueprint for all living organisms, carrying critical information for various biological processes. Analyzing these sequences is crucial for various important tasks, including detecting genetic variations. These applications rely

on understanding the intricate details encoded within genomic sequences to unlock new insights and potential treatments.

In recent years, technological advancements have significantly enhanced our ability to analyze genomic data. Techniques ranging from traditional machine learning to advanced deep learning models have played a crucial role in this progress. Machine learning algorithms have been employed to identify patterns and correlations within genomic data, facilitating tasks such as variant identification and disease susceptibility prediction. Meanwhile, deep learning models have pushed the boundaries further by automatically extracting complex features from large genomic datasets, which is particularly useful for analyzing high-dimensional data.

Numerous studies have explored and adapted traditional machine learning (ML) techniques for classifying DNA sequences. Techniques such as SVM, DT, RF, NB, and KNN have been extensively utilized [5]. Hamed et al. [6] demonstrated that SVM achieved the highest accuracy of 96.3% in DNA sequence classification compared to other machine learning models. Similarly, Qayyum et al. [7] classified the COVID-19 virus using RF, achieving an accuracy of 99.7%. While these models have proven effective in scenarios with limited data, they rely heavily on manual feature engineering and struggle to scale with larger datasets and longer sequences. This limitation has led to the adoption of deep learning models, particularly CNN, which have been shown to be powerful tools for DNA sequence classification.

Nguyen et al. investigated DNA sequence classification utilizing a 1D CNN, which leverages their kernel-based architecture to identify local nucleotide patterns. This approach achieved a 99.06% accuracy on a promoter dataset [8]. Meanwhile, Gomes et al. introduced a novel method for DNA sequence representation using a pseudo-convolutional approach, achieving 99.18% accuracy in classifying the COVID-19 virus [9]. Despite these successes, a significant challenge remains in capturing long-range dependencies due to the localized receptive fields of these models, suggesting the potential for more effective hybrid approaches.

To address the issue of capturing sequential dependencies in DNA sequences, researchers have often employed recurrent neural networks (RNNs), including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks. Gunasekaran et al. applied hybrid models that combine CNN and LSTM architectures to classify a five-class virus dataset, achieving an accuracy of 93.16% [10].

The emergence of large language models (LLMs) has significantly enhanced the classification of DNA sequences by drawing analogies between biological sequences and natural languages. Models from the BERT and GPT families have been specifically adapted for bioinformatics applications, leading to the development of specialized models like DNABERT, Protein BERT, and DNAGPT.

In the DNABERT [12] model, DNA sequences are treated as an analogy to language, with sequences divided into tokens using the k-mer tokenization method. In this framework, tokens are considered akin to words in natural language. DNABERT effectively captures DNA patterns and has been applied to various tasks, such as classification and predicting transcription factor binding sites. etc.,

DNABERT-2 [13,14], the advanced iteration of DNABERT, is tailored for multi-species genomic analysis. It employs masked language modeling techniques, where specific k-mers are masked, and the model is trained to predict these masked k-mers. This approach enables a deeper understanding of genomic sequences. DNABERT-2 is equipped for additional downstream tasks, including motif detection and cross-genomic comparisons etc.

Transformer-based models, such as GENA-LM [15] and DNAGPT [16], have shown impressive performance in DNA sequence classification. These models utilize attention

mechanisms and a hybrid architecture that combines attention with convolutions to capture local and global sequence dependencies [15,16]. For example, HyenaDNA [17] uses implicit convolutions to process extended contexts efficiently [15]. Moreover, research on promoter classification and detection using Nucleotide Transformers [18] has noted significant performance improvements [16].

However, challenges such as the quadratic complexity of attention mechanisms and limited application to genome-scale data continue to affect Transformer architectures. Genomic modeling is further influenced by feature representation strategies, with approaches such as k-mer tokenization, positional encoding, and nucleotide-specific representations commonly used to enhance model performance. Achieving a balance between the granularity of representations and computational efficiency remains a critical challenge. Despite the progress made with pre-trained Transformer-based architecture, their application to genomic data has been limited, revealing a crucial gap in adapting large language models (LLMs) for creative genomic applications [18–20].

This study aims to classify DNA sequences using pre-trained large language models (LLMs), such as DNAGPT, DNABERT, and GENA-LM.

3. Methodology

3.1. Dataset

A genomic sequence contains four nucleotides: A—Adenine, C—Cytosine, T—Thymine, G—Guanine. The dataset used in this research comprises the genomic sequences of three viruses: BAT Coronavirus, MERS, and COVID-19 Coronavirus, downloaded from the National Center for Biotechnology Information (NCBI) https://www.ncbi.nlm.nih.gov (accessed on 3 June 2024), a public repository of genomic sequences. The DNA sequence is downloaded and stored in FASTA files, comprising sequences of varying lengths, ranging from 8 to 38,000 nucleotides.

The number of sample distributions in each class is shown in Figure 1. Since the dataset is unbalanced, the Synthetic Minority Over-sampling Technique (SMOTE) [21] balances the dataset. Initially, the dataset is divided into training and testing sets in an 80:20 ratio, respectively. SMOTE is applied only to the training dataset, the testing dataset is used for model evaluation. Since SMOTE works only with numerical data, DNA sequences are encoded using a label encoder. Label encoding is a machine learning and data analysis technique to convert categorical variables into a numerical format. For each minority class instance, SMOTE identifies its k-nearest neighbors within the same class, randomly selects one of these neighbors, and creates a new synthetic sample by interpolating between the original sample and the selected neighbor. This process is repeated until the minority class is sufficiently augmented to balance with the majority class. The distribution after applying the SMOTE is shown in Figure 2.

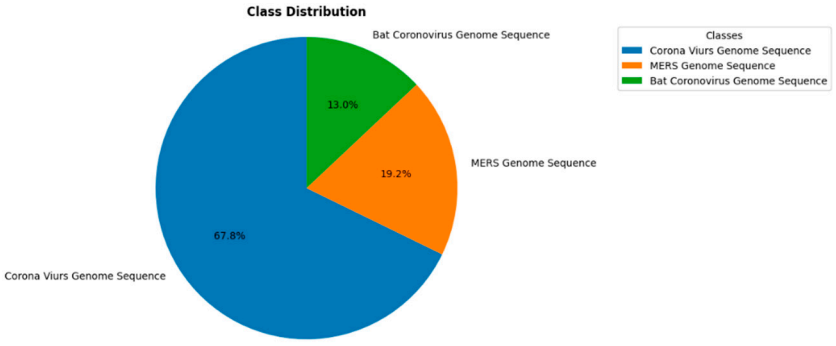


Figure 1. Dataset distribution before SMOTE.

