Journal of Advanced Zoology
ISSN: 0253-7214
Volume 45 Issue 3 Year 2024 Page 776-788

Deciphering Genetic Overlaps: A Comprehensive Study On Viral Host Determination Using Machine Learning And Deep Learning Models

Pankaj Agarwal[1*], Sapna Yadav[2]

[1*]*K.R Mangalam University, Gurgaon, pankaj.agarwal7877@gmail.com*
[2]*Jamia Millia Islamia, Delhi, sapnayadav0821@gmail.com*

***Corresponding Author:** Pankaj Agarwal*
**K.R Mangalam University, Gurgaon, pankaj.agarwal7877@gmail.com*

Abstract

The study uses machine learning and deep learning models to study the intricate relationship between viral genetic DNA sequences and host organisms. It uses a comprehensive dataset from databases like ExPASy and NCBI, which encodes crucial genetic information for viral replication.

The study aimed to create a viral DNA dataset and develop robust machine learning and deep learning models to classify viruses into eight host categories. Despite extensive experimentation using various models, performance improvement was elusive due to genetic overlaps. Viral genomes from different classes had significant shared genetic sequences, making it difficult for these models to identify unique class-specific features, blurring the lines of differentiation.

The study reduced the number of classes from eight to three, focusing on plants, animals, and microorganisms. This resulted in improved evaluation metrics, with the Random Forest Machine learning model reaching a maximum accuracy of 70% and the LSTM deep learning model surpassing 85%, overcoming earlier challenges.

The discovery that viral genomes from different classes share significant genetic overlaps challenges conventional molecular distinctions, emphasizing the complexity of molecular differentiation in viral genomes. This pragmatic approach aligns molecular understanding with genetic data in viral host determination.

Key Terms: Machine Learning; Deep Learning; Viral Host Specificity; DNA Sequences; Disease Surveillance.

1. Introduction

Understanding viral dynamics and host organism interactions is crucial in virology and molecular biology. Advancements in technology, including machine learning and deep learning, offer new opportunities to unravel these complexities.

This work aims to develop a predictive model using advanced machine learning and deep learning models to determine virus host organisms based on genetic DNA sequences, using a comprehensive dataset from databases like ExPASy and NCBI.This research aims to generate a vast viral DNA dataset and develop machine learning and deep learning models to classify viruses into eight host categories, reflecting the diversity of viruses and their potential hosts, reflecting the molecular tapestry of life.

Machine learning models like Decision Trees, Random Forest, Naïve Bayes, KNN, and deep learning models like CNN and LSTM face a challenge due to genetic overlaps among different viral classes. These shared genetic sequences make it difficult for conventional models to identify unique class-specific features.
The research reduced the number of classes from eight to three, focusing on plants, animals, and microorganisms, to better understand viral genetic information. This approach aligns molecular understanding with genetic data, resulting in improved metrics and clarity in viral genome classification.
The discovery that viral genomes from different classes share significant genetic overlaps challenges traditional molecular distinctions and emphasizes the need for advanced computational approaches to navigate genetic data. This not only advances computational virology but also offers a nuanced perspective on viral host classification.
Advanced computational methodologies and molecular insights will enhance our understanding of viruses and host organisms, refining predictive models and contributing to scientific discourse on viral-host interactions.

1.1 The Significance of Virus Host Categorization

The categorization of viruses based on their host organisms holds paramount significance in elucidating the intricate dynamics of viral infections and advancing our comprehension of host-virus interactions. Accurate classification allows for targeted research on specific host groups, aiding in the development of tailored therapeutic interventions and preventive measures. Moreover, it facilitates precise disease surveillance, enabling a proactive response to potential outbreaks. The proposed work's significance lies in its potential to refine the categorization process, shedding light on the shared genetic elements among different viral classes and providing a nuanced understanding of viral diversity, crucial for both scientific research and public health initiatives.

1.2 Data Set & Domain

Our objective is to prepare the data to develop classification models involving eight host classes, namely humans, plants, vertebrates, invertebrates, bacteria, eukaryotic microorganisms, archaea, and fungi. The corresponding DNA sequences of viruses belonging to these classes were collected from reputable databases such as ExPASy (operated by the SIB Swiss Institute of Bioinformatics,(https://www.expasy.org/) and NCBI (National Center for Biotechnology Information,(https://www.ncbi.nlm.nih.gov/). These databases serve as valuable repositories of viral genomic data, enabling us to compile a comprehensive and diverse dataset for training and evaluating our machine learning models. By leveraging these reliable sources, we aim to ensure the data's quality, integrity, and relevance, which are critical factors in achieving accurate and robust predictions of viral host specificity.

The dataset consists of 280 rows and 2 columns

classes	labels
archae	0
bacteria	1
eukartoyic_microorganisms	2
humans	3
invertebrates	4
plants	5
vertebrates	6
fungi	7

Figure 1: Host Classes under consideration

2. Literature Survey

The referenced research articles collectively contribute to the field of predicting viral host specificity through machine learning and deep learning approaches. Several key themes emerge from the synthesis of these works.

