A Deep Learning Approach for Viral DNA Sequence Classification using Genetic Algorithm

Ahmed El-Tohamy, Huda Amin Maghwary, Nagwa Badr
Information Systems Department
Faculty of Computer and Information Sciences, Ain Shams University
Cairo, Egypt

***Abstract*—DNA sequence classification is one of the major challenges in biological data processing. The identification and classification of novel viral genome sequences drastically help in reducing the dangers of a viral outbreak like COVID-19. The more accurate the classification of these viruses, the faster a vaccine can be produced to counter them. Thus, more accurate methods should be utilized to classify the viral DNA. This research proposes a hybrid deep learning model for efficient viral DNA sequence classification. A genetic algorithm (GA) was utilized for weight optimization with Convolutional Neural Networks (CNN) architecture. Furthermore, Long Short-Term Memory (LSTM) as well as Bidirectional CNN-LSTM model architectures are employed. Encoding methods are needed to transform the DNA into numeric format for the proposed model. Three different encoding methods to represent DNA sequences as input to the proposed model were experimented: k-mer, label encoding, and one hot vector encoding. Furthermore, an efficient oversampling method was applied to overcome the imbalanced dataset issues. The performance of the proposed GA optimized CNN hybrid model using label encoding achieved the highest classification accuracy of 94.88% compared with other encoding methods.**

***Keywords*—Deep learning; sequence classification; convolutional neural networks; genetic algorithm; sequence encoding**

I. Introduction

Viruses are the leading cause of infectious diseases and have a harmful impact on the human population. Recent examples of such viruses include COVID-19, SARS, and MERS. As a result of viral outbreaks, new vaccines have been developed for such pathogens [1]. When it comes to virus subtyping and taxonomy classification, the classification of a virus's genomic sequence is extremely vital to analyze and understand for faster production of such vaccines. A virus's genome is either made up of DNA or RNA, and it is referred to as a DNA virus or an RNA virus, accordingly [2]. An organism's genetic code is encoded in the deoxyribonucleic acid (DNA). Adenine (A), thymine (T), cytosine (C), and guanine (G) are the four nucleotides that the DNA consists of. These are referred to as the DNA nucleotide bases [3]. Each type of nucleotide has a binding to its complementary pair on the opposite strand in the double-stranded DNA. Adenine and cytosine form a pair with thymine and guanine, respectively. Single-stranded or double-stranded RNA are both possible for ribonucleic acid. T is replaced by U in RNA. The improvement of phylogenetic and functional research of viruses may be enhanced by the correct classification of genomic sequences [4,5]. Genomic sequences are classified into different groups based on their qualities, traits, or attributes, and this process is known as genomic sequencing classification. The more information is known about the virus, the closer an efficient vaccine can be developed quickly. Because viruses' genomic sequences may have little in common with those of other viruses, it is difficult to classify them. The genomic sequence can be classified using several different approaches. Machine learning models can be trained using well-understood sequences to predict the profile of unknown sequences [6]. As a new branch of machine learning, deep learning has emerged in the last several years. To represent data at increasingly higher levels of abstraction, these models employ multiple non-linear transformations. These models can deal with complex challenges because of their many hidden layers. Many studies have used machine learning and deep learning algorithms to analyze DNA sequences [6,7]. Manual feature extraction is used in these machine learning models [8]. On the other hand, this can lead to various complications, such as selecting features that do not lead to the optimal solution or missing out on key features. Most significantly, the main key features from the DNA dataset extracted are not clear. Besides, it is difficult to extract these features manually using traditional machine learning algorithms. Therefore, an automatic feature selection approach is proposed to overcome this issue. One of the greatest deep-learning methods for extracting important characteristics from a dataset is convolutional neural networks (CNNs) [9,10]. This study proposes an optimized convolutional neural network architecture for DNA sequence classification using genetic algorithm (GA) optimization layer as well as a long short-term memory (LSTM) layer. LSTM is a kind of recurrent neural network (RNN). It can process entire sequences of data effectively [11]. Besides, A genetic algorithm (GA) is proposed to optimize the deep learning model. GA is a heuristic approach inspired by the process of natural selection that is used in computer science and operations research [12]. It is a subclass of evolutionary algorithms (EA) that includes other metaheuristics. Genetic algorithms are commonly employed to generate solutions to optimization and search problems by utilizing bio-inspired operators such as mutation, crossover, and selection [13,14]. A genetic algorithm optimization layer was implemented to improve the accuracy of the classification model. The introduction of evolutionary algorithms such as genetic algorithms showed to optimize deep neural network weight matrix [15]. Thus, optimizing the weight matrix of the convolutional neural network can achieve a better classification accuracy. It can also give better classification results for the LSTM models as the CNN layer output is used as an input to them. As a proof of concept, the optimized model was compared with and without the proposed GA optimization layer. The accuracy of the model with the GA optimization layer is shown to be better than the model without it. Moreover, a comparison was held using the same dataset with previously implemented models. The used dataset contains more viral sequences that may dominate the learning process which lead to a false increase in the overall accuracy. Therefore, an improved oversampling approach was applied to overcome the imbalanced dataset issue. The main contributions of this paper include a proposal of a hybrid deep learning model for efficient viral DNA sequence classification and an introduction of an optimization evolutionary algorithm to the proposed classification framework to improve the overall accuracy. In addition, an efficient oversampling approach is applied for handling the imbalanced dataset as well as increasing the dataset class variability. Besides, one-hot encoding is newly experimented on the viral DNA sequence dataset as an encoding method whereas k-mer encoding [16] and label encoding was used before. The paper is organized as follows: in Section II, the related work is reviewed. Section III describes the dataset and the different preprocessing techniques applied on the dataset. Then, the proposed approach is presented. In Section IV, the experimental results and comparisons with other models are demonstrated. Finally, the paper is concluded in Section V.