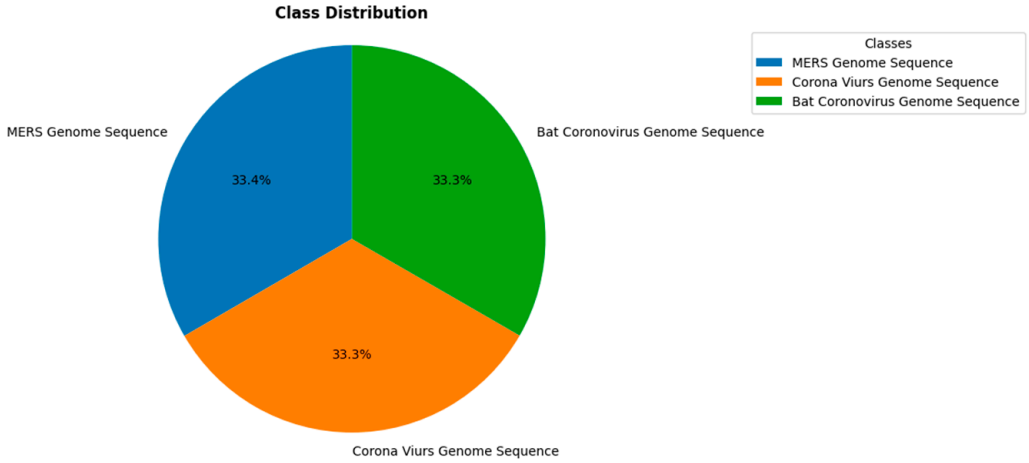


Figure 2. Dataset distribution after SMOTE.

3.2. Transformer

Transformers are mainly used for solving sequence-to-sequence problems. The Transformer architecture begins with the fundamental step of tokenization, where each word in the input is broken down into tokens. These tokens are then converted into numerical representations through a process called embedding. Embeddings transform discrete tokens into continuous vectors, enabling the model to process and learn from them effectively. Once the tokens are transformed into numbers, it is crucial to consider the word's position within a sentence to capture the sentence's meaning accurately. This is achieved through positional encoding, which enables the model to maintain the token order and comprehend the sequence context. As a result, we obtain a vector representation for every token that is fed into the model.

Each Transformer block comprises two essential components:

- The Attention Component;
- The Feedforward Component.

Attention is a powerful mechanism that enables language models to grasp context. Attention is the process of giving more weight to the important words in a sentence [18,22]. Every word in a sequence will have a context vector, which is a linear combination of the weighted sum of all the hidden vectors; hence, the ith context vector is as follows:

$$c_i = \sum_{j=1}^{n} a_{ij} h_j \quad (1)$$

where a is the attention weight vector. It is a set of weights that determines the focus each word (or token) in the sequence should receive when computing the context vector, and h typically denotes the hidden vectors associated with each word (or token) in the sequence.

3.3. DNABERT

Like BERT, DNABERT employs a Transformer encoder architecture. It consists of 12 layers of Transformer blocks. DNABERT was the initial Transformer-based foundational model for DNA sequence classification, capable of handling sequences of length 512 base pairs [12]. The model was initially pre-trained only on the human reference genome. DNABERT uses k-mer tokenization, where a k-mer is a substring of a long sequence. For example, consider a sequence "ATGTAC"; 3-mers of this sequence will be ATG, TGT, GTA, and TAC. The most commonly used k-mer sizes in DNABERT are three, four, five, and six. These k-mer sizes help the model capture varying levels of sequence context, which is crucial for different genomic tasks. However, k-mer tokenization has limitations, such as information leakage and reduced computational efficiency during training. These

limitations were addressed in DNABERT-2 by replacing k-mer tokenization with Byte Pair Encoding (BPE) [13]. Additionally, DNABERT-2 was pre-trained on multi-species genomic sequences, including those of humans, mice, yeast, and viruses. This version can handle sequences of lengths greater than those of the baseline model. DNABERT-2 replaced the positional embedding of DNABERT with Attention with Linear Biases (ALiBi) to eliminate input limitation. Additionally, DNABERT-2 utilizes flash attention and low-precision layer normalization to enhance computation and memory efficiency. It also employs Low-Rank adaptation (LoRA) for fine-tuning the hyperparameter. DNABERT is more suited for applications demanding fewer computational resources, while DNABERT-2 is more suited for tasks that require more extensive model capacity and cross-species sequence understanding. Primarily, it is used to capture deeper dependencies within the DNA sequence.

3.4. GENA-LM

GENA-LM utilizes the BigBird (Big Transformer with Spare Attention) architecture, specifically designed to handle long sequences effectively [15]. It can handle sequence lengths ranging from 4.5 kbp to 36 kbp. A standard Transformer uses a quadratic attention mechanism, whereas BigBird uses a sparse attention mechanism, which can process long sequences and understand extensive context. This model can be utilized for multiple downstream applications, including promoter activity prediction, splicing analysis, identification of polyadenylation sites, enhancer annotation, and profiling of chromatin states. It utilizes the BPE (Byte Pair Encoding) algorithm, also known as the sub-word tokenization model. BPE consists of two parts: a token learner and a token segmenter. A token learner takes a raw training corpus and induces a vocabulary. A token segmenter takes a raw test sentence and tokenizes it according to the vocabulary. Frequently occurring tokens are merged into the vocabulary. These steps are repeated until the vocabulary reaches a specific size. GENA-LM features a conventional attention mechanism, enabling the model to extract essential features from all tokens. GENA-LM is trained on the latest T2T human genome assembly and augmented with 1000-genome SNP (Single Nucleotide Polymorphism) and multispecies data. It includes different variants, such as GENA–LM-base, GENA-LM-large, and GENA-LM-BigBird, for handling varying sequence lengths of up to 36,000 base pairs.

3.5. DNAGPT

DNAGPT is based on the classical GPT (Generative Pre-trained Transformer) architecture [16]. Like GPT, DNA-GPT relies on a Transformer decoder architecture. It processes the sequence unidirectionally. The DNA sequences are first converted into non-overlapping k-mer tokens and are sent to a sequential embedding layer. The embedding obtained from this layer is sent to the numerical embedding layer for co-training with the sequential embedding. DNAGPT produces two types of embedding, one for classification tasks and another for regression tasks. The model is trained on 200 billion genomic sequences from various species, including mammals and humans. The model can be downstream to various tasks such as binary classification, numerical regression, and sequence generation etc. The model uses the BPE tokenization technique. DNAGPT has two variants, DNAGPT-0.1B and DNAGPT-3B, which can handle 0.1 billion parameters and 3 billion parameters, respectively [16].

4. Materials and Methods

In this research, the DNA sequences for three different viruses are provided as input to pre-trained models, including DNAGPT, DNABERT, DNABERT-2, and GENA-LM, and the models are fine-tuned to enhance their performance. The block diagram in Figure 3

explains the steps involved in DNA sequence classification. The input DNA sequence is processed by dividing it into smaller segments referred to as tokens. Tokenization can be performed using k-mer tokenization or Byte Pair Encoding (BPE) tokenization. DNAGPT, DNABERT-2, and GENA-LM use BPE, while DNABERT uses k-mer tokenization.