Reference	Work Summary	Methodology Used	Limitations
Zheng et al., 2018 [1]	Exploration of virus-host protein interactions through feature extraction and machine learning.	Feature extraction, machine learning approaches.	- Sensitivity to choice of features. - Impact of feature selection on model robustness.
Cho and Won, 2003 [2]	Application of machine learning in DNA microarray analysis for cancer classification.	Machine learning for DNA microarray analysis.	- Reliance on curated datasets may introduce biases. - Sensitivity to the quality and representativeness of training data.
Nguyen et al., 2016 [3]	DNA sequence classification using Convolutional Neural Network (CNN).	CNN for DNA sequence classification.	- Performance sensitive to dataset quality. - Potential bias in models due to training data limitations.
Tampuu et al., 2019 [4]	ViraMiner: Deep learning on raw DNA sequences for identifying viral genomes in human samples.	Deep learning, convolutional neural networks (CNNs) on raw DNA sequences.	Lack of interpretability in deep learning models. - Potential overfitting to specific virus types.
Santoso et al., 2022[5]	Systematic literature review on virus prediction based on DNA sequences using machine learning and deep learning.	Literature review, synthesis of machine learning and deep learning methods.	Relies on existing studies with varied methodologies. - Limited control over the quality and consistency of source studies.
Muflikhah et al., 2022 [6]	Profiling DNA sequence of SARS-Cov-2 virus using machine learning algorithm.	Machine learning algorithm for profiling SARS-CoV-2 DNA sequences.	Potential challenges in adapting the model to emerging SARS-CoV-2 variants. - Generalization limitations to other viruses.
Chaturvedi et al., 2023[7]	PREHOST: Host prediction of coronaviridae family using machine learning.	Machine learning for predicting hosts of coronaviridae family.	Limited to coronaviridae family, may lack generalizability. - Dependence on available data for training.
Kwon et al., 2019[8]	Study on host tropism determinants of influenza virus using machine learning.	Machine learning for identifying host tropism determinants.	Challenges in adapting models to evolving influenza strains. - Complex host-virus interactions not fully captured by models.
Xu and Wojtczak, 2022[9]	Dive into machine learning algorithms for influenza virus host prediction with hemagglutinin sequences.	Machine learning algorithms for influenza virus host prediction using hemagglutinin sequences.	Model adaptability challenges to new influenza strains. - Sensitivity to sequence variations in hemagglutinin.
Salama et al., 2016[10]	Prediction of virus mutation using neural networks and rough set techniques.	Neural networks, rough set techniques for predicting virus mutation.	Generalization limitations to diverse viruses. - Dependence on the availability and diversity of mutation data.
Eng et al., 2017 [11]	Predicting zoonotic risk of influenza A viruses from host tropism protein signature using random forest.	Random forest for predicting zoonotic risk of influenza A viruses.	-Challenges in predicting zoonotic risk due to evolving viral strains. - Dependence on the quality of input protein signature data.
Ghosh et al., 2022 [12]	Application of Machine Learning in Understanding Plant Virus Pathogenesis.	Application of machine learning in plant virus pathogenesis.	Limited availability of comprehensive plant virus datasets. - Challenges in generalizing models from human to plant viruses.
Barman et al., 2014 [13]	Prediction of interactions between viral and host proteins using supervised machine learning methods.	Supervised machine learning methods for predicting viral-host interactions.	Sensitivity to the quality and diversity of interaction data. - Potential biases in curated datasets.
Qiang et al., 2018 [14]	Scoring amino acid mutations to predict avian-to-human transmission of avian influenza viruses.	Scoring amino acid mutations for predicting avian-to-human transmission.	Limited predictability of avian-to-human transmission based on amino acid mutations. - Sensitivity to the quality and completeness of mutation data.
Agor and Özaltın, 2018 [15]	Models for predicting the evolution of influenza to inform vaccine strain selection.	Models for predicting influenza evolution to inform vaccine strain selection.	Challenges in predicting influenza evolution due to constantly evolving viral strains. - Dependence on the quality and diversity of input data.
Phute et al., 2021 [16]	A Survey on Machine Learning in Lithography.	Survey on machine learning applications in lithography.	Relies on existing studies with varied methodologies. - Limited control over the quality and consistency of source studies.

Despite the advancements presented in these studies, certain limitations are notable. One of the key limitations of above referenced works is that authors have not tried with all the categories of host viruses. Any host viruses on this earth can be classified into eight discrete host categories, namely humans, plants, vertebrates, invertebrates, bacteria, eukaryotic microorganisms, archaea, and fungi. Authors have tried limited classes of host viruses in their research. This is the key motivation behind this undertaken research work.

3. Methodology

Our journey commences with the preprocessing of DNA fasta sequences to extract pertinent features that encapsulate essential genetic attributes. Importantly, meticulous attention was devoted to ensuring balanced representation across these eight host classes. In addition to the initial eight-class classification, the project explores reduced classifications, with three classes. However, in this work we have focused only on 8 classes.