II. Related Work

Different studies employed several models and techniques for the classification of viral sequences. In [17], a new approach for classifying the Avian Influenza A viral (AIAV) sequences of the hemagglutinin (HA) and neuraminidase (NA) genes into subtypes using DNA sequence data and physicochemical properties is proposed. Mainly using machine learning techniques, four different classifiers, Naïve Bayes, Support Vector Machine (SVM), K-nearest neighbor (KNN), and Decision Tree were compared. The Decision Tree achieved the best accuracy of 95%.

In [18], the author proposed three models for the classification of different viral DNA sequences using raw DNA sequence data. The three classification models were CNN, long short-term memory (LSTM), and convolutional neural network bidirectional long short-term memory (CNN-Bidirectional LSTM). He used the Synthetic Minority Oversampling Technique (SMOTE) algorithm for data oversampling to overcome imbalanced dataset problem with two encoding methods: label encoding and k-mer encoding. Results showed that k-mer encoding achieved the best results with 93.16% accuracy of the CNN model.

In [19], the author used Random Forest and Artificial Neural Network models with metagenomic sequences that were taxonomically sorted into virus and non-virus categories. The algorithms attained accuracy considerably above the level of chance, with an area under the ROC curve of 0.79. There were two codons (TCG and CGC) that showed the most discriminative features for classification.

In [20], the author utilized combining two classification algorithms with ensemble techniques such as Xgboost and random Forest to improve the accuracy of classifying DNA sequence splice junction types for small example datasets. They achieved an accuracy of 96.24% for Xgboost and 95.11% for Random Forest.

The author in [9] developed a novel method for classifying DNA sequences using a convolutional neural network and treating the sequences as text input. The author employed one-hot vectors to represent sequences as input to the model. The approach was evaluated on 12 DNA sequence datasets. Significant improvements were found in all the previous models using his proposed approach for DNA sequence classification with improved accuracy up to 6.12% on the H3K4me3 dataset.

Most of the existing works tend to focus on training the classification models without any kinds of optimization both on the preprocessing step and prior to the classification step. Therefore, in this research, a hybrid deep learning model with a genetic algorithm optimization layer is proposed. The genetic algorithm layer is applied to optimize the weights of the CNN model. The CNN model is then utilized for classification as a separate model as well as an input to the LSTM and CNN-LSTM Bidirectional models. This will greatly improve the overall accuracy. Thus, the classification method uses the optimized genetic algorithm to generate CNN weights. As a prior step in data preprocessing, Adaptive Synthetic Sampling Approach (ADASYN) is used to handle the imbalanced dataset issues.