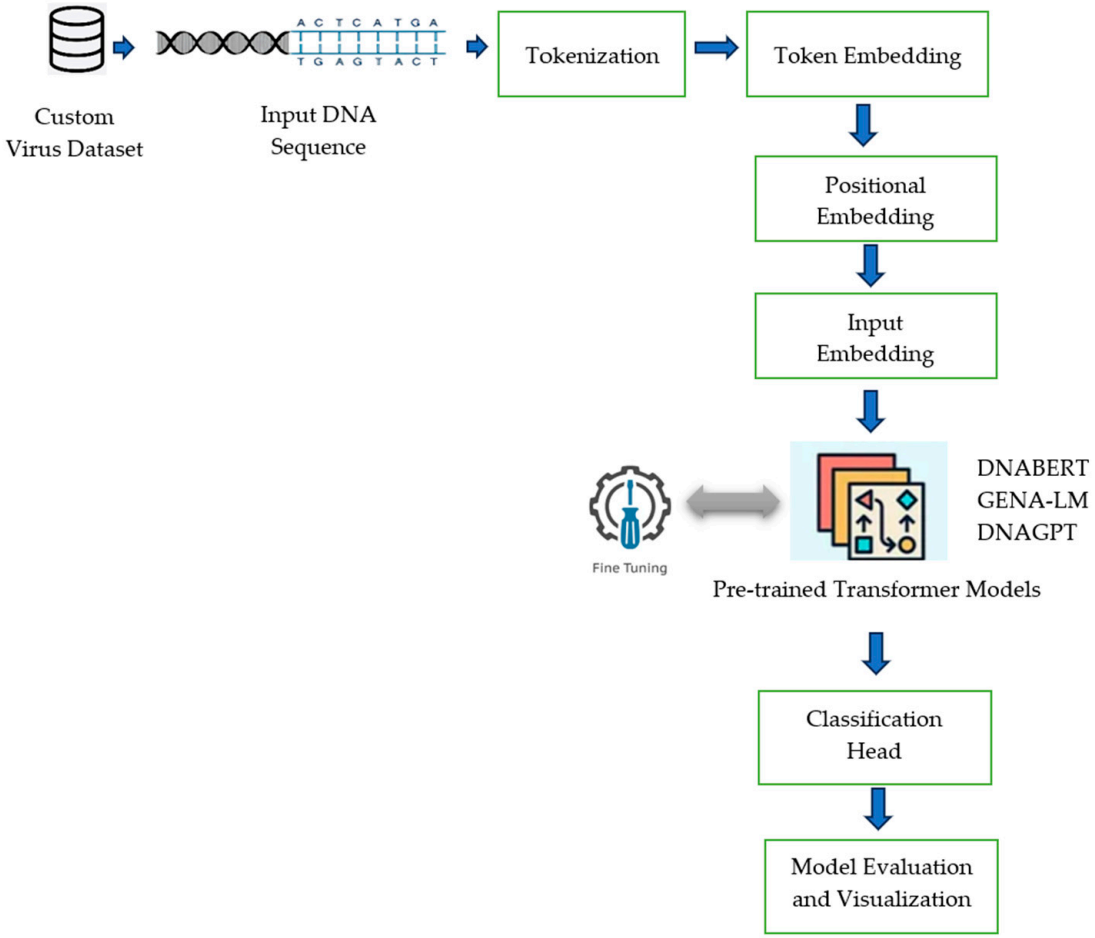


Figure 3. Block diagram.

K-mer tokenization involves breaking the DNA sequence into overlapping subsequences of fixed length k, where k = 2, 3, 4, 5, or 6. This method captures the sequential nature of the DNA, enabling a detailed analysis of local nucleotide patterns preserved throughout the sequence. BPE tokenization, on the other hand, is a compression-based technique initially designed for text data. It iteratively merges the most frequent pairs of characters or sequences into new tokens, thereby reducing the sequence length while retaining essential biological patterns and structures. These tokenization techniques facilitate the conversion of DNA sequences into a format suitable for subsequent processing and analysis. The tokenized DNA sequence is sent to embedding, where each token is converted into a dense vector representation of size 768 that captures its semantic meaning. Each token in the sequence is assigned a unique positional encoding that represents its position in the sequence. By incorporating positional information, the model can better comprehend the structure and context of the DNA sequence, resulting in more accurate predictions and analyses. The combined embedding is passed to the pre-trained Transformer models to obtain the prediction. The detailed steps involved in DNA sequence classification are explained in Algorithm 1.

Algorithm 1: Steps for DNA Sequence Classification

1: Load the DNA sequence dataset
2: Download the pre-trained model and its respective tokenization for the models: DNAGPT, DNABERT-S, DNABERT-2, and GENL-LM from the Hugging Face library.
Model <--- Load the model from the Hugging Face library
3: Apply the tokenizer to convert the input DNA sequence to a sequence of tokens.
4: Embedding: For each tokenized sequence, convert the tokens into dense vectors of size 768.
5: Add the Transformer layers and the task-specific (classification head) to the model.
6: Fine-tune the pre-trained model using a custom dataset to optimize its parameters for the specific classification task.
7: Measure the model's performance using appropriate metrics (e.g., accuracy, F1-score).

5. Experimental Results and Discussion

The experiment was conducted in Google Colab Pro, utilizing a CPU with an 11th Gen Intel(R) Core(TM) i7-1165G7 processor at 2.80 GHz, a GPU with a Tesla T4, and 16.0 GB of RAM, with Python 3.7. The pre-trained models, namely DNABERT, DNABERT-2, DNAGPT, and GENA-LM, are used to evaluate the results. The pre-trained Transformer models are fine-tuned and optimized by adjusting various hyperparameters. These hyperparameters include learning rate, optimizer type, number of training epochs, number of k-folds, and weight decay, as detailed in Table 1.

Table 1. Hyper-parameter for the models.

Parameter	Value
learning rate	5×10^{-6}
training epochs	10
Optimizer	adamw_torch
Stratified K Fold (k-fold)	10
weight_decay	0.01

5.1. Model Evaluation and Performance Metrics

The performance of each pre-trained model is measured using accuracy, precision, recall, F1-score, and ROC curve. Accuracy measures the overall correctness of the model by calculating the ratio of correctly predicted instances to the total number of instances. Precision, also known as positive predictive value, measures the ratio of correctly predicted positive instances to the total predicted positives. Recall, also known as sensitivity or true positive rate, measures the ratio of correctly predicted positive instances to actual positives. The F1-score is the harmonic mean of precision and recall. The formula for calculating accuracy, precision, recall, and F1-score is given in Formulas (2)–(5). The ROC curve plots the true positive rate and false positive rate of classification for different thresholds; thus, it portrays the quality of the model.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

$$\text{F1} - \text{Score} = \frac{2 * \ Precision \ * \ Recall}{Precision + Recall} \tag{5}$$

The performance of the pre-trained models is presented in Table 2. Each model is trained using the cross-validation technique with a 10-fold split, and each fold is trained for 25 epochs. Initially, the model is trained with genome sequences of length 100 base pairs (bp). GENA-LM and DNAGPT achieved the highest accuracy of 95%, compared to DNABERT at 92% and DNABERT-2 at 94%. In terms of precision, recall, and F1 score, DNAGPT outperformed GENA-LM. DNAGPT has a slightly higher precision rate when compared to GENA-LM. The precision scores of both Bat and COVID classes are 0.01 higher than the GENA-LM model. The higher recall rates of the DNAGPT model indicate fewer false negative errors. Additionally, DNAGPT also has a slightly higher F1-score value for all three classes when compared to GENA-LM. To further enhance the model's performance, the evaluation is conducted by increasing the genome sequence length to 150 bp. The model is trained with a 150 bp sequence length for a cross-validation split of 15 folds for 10 epochs. The performance metrics of the pre-trained model, after being trained with 150 bp, are presented in Table 3.