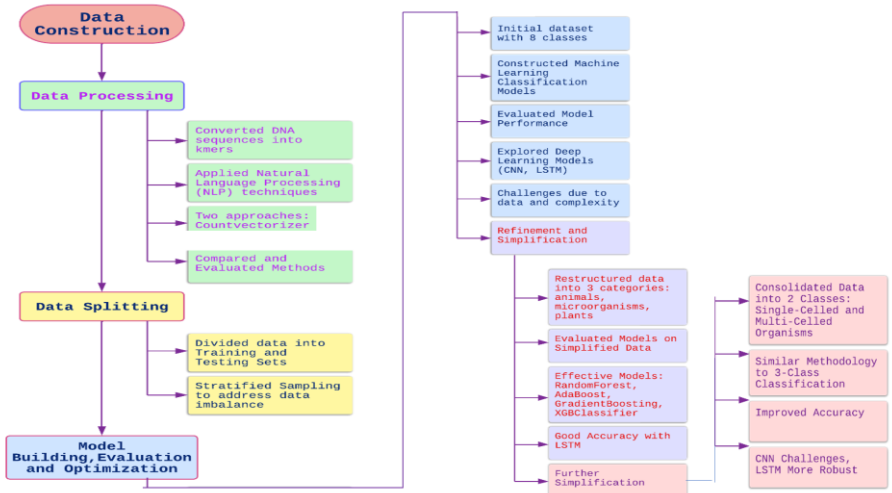

Figure 2: Model Development Phases

Figure 3: Data Curation Steps

3.1 Data Preprocessing:

3.1.1 Data Cleaning: The overall strategy applied for data cleaning is to remove characters from the DNA sequences that are either non-standard or do not convey specific nucleotide information. The newline character '\n' is removed to ensure that sequences are in a clean, continuous format. Ambiguous characters like 'Y', 'M', 'R', 'N', 's', 'S', 'K', and 'W' are removed to standardize the representation of nucleotides (A, T, G, C). The result is that the 'DNA_sequences' column contains cleaned and standardized DNA sequences, making them suitable for further analysis or modeling.

3.1.2 Encoding the hosts columns: The label encoding process is applied to the "hosts" column from the data, which presumably contains categorical host labels (e.g., "human," "vertebrates," etc.). After the label encoding is complete, the encoded labels are stored in the variable y.The y variable now contains numerical representations of the original categorical host labels. These numerical labels can be used as target variables in machine learning models.

3.1.3 Converting Sequences into Kmers: As the length of biological sequences varies by species and has distinct patterns, we extract **k-mers** from the sequences to understand and group the biological data based on the extracted k-mers present across all the sequences in the data set. This process of the creation and grouping of k-mers forms a relationship between the sequences and helps in the categorisation of the species. K-mers are widely used in tasks like sequence analysis, genome assembly, and feature extraction because they capture local patterns within DNA sequences.

3.1.4 Generate a Document term Matrix: From the processed samples of the k-mer strings, a document term matrix is created. A document term matrix is a representation of text data in the numeric form. In order to create a document term matrix, CountVectorizer is used from the python sklearn library. We fit the processed data into the CountVectorizer. It creates a matrix based on the count of distinct k-mers for the entire fitted data.

3.1.5 Labelling the target variable: The target consists of 8 different classes namely archaea, bacteria, eukaryotic microorganisms, humans, invertebrates, plants, vertebrates and fungi. These classes are labelled using a Label encoder from the sklearn library.This data of the document term matrix of DNA sequences along with the target variable of class labels are used to further build the models. To address data imbalance, Stratified K-Fold cross-validation was employed, yielding varying model performance.

3.1.6 Splitting the data into train and test split: The data was split into train and test split with a test size of 0.3

3.1.7 Data Transformation: Using a Count Vectorizer for data transformation, which involves classifying DNA sequences into different host categories, was found to be a suitable choice.

4. Model Building

For classification of viral hosts based on DNA sequences, the choice of the model depends on factors like the size of dataset, the complexity of the problem, and the computational resources available.

4.1 Machine Learning Models: For this work, we have used following ML models.

- Random Forest: Random Forest is an ensemble learning method that is robust, handles high-dimensional data well, and is resistant to overfitting. It's a good choice for both small and large datasets.
- K-Nearest Neighbors (K-NN): K-NN is a simple yet effective classification algorithm. It's particularly useful when much about the data distribution is unknown and can work with small to moderately sized datasets.
- Gradient Boosting Models (e.g., XGBoost): These are ensemble methods that often perform very well in classification tasks. They are known for their speed and accuracy and can handle both small and large datasets.
- Multinomial Naive Bayes: Naive Bayes classifiers are simple and work well with high-dimensional data. They are particularly useful when dealing with text-based or sequence data.
- Decision Trees: Decision trees are interpretable and can be useful for understanding feature importance. They can be used as standalone models or within ensemble methods like Random Forest.

4.2 Deep Learning Models:

4.2.1 CNN: CNNs are well-suited for tasks involving image and sequence data, making them a natural choice for analysing DNA sequences. They can capture local patterns and motifs within the DNA sequences, which can be informative for predicting viral host specificity. CNNs have shown impressive performance in various bioinformatics tasks, including DNA sequence classification, and can automatically learn hierarchical features from the data. For data transformation, word embedding was used. Word embeddings are typically considered a better method than count vectorization for deep learning models when working with text data.

Model building:

- **Model Architecture:**
  - **Input Layer:** The model takes input sequences of length 200.
  - **Hidden Layer 1:** A dense layer with 100 units and ReLU activation function, which introduces non-linearity into the model.
  - **Hidden Layer 2:** Another dense layer with 50 units and ReLU activation function.
  - **Output Layer:** The final layer with 8 units (equal to the number of classes) and a softmax activation function, which provides probability distribution over the classes.

- **Loss Function:** The model uses the "sparse_categorical_crossentropy" loss function, which is commonly used for multi-class classification tasks where the target values are integers (class labels).

- **Optimizer:** The "adam" optimizer is used for gradient-based optimization during training. Adam is an efficient and commonly used optimizer for deep learning.

- **Metrics**: **The model is evaluated using two metrics during training:**

- **"accuracy":** This metric calculates the accuracy of the model's predictions on the training and validation datasets.