Table 2. Class-wise performance of pre-trained model with cross-fold validation (splits n = 10) with a genome sequence length of 100 bp.

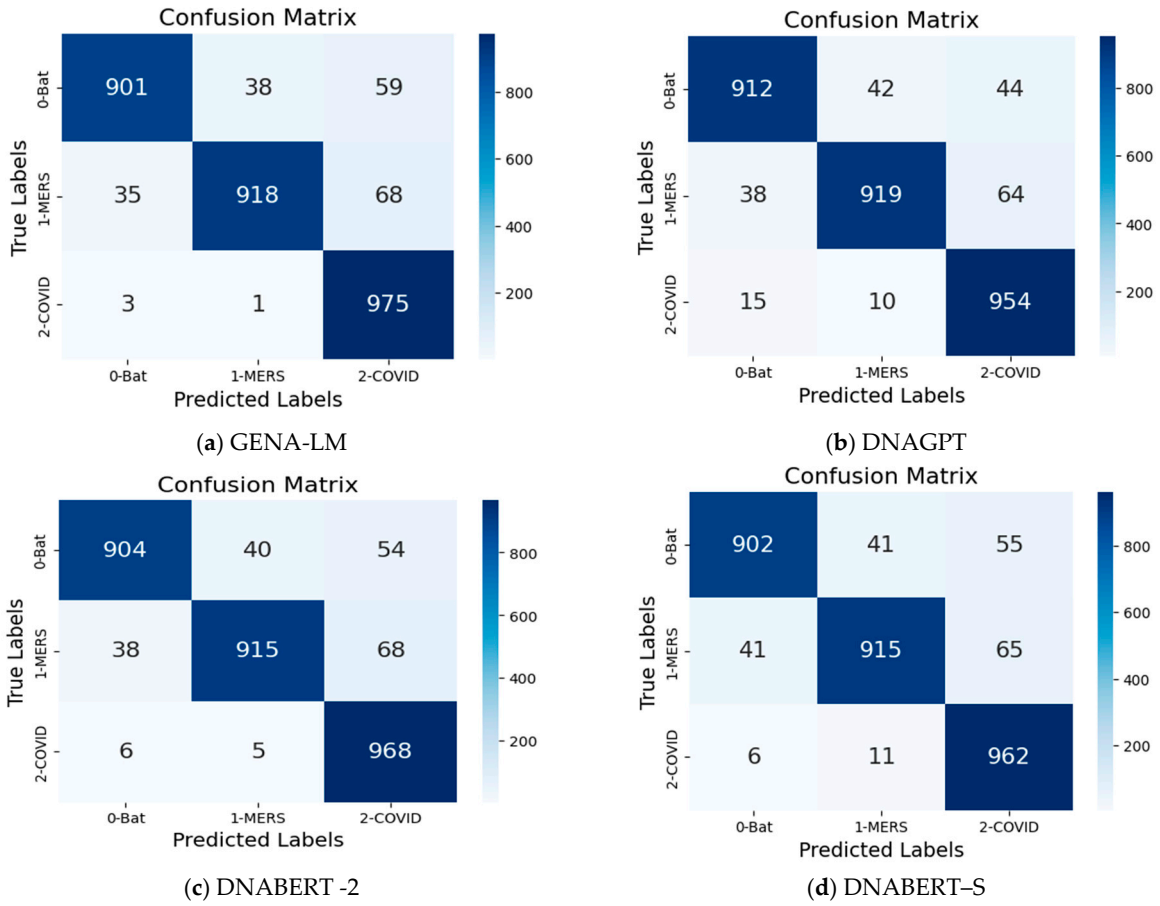
Family	Model	Classes	Performance Metrics			Weighted Average Accuracy
			Precision	Recall	F1	
GENA-LM	BigBird	Bat-Corona	0.96	0.94	0.95	0.95
		MERS	0.96	0.90	0.93	
		COVID-2	0.91	0.99	0.95	
DNAGPT		Bat-Corona	0.97	0.93	0.95	0.95
		MERS	0.97	0.92	0.94	
		COVID-2	0.90	0.99	0.94	
DNABERT	DNABERT	Bat-Corona	0.95	0.90	0.93	0.92
		MERS	0.92	0.90	0.91	
		COVID-2	0.89	0.96	0.92	
	DNABERT-2	Bat-Corona	0.95	0.93	0.94	0.94
		MERS	0.96	0.91	0.93	
		COVID-2	0.91	0.98	0.95	

Table 3. Class-wise performance of pre-trained model trained for cross-fold validation (splits n = 15) with genome sequence of length 150 bp.

Family	Model	Classes	Performance Metrics			Weighted Average Accuracy
			Precision	Recall	F1	
GENA-LM	BigBird	Bat-Corona	0.97	0.92	0.94	0.95
		MERS	0.97	0.92	0.94	
		COVID-2	0.90	1.00	0.95	
DNAGPT		Bat-Corona	0.96	0.97	0.96	0.96
		MERS	0.98	0.91	0.94	
		COVID-2	0.93	0.99	0.96	
DNABERT	DNABERT	Bat-Corona	0.95	0.90	0.93	0.92
		MERS	0.92	0.90	0.91	
		COVID-2	0.89	0.96	0.92	
	DNABERT-2	Bat-Corona	0.96	0.94	0.95	0.95
		MERS	0.97	0.92	0.94	
		COVID-2	0.92	0.99	0.95	

The DNAGPT model achieves an improved accuracy of 96% after increasing the DNA sequence length to 150 bp. Similarly, DNABERT-2 accuracy also improved to 95%, whereas the accuracy of the other two models, GENA-LM and DNABERT, remains the same irrespective of the length of the genome sequence. The advanced contextual understanding of the genomic sequence allowed the DNAGPT model to capture more complex patterns. Additionally, the DNAGPT model is trained with genomes from all mammals, including viruses and bacteria. This specialized training data have enabled the model to exhibit improved performance.

The confusion matrix represents the correctly classified instances in the diagonal and misclassified instances in the off-diagonal. The DNAGPT model has more correctly classified instances for BAT and MERS, while GENA-LM has slightly more correctly classified instances for COVID than DNAGPT. GENA-LM correctly classified more instances of COVID class than other models. The confusion matrix of all the pre-trained models and their variants is shown in Figure 4. The observation that DNAGPT classified MERS and BAT classes well, even after SMOTE-based data augmentation, suggests a greater robustness to learning from synthetic data compared to GENA-LM. GENA-LM's higher accuracy on the COVID class likely reflects the initial imbalance in the dataset, where COVID instances were more abundant. The subsequent addition of synthetic data to the MERS and BAT classes through SMOTE seems to pose a more significant challenge for GENA-LM to learn effective representations, unlike DNAGPT.



(**a**) GENA-LM

(**b**) DNAGPT

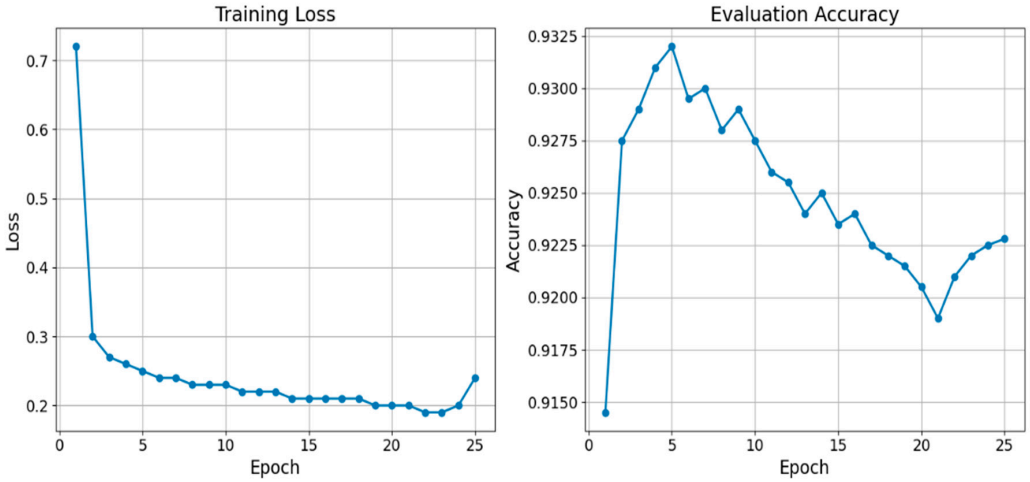
(**c**) DNABERT -2

(**d**) DNABERT–S

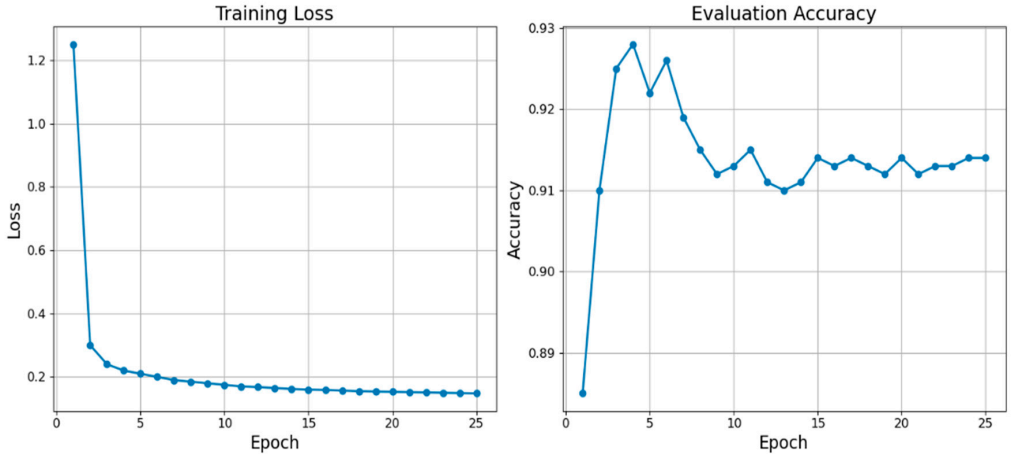
Figure 4. Confusion matrix of the pre-trained model.

The training loss and validation accuracy for the pre-trained models and their variants are shown in Figure 5a–d. During training, as shown in Figure 5a, the GENA-LM model exhibits fluctuations in loss across epochs. Initially, the training loss decreased; however,

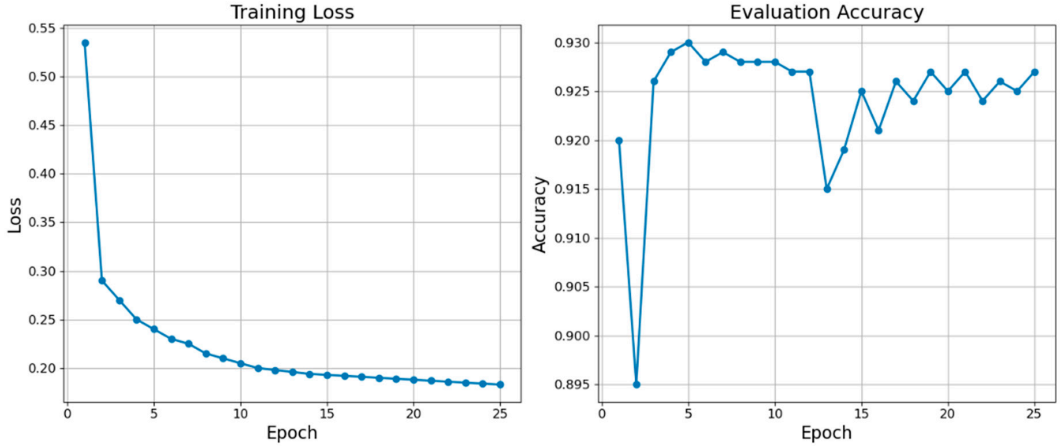
in later epochs, after epoch 23, it began to increase. In contrast, the DNAGPT, DNABERT, and DNABERT-2 models, as shown in Figure 5b–d, exhibit a gradual and steady decrease in training loss over the epochs. Additionally, the validation accuracy for the DNABERT model steadily increases, whereas for GENA-LM and DNAGPT, the validation accuracy fluctuates, with periods of increase followed by a drop.