- **"sparse_categorical_accuracy":** This is a specific accuracy metric for the parse_categorical_crossentropy" loss, ensuring that it handles integer target values correctly.

➢ **Training Configuration:**

- **Number of Epochs:** The model is trained for 35 epochs, indicating the number of times the entire training dataset is processed.

- **Batch Size:** The training data is divided into batches of 32 samples for each update of the model's weights.

- **Verbose:** It specifies whether to display training progress during each epoch (1 for display, 0 for no display).

- **Validation Data:** The validation data (X_test and y_test) is used to evaluate the model's performance after each epoch.

4.2.2 LSTM: LSTM (Long Short-Term Memory) is specifically designed for sequential data, which is a natural fit for DNA sequences. The ability of LSTM to process sequences of variable lengths and capture long-range dependencies between DNA bases can be crucial for predicting viral host specificity. LSTM models can be initialised with pre-trained embeddings, which can capture useful information about DNA bases from large-scale genomic datasets. Transfer learning from related tasks can be leveraged to improve model performance, even with limited viral genomic data.

Model building:

This model architecture leverages word embeddings to convert text data into numerical form, processes it through two LSTM layers to capture sequential patterns, and then uses a dense layer for classification. It's a common choice for text classification tasks and can be further tuned and optimized based on the specific dataset and problem requirements.

➢ **Embedding Layer:** The model starts with an Embedding layer. This layer is responsible for converting input text data (sequences of words) into dense numerical vectors. The input_dim parameter is set to the size of the vocabulary, which is determined by the number of unique words in your dataset plus one (to account for out-of-vocabulary words). The output_dim parameter specifies the dimensionality of the dense embedding vectors, in this case, 32. The input_length parameter defines the length of input sequences, which is set to max_sequence_length.

➢ **LSTM Layers:** Two LSTM (Long Short-Term Memory) layers are stacked on top of the embedding layer. LSTM is a type of recurrent neural network (RNN) that is well-suited for sequence data like text. The first LSTM layer has 128 units and is set to return sequences (return_sequences=True). This means it produces sequences of outputs for each time step in the input sequence. The second LSTM layer has 64 units and operates in the default mode, which returns only the final output for each sequence.

➢ **Dense Layer:** A Dense layer with 8 units and a softmax activation function is added as the output layer. This layer is responsible for classifying input text into one of eight possible classes (hosts).

➢ Compilation: The model is compiled with the categorical cross-entropy loss function, which is commonly used for multi-class classification tasks. The optimizer used is Adam, a popular choice for gradient-based optimization. The model also tracks the accuracy metric during training.

➢ **Training:** The model is trained using the training data (X_train and y_train) for 35 epochs with a batch size of 64. During training, the model learns to map input sequences to their corresponding host labels. The validation data (validation_split=0.2) is used to monitor the model's performance on unseen data and prevent overfitting.

5. Experimental Results

Our dataset has 35 sequences for each considered class of virus

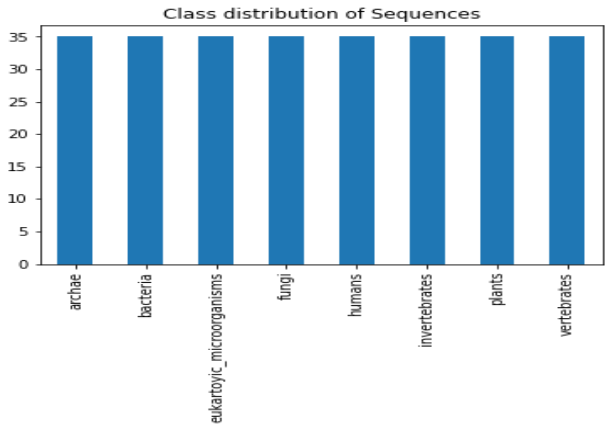

Figure 4: 8 classes of considered Host Viruses

5.1 Performance of Machine Learning Models

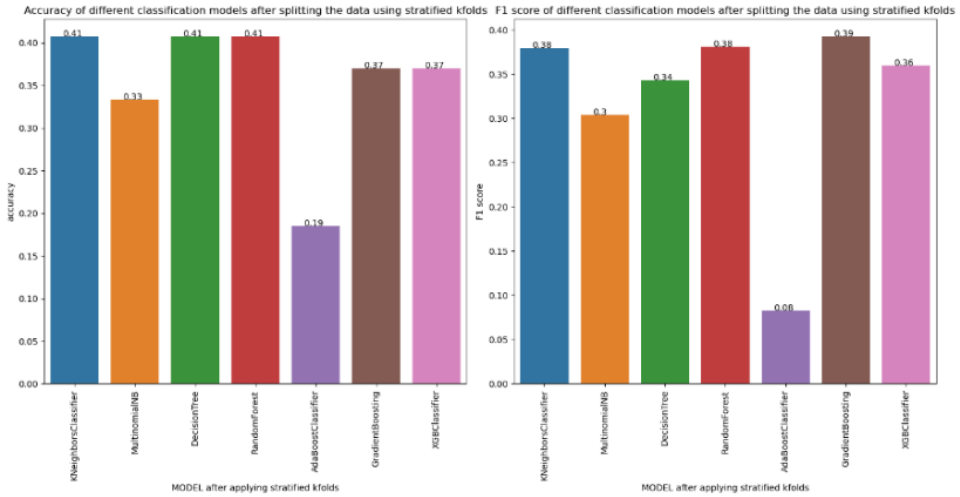

Figure 5: Performance metrics of applied Algorithms

The 8-class classification models exhibited varying degrees of performance, with accuracies ranging from approximately 11.90% to 42.86%. Multinomial Naive Bayes, Random Forest, Gradient Boosting, and XGBoost showed better accuracy and precision compared to other models, while AdaBoost performed poorly. Among all the models, KNearestNeighbors and RandomForest algorithms have done a better job at the classification.