(**a**) DNAGPT

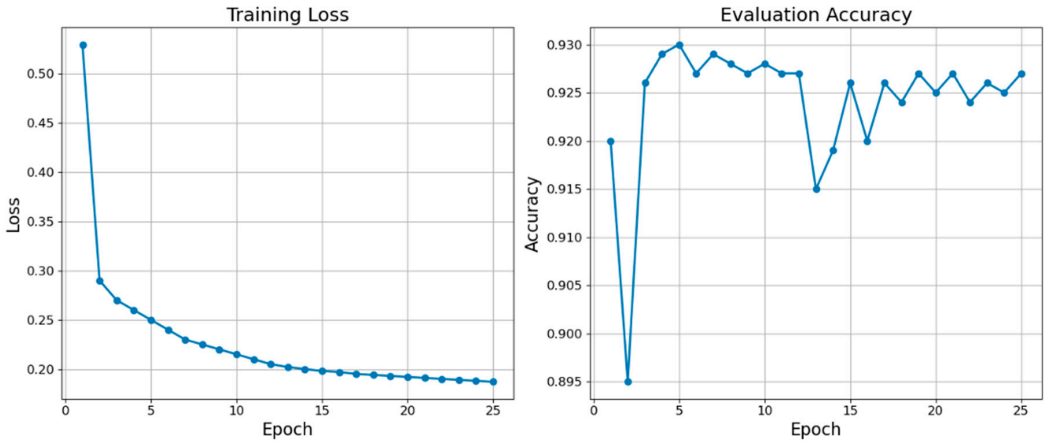


(**b**) GENALM



(**c**) DNABERT

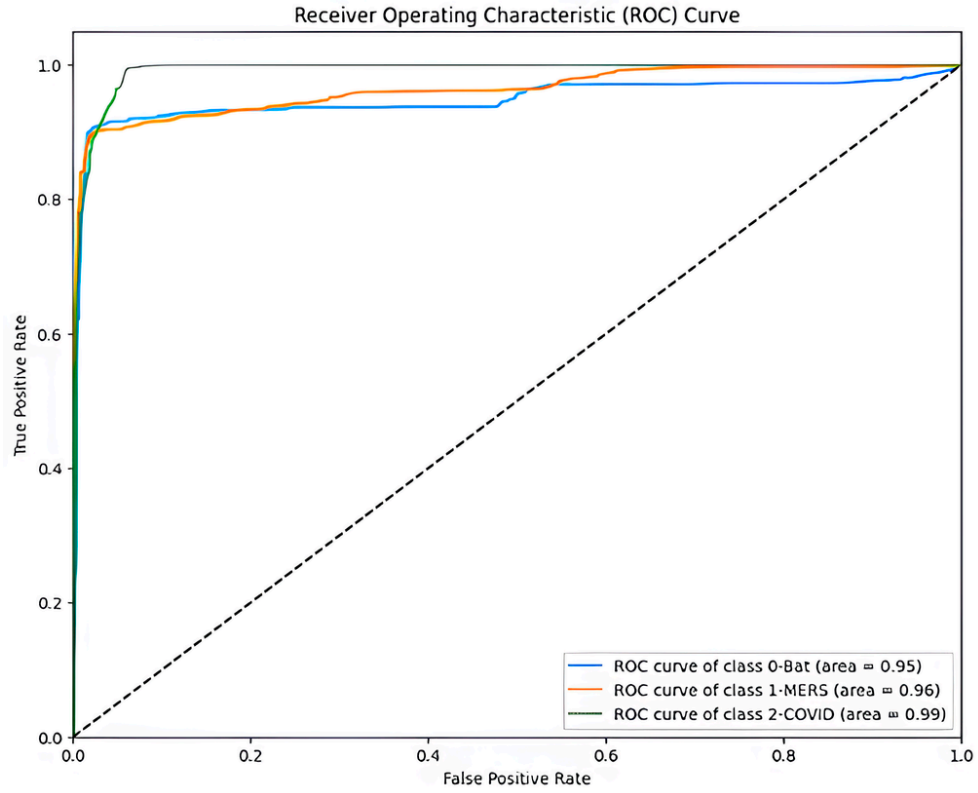
Figure 5. *Cont.*



(**d**) DNABERT2

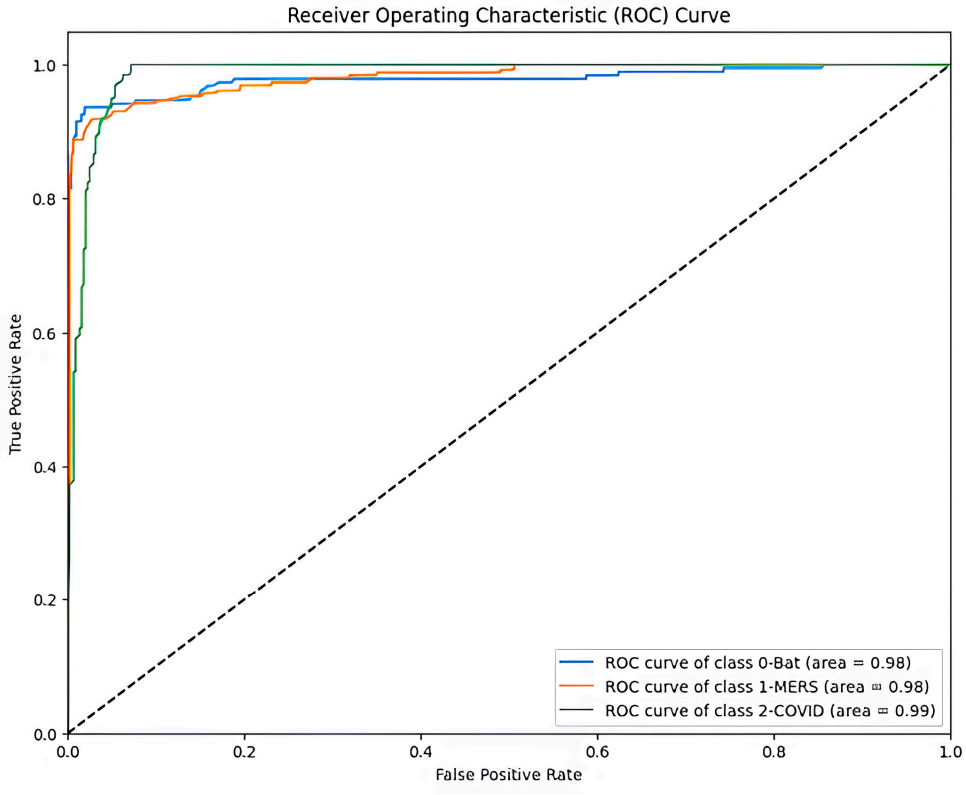
Figure 5. Training loss and validation accuracy.

The Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve is a performance measurement for classification models at various threshold settings. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) as shown in Figure 6, and measures the model's ability to distinguish between classes. A higher AUC value indicates that the model can better differentiate between positive and negative classes. The DNAGPT model has the highest AUC value for all three classes: 0.99 for the Bat class, 0.98 for the MERS class, and 1.00 for the COVID class, which is the highest of all the other models.

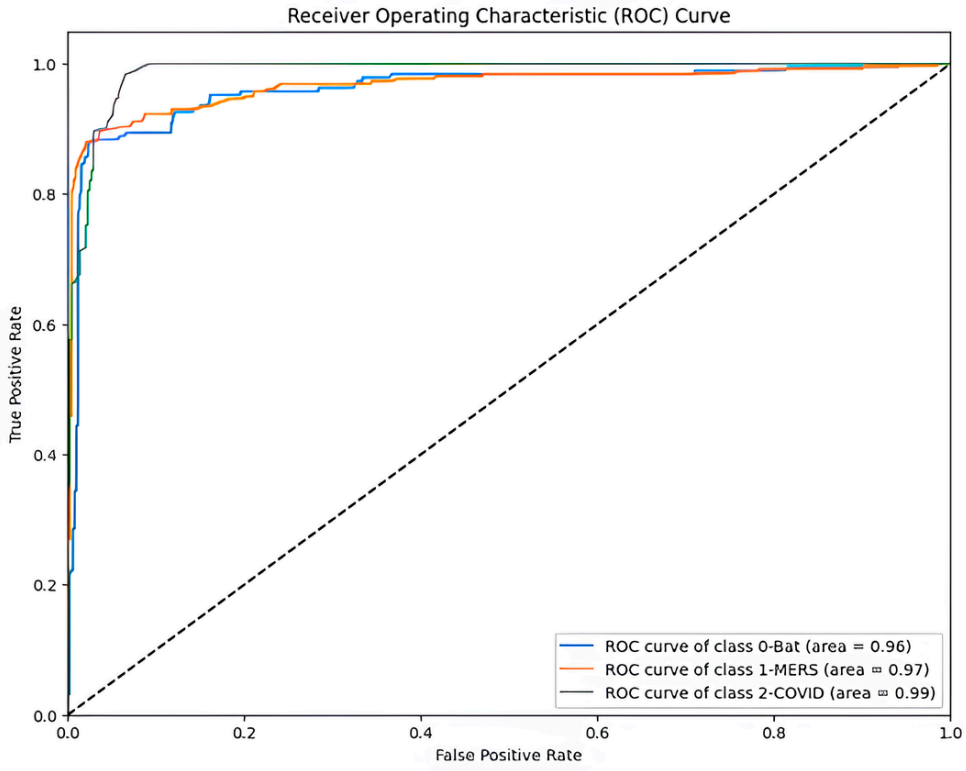


(**a**) GENA-LM

Figure 6. *Cont.*

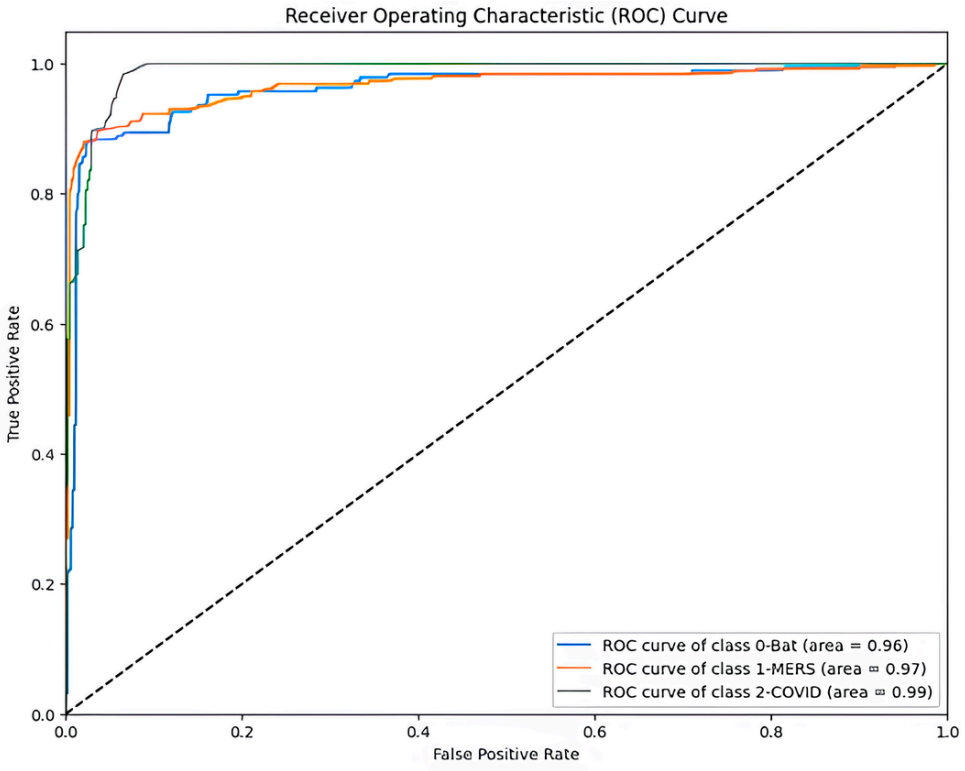


(**b**) DNAGPT



(**c**) DNABERT

Figure 6. *Cont.*



(**d**) DNABERT2

Figure 6. AUC for the pre-trained LLM.

5.2. Model Comparison

This section compares the proposed pre-trained DNAGPT model with state-of-the-art models in machine learning and deep learning. The DNAGPT model achieved 96% accuracy on a three-class dataset, outperforming an LSTM model, which achieved 93% accuracy on a four-class dataset. In contrast, machine learning models such as XGBoost and SVM achieved accuracies of 88.82% and 92.5%, respectively. Furthermore, another LLM, Nucleic Transform, achieved 88% accuracy in E. coli classification tasks. Overall, DNAGPT demonstrates a significant performance improvement compared to deep learning, machine learning, and other large language models (LLMs) in classification tasks, as shown in Table 4.

Table 4. Model comparison.

Authors and Years	Method	Dataset	Accuracy
Zhang X et al. [23]	XGBoost	DNA chromosome	0.88
Hamed, B.A et al. [6]	SVM Sigmoid	DNA dataset	0.92
Gunasekaran et al. [10]	Deep learning—LSTM	Four-class dataset	0.93
Shujun He et al. [18]	Nucleic Transformer	E. Coli Classification	0.88
Zhihan Zhou et al. [13]	DNABERT-2	Transcription Factor Prediction (Mouse)	0.92
R. K. Mohammed et al. [24]	U-Net	Three-class dataset	0.96
Proposed pre-trained LLM	LLM—DNAGPT	Three-class dataset	0.96

6. Conclusions

In this study, we evaluated the performance of pre-trained LLMs and their variants —DNABERT, DNABERT-2, GENA-LM, and DNAGPT—on a viral three-class dataset obtained from NCBI. Among these models, DNAGPT demonstrated the highest accuracy of 0.96. The experiments utilized genomic sequences of length 150 nucleotides and were trained over cross-validation with split $n = 15$. Various metrics, including accuracy, loss curves, confusion matrices,

and ROC curves, were analyzed. The results indicate that the performance of these models can be further enhanced through fine-tuning and by increasing the number of base pairs in the sequences. However, it is essential to note that increasing the number of base pairs directly impacts the model's runtime and storage requirements. By employing these DNA-specific language models, our research seeks to advance the field of bioinformatics, providing tools that can more effectively classify and interpret complex genomic information. Through this methodology, we aim to contribute to a deeper understanding of genetic data, facilitating advancements in personalized medicine, genetic research, and biotechnology.