Evaluation metrics of models after using stratified k fold for optimization:

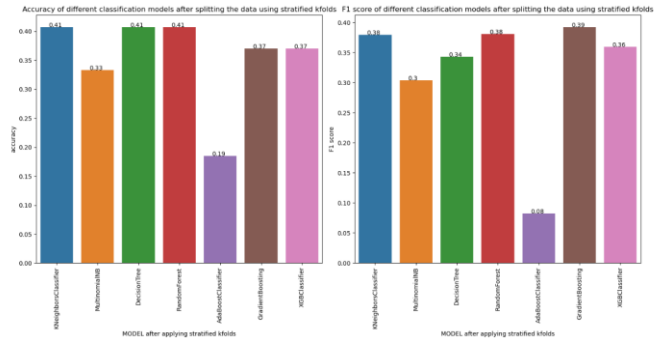

Figure 6: Performance metrics of applied Algorithms using stratified k fold for optimization

After applying the Stratified K-Folds technique to split the data for the 8-class classification models, we observed some interesting trends. While the overall performance of the models remained consistent in terms of accuracy and F1 score, there were variations in their capabilities.Among the models, the Random Forest and K-Nearest Neighbors (KNN) classifiers displayed accuracy levels around 40%, while their F1 scores indicated a relatively lower capacity for precision and recall. Multinomial Naive Bayes exhibited a balanced but slightly lower accuracy and F1 score. On the other hand, AdaBoost performed the least effectively with the lowest accuracy and F1 score among all models. In contrast, Gradient Boosting showed promise with a higher precision and F1 score, suggesting its potential for certain classes within the dataset. Overall, these models have shown varying degrees of performance when considering both accuracy and F1 score.

Statistical test to validate whether there is significant difference in the model performances after using stratified k folds to split the train and test data:
The paired t-tests were conducted to compare the performance metrics (accuracy and F1-score) between models trained using the Stratified K-Folds technique and models trained with the normal data split. Here are the key findings:

Accuracy Comparison: The t-statistic for accuracy is approximately -0.036, with a corresponding p-value of approximately 0.973. The p-value is well above the significance level of 0.05. This indicates that there is no statistically significant difference in accuracy between the two methods of data splitting. In other words, the choice of data splitting technique did not significantly impact model accuracy.

F1-Score Comparison: The t-statistic for F1-score is approximately -0.583, and the p-value is approximately 0.581. Similar to accuracy, the p-value for F1-score is also well above the significance level of 0.05. This suggests that there is no statistically significant difference in F1-score between models trained with Stratified K-Folds and those trained with the normal data split. Thus, the choice of data splitting technique did not significantly affect the F1-score.
The statistical analysis indicates that there is no significant difference in model performance (accuracy and F1-score) when using the Stratified K-Folds technique for data splitting compared to the normal data split. Therefore, either method can be chosen based on other considerations such as data distribution and modelling objectives.

5.2 Performance of Deep Learning Models:

5.2.1 CNN:

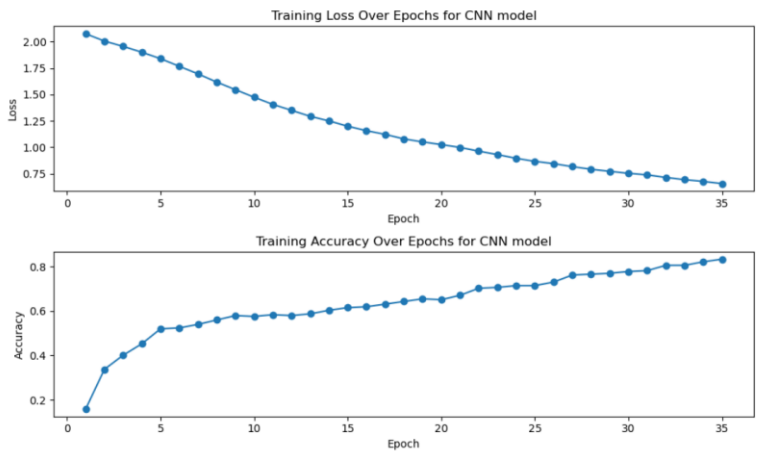

Figure 7: Performance of CNN model

Training Performance: The neural network model was trained over 35 epochs. During this time, the training accuracy steadily increased from approximately 15.87% to about 83.33%. This indicates that the model learned effectively from the training data and improved its ability to classify the training samples.

Validation Performance: The validation accuracy also improved throughout training, reaching approximately 51.85% by the end. This suggests that the model was learning to generalize well to the validation data.

Test Performance: When evaluated on the test dataset, the model achieved an accuracy of about 51.85%. This result is consistent with the validation accuracy and indicates that the model was able to generalize reasonably well to unseen data.

Loss: The loss, which measures the error during training, steadily decreased throughout the training process. This indicates that the model was fitting the training data well and improving its ability to make accurate predictions.
In summary, the neural network model showed good training, validation, and test performance, with the ability to generalize to unseen data. The accuracy on the test dataset reached approximately 51.85%, indicating that the model performed reasonably well in classifying samples into the 8 defined classes.