Author Contributions: Conceptualization, H.G. and U.S.; methodology, H.G.; software, N.R.W.B.; validation, N.R.W.B. and U.S.; formal analysis, M.S.H.; investigation, M.S.H.; resources, N.R.W.B.; data curation, H.G.; writing—original draft preparation, H.G.; writing—review and editing, U.S.; visualization, H.G.; supervision, N.R.W.B.; project administration, H.G.; funding acquisition, H.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research project was funded by the University of Technology and Applied Sciences through the Internal Research Funding Program, grant no. IRFP-IBRI-24-15.

Data Availability Statement: The original data presented in the study are openly available in [GitHub] at [https://github.com/hema2107/DNA-Sequence-Dataset] accessed on 9 March 2025.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Qayyum, A.; Benzinou, A.; Saidani, O.; Alhayan, F.; Khan, M.A.; Masood, A.; Mazher, M. Assessment and classification of COVID-19 DNA sequence using pairwise features concatenation from multi-transformer and deep features with machine learning models. *J. Assoc. Lab. Autom.* **2024**, *29*, 100147. [CrossRef]
2. Bento, A.I.; Nguyen, T.; Wing, C.; Lozano-Rojas, F.; Ahn, Y.; Simon, K. Evidence from internet search data shows information-seeking responses to news of local COVID-19 cases. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 11220–11222. [CrossRef] [PubMed]
3. Karlicki, M.; Antonowicz, S.; Karnkowska, A. Tiara: Deep learning-based classification system for eukaryotic sequences. *Bioinformatics* **2022**, *38*, 344–350. [CrossRef] [PubMed]
4. Klapproth, C.; Sen, R.; Stadler, P.F.; Findeiß, S.; Fallmann, J. Common Features in lncRNA Annotation and Classification: A Survey. *Non-Coding RNA* **2021**, *7*, 77. [CrossRef]
5. Yang, A.; Zhang, W.; Wang, J.; Yang, K.; Han, Y.; Zhang, L. Review on the Application of Machine Learning Algorithms in the Sequence Data Mining of DNA. *Front. Bioeng. Biotechnol.* **2020**, *8*, 1032. [CrossRef]
6. Hamed, B.A.; Ibrahim, O.A.S.; El-Hafeez, T.A. Optimizing classification efficiency with machine learning techniques for pattern matching. *J. Big Data* **2023**, *10*, 124. [CrossRef]
7. Rahman, A.; Zaman, S.; Das, D. Cracking the Genetic Codes: Exploring DNA Sequence Classification with Machine Learning Algorithms and Voting Ensemble Strategies. In Proceedings of the 2024 International Conference on Advances in Computing, Communication, Electrical, and Smart Systems (iCACCESS), Dhaka, Bangladesh, 8–9 March 2024; pp. 1–6. [CrossRef]
8. Nguyen, N.G.; Tran, V.A.; Ngo, D.L.; Phan, D.; Lumbanraja, F.R.; Faisal, M.R.; Abapihi, B.; Kubo, M.; Satou, K. DNA Sequence Classification by Convolutional Neural Network. *J. Biomed. Sci. Eng.* **2016**, *9*, 280–286. [CrossRef]
9. Gomes, J.C.; Masood, A.I.; Silva, L.H.d.S.; Ferreira, J.R.B.d.C.; Júnior, A.A.F.; Rocha, A.L.d.S.; de Oliveira, L.C.P.; da Silva, N.R.C.; Fernandes, B.J.T.; dos Santos, W.P. Covid-19 diagnosis by combining RT-PCR and pseudo-convolutional machines to characterize virus sequences. *Sci. Rep.* **2021**, *11*, 11545. [CrossRef]
10. Gunasekaran, H.; Ramalakshmi, K.; Arokiaraj, A.R.M.; Kanmani, S.D.; Venkatesan, C.; Dhas, C.S.G. Analysis of DNA Sequence Classification Using CNN and Hybrid Models. *Comput. Math. Methods Med.* **2021**, *2021*, 1835056. [CrossRef]
11. Choi, S.R.; Lee, M. Transformer Architecture and Attention Mechanisms in Genome Data Analysis: A Comprehensive Review. *Biology* **2023**, *12*, 1033. [CrossRef]
12. Ji, Y.; Zhou, Z.; Liu, H.; Davuluri, R.V. DNABERT: Pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics* **2021**, *37*, 2112–2120. [CrossRef] [PubMed]
13. Zhou, Z.; Ji, Y.; Li, W.; Dutta, P.; Davuluri, R.; Liu, H. DNABERT-2: Efficient Foundation Model and Benchmark for Multi-Species Genome. *arXiv* **2023**, arXiv:2306.15006v2.

14. Kabir, A.; Bhattarai, M.; Peterson, S.; Najman-Licht, Y.; Rasmussen, K.Ø.; Shehu, A.; Bishop, A.R.; Alexandrov, B.; Usheva, A. DNA breathing integration with deep learning foundational model advances genome-wide binding prediction of human transcription factors. *Nucleic Acids Res.* **2024**, *52*, e91. [CrossRef]
15. Fishman, V.; Kuratov, Y.; Shmelev, A.; Petrov, M.; Penzar, D.; Shepelin, D.; Chekanov, N.; Kardymon, O.; Burtsev, M. GENA-LM: A family of open-source foundational DNA language models for long sequences. *Nucleic Acids Res.* **2025**, *53*, gkae1310. [CrossRef]
16. Zhang, D.; Zhang, W.; Zhao, Y.; Zhang, J.; He, B.; Qin, C.; Yao, J. DNAGPT: A Generalized Pre-trained Tool for Versatile DNA Sequence Analysis Tasks. *arXiv* **2023**, arXiv:2307.05628.
17. Nguyen, E.; Poli, M.; Faizi, M.; Thomas, A.; Birch-Sykes, C.; Wornow, M.; Patel, A.; Rabideau, C.; Massaroli, S.; Bengio, Y.; et al. HyenaDNA: Long-Range Genomic Sequence Modeling at Single Nucleotide Resolution. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 43177–43201.
18. He, S.; Gao, B.; Sabnis, R.; Sun, Q. Nucleic Transformer: Classifying DNA Sequences with Self-Attention and Convolutions. *ACS Synth. Biol.* **2023**, *12*, 3205–3214. [CrossRef]
19. Wang, Z.; Wang, Z.; Jiang, J.; Chen, P.; Shi, X.; Li, Y. Large Language Models in Bioinformatics: A Survey. *arXiv* **2025**, arXiv:2503.04490.
20. Sarumi, O.A.; Heider, D. Large language models and their applications in bioinformatics. *Comput. Struct. Biotechnol. J.* **2024**, *23*, 3498–3505. [CrossRef]
21. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
23. Zhang, X.; Beinke, B.; Kindhi, B.A.; Wiering, M. Comparing Machine Learning Algorithms with or without Feature Extraction for DNA Classification. *arXiv* **2020**, arXiv:2011.00485.
24. Mohammed, R.K.; Alrawi, A.T.H.; Dawood, A.J. U-Net for genomic sequencing: A novel approach to DNA sequence classification. *Alex. Eng. J.* **2024**, *96*, 323–331. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.