5.2.2 LSTM:

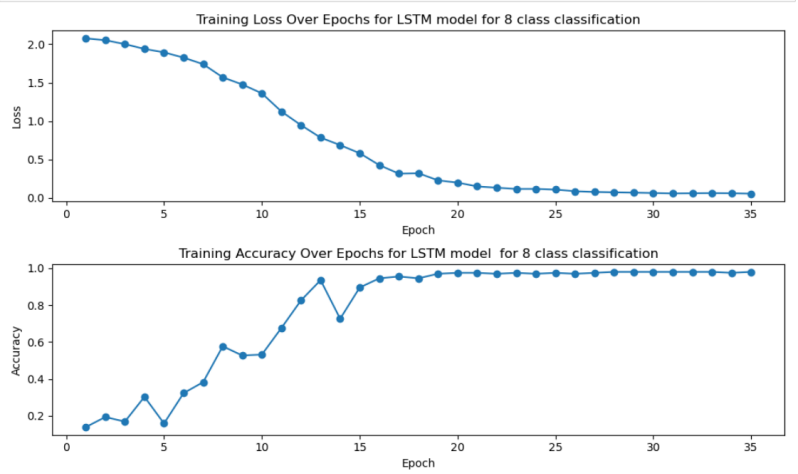

Figure 7: Performance of LSTM model

Training Performance: The training accuracy improved steadily over the epochs, reaching approximately 98.01% accuracy by the final epoch. This indicates that the model effectively learned from the training data.

Validation Performance: The validation accuracy, however, remained relatively low, around 0.00% throughout the training process. This suggests a potential issue with overfitting, where the model performs well on the training data but fails to generalize to new, unseen data.

Test Performance: When evaluated on the test dataset, the model achieved an accuracy of approximately 29.63%. This result is consistent with the low validation accuracy and indicates that the model struggled to generalize to the test data.

Loss: The loss, which measures the error during training, steadily decreased throughout the training process, indicating that the model was fitting the training data well. However, the high loss value on the test data suggests that the model may have overfit the training data.
In summary, the CNN model achieved the best training accuracy.

6. From 8 Classes to 3 Classes- 'Plants, Animals, and Microorganisms'

The intricate genetic overlaps among viral genomes from distinct classes presented a formidable challenge to machine learning and deep learning models, hindering the identification of unique class-specific features. To address this, condensing the classes to three—plants, animals, and microorganisms—was strategically chosen to mitigate the impact of ambiguous distinctions and enhance the models' ability to discern host-specific features. This simplification aligns with a pragmatic approach to molecular understanding, acknowledging the observed complexities in genetic data while optimizing performance metrics. The resultant improvement in model accuracy and interpretability in the three-class problem validates the decision, offering a more practical and robust solution for real-world applications such as disease surveillance and outbreak monitoring.

6.1 Evaluation of machine learning models for 3 Classes

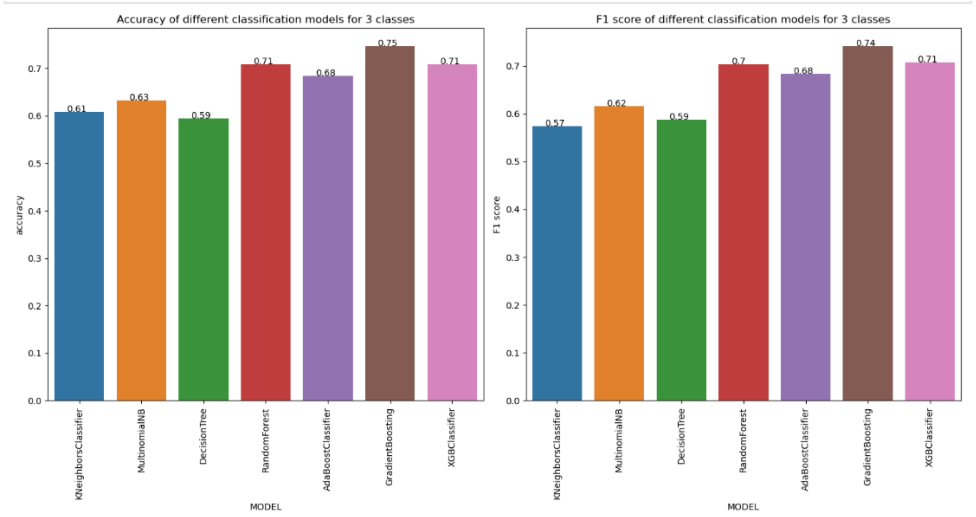

Figure 10: Performance metrics of applied Algorithms for 3 Classes

We can observe from the barplot that most of the ensemble methods are able to classify the hosts effectively. All the ensemble methods show 70% accuracy on average. we can observe that random forest, gradient boosting and xgbclassifier have given the highest accuracy, precision, recall and f1 score.

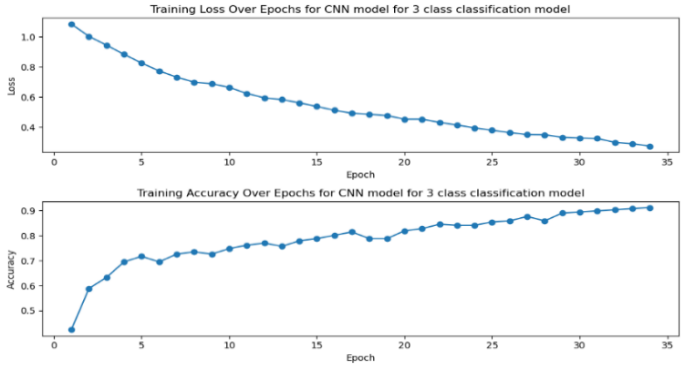

Figure 11: Performance of CNN Model for 3-Classes

Table 1: Evaluating the training and test accuracy of models.

	accuracy_training_data	accuracy_test_data
KNeighborsClassifier	0.800847	0.607595
MultinomialNB	0.716102	0.632911
DecisionTree	1.000000	0.620253
RandomForest	1.000000	0.708861
AdaBoostClassifier	0.830508	0.683544
GradientBoosting	1.000000	0.746835
XGBClassifier	1.000000	0.708861

In general, the models demonstrated reasonable performance in classifying DNA sequences into three categories (plants, animals, microorganisms) before hyperparameter tuning. RandomForest, GradientBoosting, and XGBClassifier initially exhibited the highest accuracy among the models. However, after hyperparameter tuning, the performance of these models further improved.

The hyperparameter tuning process effectively optimized the model parameters, enhancing the classification performance. Among the models, XGBClassifier emerged as the top-performing classifier with the highest accuracy, precision, recall, and F1 score of 0.76,0.76,0.75,0.77 respectively after tuning. This suggests that

XGBoost is a suitable choice for classifying DNA sequences into the specified categories. However, it's essential to note that model selection should consider both performance and computational efficiency, as XGBoost can be computationally intensive.

6.2 Evaluation of Deep learning models for 3 Classes:

CNN: The model learned to classify DNA sequences into host categories effectively. It started with relatively low accuracy but progressively improved as more epochs were completed. The training accuracy reached approximately 89.82%, which means that, on the training data, the model correctly predicted the host category for about 89.82% of the sequences. The validation accuracy reached approximately 75.44%. This indicates that the model generalized well to unseen data, as the validation accuracy is close to the training accuracy.

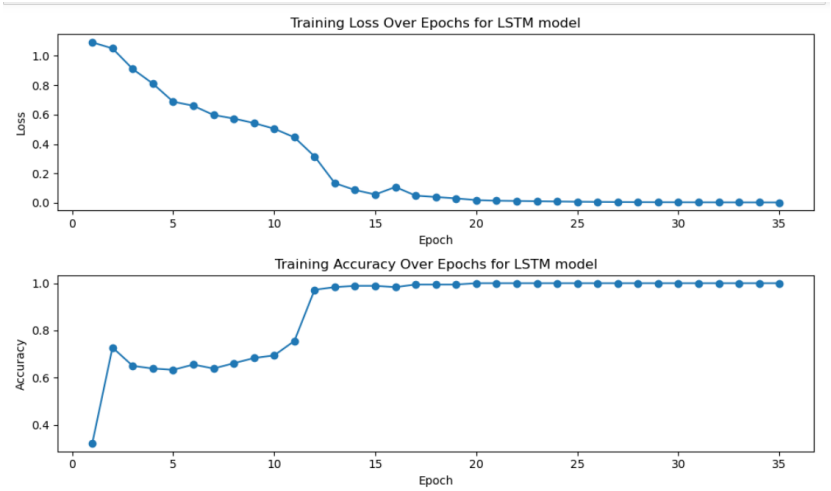

Figure 12: Performance of LSTM Model for 3-Classes

The LSTM model demonstrated strong learning capabilities during training, achieving nearly perfect accuracy on the training data. However, the model's performance on the validation and test datasets suggests that it might be overfitting to some extent. It may have learned noise or specific details of the training data that do not generalize well. The observed overfitting in both the LSTM and CNN models can be attributed to the presence of duplicate data in our genomes, particularly for viruses that infect multiple hosts. These duplicates introduce bias into the training process, as the model may inadvertently memorize repeated examples rather than learning meaningful patterns. As a result, the models perform exceptionally well on the training data but struggle to generalize to unseen data, leading to reduced performance on the validation and test sets.

7. Conclusion:

In the context of our research project, which encompassed the classification of viral genomes into eight distinct classes, including humans, vertebrates, invertebrates, plants, fungi, protists, bacteria, and archaea, the application of various machine learning and deep learning models yielded notably suboptimal performance results. This outcome, despite our exhaustive efforts in parameter optimization, brings to light the potential challenges embedded within the dataset classes. One of the central issues we have encountered is the likelihood of Genetic Overlaps. Viruses, particularly those hosted by closely related organisms or within similar environments, may exhibit genetic overlaps or similarities that are challenging to distinguish solely based on DNA sequences. When viral genomes from different classes share considerable genetic sequences or motifs, it becomes arduous for machine learning models to identify unique, class-specific features that can reliably distinguish between these viruses. The presence of common genetic elements across classes blurs the lines of differentiation. Genetic overlap gives rise to ambiguity in classifying viral genomes. Models may encounter difficulties in accurately assigning a viral genome to the correct host class due to the presence of shared genetic segments. This ambiguity contributes to misclassifications and reduced model performance. Genetic similarity often results in a higher rate of misclassifications, where viruses from one class may be erroneously classified into a different class. These misclassifications are particularly problematic when addressing complex, closely related classes such as vertebrates and invertebrates.

8. Future Directions

The study's findings shed light on the discovery that viral genomes from different classes exhibit significant genetic overlaps, challenging conventional molecular distinctions. Moving forward, potential research directions in viral host determination involve refining classification models through optimization techniques and incorporating multi-omics data for a comprehensive understanding of viral-host interactions. Exploring transfer learning methods, advanced deep learning architectures, and enhancing model interpretability could provide insights into the intricate relationships within viral genetic sequences and host determinants. Integrating epidemiological data and conducting rigorous validation studies are crucial for assessing model generalizability and reliability. Additionally, investigating clinical and therapeutic implications of viral-host interactions may inform the development of targeted interventions and improve public health outcomes. Collaboration across interdisciplinary teams will be pivotal in advancing research in this field.

References

1. Zheng, N., Wang, K., Zhan, W., & Deng, L. (2018). Targeting Virus-host Protein Interactions: Feature Extraction and Machine Learning Approaches. Current Drug Metabolism, 20(3), 177–184. https://doi.org/10.2174/1389200219666180829121038
2. Cho, Sung-Bae & Won, Hong-Hee. (2003). Machine Learning in DNA Microarray Analysis for Cancer Classification.. Proceedings of the First Asia-Pacific bioinformatics Conference. 34. 189-198.
3. Nguyen, N. G., Tran, V. A., Ngo, D. L., Phan, D., Lumbanraja, F. R., Faisal, M. R., Abapihi, B., Kubo, M., & Satou, K. (2016). DNA Sequence Classification by Convolutional Neural Network. Journal of Biomedical Science and Engineering, 09(05), 280–286.https://doi.org/10.4236/jbise.2016.95021
4. Tampuu, A., Bzhalava, Z., Dillner, J., & Vicente, R. (2019). ViraMiner: Deep learning on raw DNA sequences for identifying viral genomes in human samples. PLoS ONE, 14(9), 1–17. https://doi.org/10.1371/journal.pone.0222271
5. Santoso, W., Hulliyah, K., Nurjannah, W., & Setianingrum, A. H. (2022). Systematic Literature Review: Virus Prediction Based on DNA Sequences using Machine Learning and Deep Learning method. 2022 10th International Conference on Cyber and IT Service Management, CITSM 2022, September, 1–7. https://doi.org/10.1109/CITSM56380.2022.9935921
6. Muflikhah, L., Rahman, M. A., & Widodo, A. W. (2022). Profiling DNA sequence of SARS-Cov-2 virus using machine learning algorithm. Bulletin of Electrical Engineering and Informatics, 11(2), 1037–1046. https://doi.org/10.11591/eei.v11i2.3487
7. Chaturvedi, A., Borkar, K., Priyakumar, D., & Vinod, P. K. (2023). PREHOST: Host prediction of coronaviridae family using machine learning. https://doi.org/10.1016/j.heliyon.2023.e13646
8. Kwon, E., Cho, M., Kim, H., & Son, H. S. (2019). A Study on Host Tropism Determinants of Influenza Virus Using Machine Learning. Current Bioinformatics, 15(2), 121–134. https://doi.org/10.2174/1574893614666191104160927
9. Xu, Yanhua, and Dominik Wojtczak. "Dive into Machine Learning Algorithms for Influenza Virus Host Prediction with Hemagglutinin Sequences." BioSystems 220, no. August (2022): 104740. https://doi.org/10.1016/j.biosystems.2022.104740.
10. Salama, Mostafa A., Aboul Ella Hassanien, and Ahmad Mostafa. "The Prediction of Virus Mutation Using Neural Networks and Rough Set Techniques." Eurasip Journal on Bioinformatics and Systems Biology 2016, no. 1 (2016): 1–11. https://doi.org/10.1186/s13637-016-0042-0.
11. Eng, Christine L.P., Joo Chuan Tong, and Tin Wee Tan. "Predicting Zoonotic Risk of Influenza a Viruses from Host Tropism Protein Signature Using Random Forest." International Journal of Molecular Sciences 18, no. 6 (2017). https://doi.org/10.3390/ijms18061135.
12. Ghosh, Dibyendu, Srija Chakraborty, Hariprasad Kodamana, and Supriya Chakraborty. "Application of Machine Learning in Understanding Plant Virus Pathogenesis: Trends and Perspectives on Emergence, Diagnosis, Host-Virus Interplay and Management." Virology Journal 19, no. 1 (2022): 1–11. https://doi.org/10.1186/s12985-022-01767-5.
13. Barman, R. K., Saha, S., & Das, S. (2014). Prediction of interactions between viral and host proteins using supervised machine learning methods. PLoS ONE. https://doi.org/10.1371/journal.pone.0112034
14. Qiang, X., Kou, Z., Fang, G., & Wang, Y. (2018). Scoring amino acid mutations to predict avian-to-human transmission of avian influenza viruses. Molecules. https://doi.org/10.3390/molecules23071584

15. Agor, J. K., & Özaltın, O. Y. (2018). Models for predicting the evolution of influenza to inform vaccine strain selection. In Human Vaccines and Immunotherapeutics. https://doi.org/10.1080/21645515.2017.1423152
16. M. Phute, A. Sahastrabudhe, S. Pimparkhede, S. Potphode, K. Rengade and S. Shilaskar, "A Survey on Machine Learning in Lithography," 2021 International Conference on Artificial Intelligence and Machine Vision (AIMV), Gandhinagar, India, 2021, pp. 1-6, doi: 10.1109/AIMV53313.2021.9670977.