III. Materials and Methods

A. Dataset

The DNA dataset was extracted from the National Center for Biotechnology Information (NCBI) (https://www.ncbi.nlm.nih.gov). NCBI contains entire DNA sequences for viruses which is publicly available. The acquired virus DNA sequence datasets are COVID, SARS, MERS, dengue, hepatitis, and influenza. In addition, entire DNA sequences for Zika and EBOLA viruses were collected. A FASTA file for each sequence data was collected and downloaded for complete genetic sequences of each class label with sequence ranges from 8 to 38,012 nucleotides. The collected dataset consists of 86,637 inputs. A distribution of each class label and the count of samples in each label is shown in Table I.

TABLE I. Dataset Class Distribution

Class Label	Number of Samples
COVID	45216
SARS	7311
MERS	6735
Dengue	1994
Hepatitis	8577
Influenza	11862
Zika	1920
EBOLA	3022

As shown in Table I, the minority classes like MERS, SARS, Zika, Ebola, and Dengue have low counts unlike COVID, Hepatitis, and Influenza. To overcome this imbalanced dataset issue, the Adaptive Synthetic Sampling Approach (ADASYN) [21] was applied. ADASYN is used to generate synthetic data for the minority classes to oversample them to match the majority classes. ADASYN is a generalized form of the SMOTE (Synthetic Minority Oversampling Technique) algorithm. SMOTE [22] is an oversampling technique in which synthetic samples are generated for the minority data class. Random oversampling can lead to overfitting, which is why this approach helps alleviate that problem [23]. The main difference between ADASYN and SMOTE is that by using ADASYN the number of synthetic instances generated for samples that are difficult to learn is determined by taking the density distribution into account. As a result, difficult-to-learn samples can be used to adaptively alter decision boundaries. ADASYN works by locating the closest k-nearest neighbors of the minority class using Euclidean distance. Then, it chooses a random neighbor, and a line is constructed between the neighbor and the minority class data point. A synthetic sample is generated between them. Fig. 1 demonstrates how the synthetic data points are generated using ADASYN.

B. Data Preproccessing

The most important aspect of both machine learning and deep learning algorithms is preprocessing of data. It affects the accuracy of the proposed model drastically. DNA sequences, unlike text data, are sequences of consecutive letters without a space between them. No words or phrases can be found in the DNA sequence. As a result, k-mer encoding [16] is used for converting DNA sequences into word sequences. This preserves the nucleotide positions of each word in the sequence. Two vector encoding methods, one hot vector encoding, and label encoding are also used to represent the numerical values of the sequences [24]. One hot vector encoding, and label encoding are used because in contrast to image data, which is represented as a two-dimensional numerical matrix as an input to the CNN, text data is represented as a one-dimensional series of consecutive characters. As a result, it must be converted to numerical values to use as the input for the CNN model. A demonstration of both sequence encodings is shown in Fig. 2.

Thus, encoding is the process of transforming nucleotide categorical data into numerical data. In this research paper, three different types of encoding methods, Label encoding, one hot vector encoding, and k-mer encoding, were experimented with separately to encode the DNA sequence and convert it to the suitable numerical form for deep learning. Label encoding is a popular method for representing categorical data as binary vectors efficiently. For each of the four classes of nucleotides (A, T, G, and C), each one is represented as a number to form an array. A is given the value of 1, C is given the value of 2, G is given the value of 3, and finally, T is given the value of 4. An example sequence of (AACG) will be represented as an array of integers of (1,1,2,3). In decimal-binary vector encoding, one-hot vector encoding for DNA sequences is another way of representing nucleotide sequence data in numerical vector representation. Each nucleotide is represented by a binary vector of length 4. A is represented as (1,0,0,0), C as (0,1,0,0), G as (0,0,1,0) and T as (0,0,0,1). Each nucleotide holds a vector representation of 4x1 dimension. Finally, k-mer encoding transforms the complete DNA sequence into smaller substrings of length k, which represents a word. These words can be used effectively in natural language processing techniques.

C. Classification Methods

Three deep learning models were applied. One model consists of the CNN layer only. The other two models consist of two layers. The first layer of both models is the CNN layer. The CNN layer is used as a feature extraction layer. The output of the CNN layer is given as an input to the second layer. The second layer of the first model is CNN-LSTM. The second layer of the second model is CNN-Bidirectional LSTM. One of the main contributions of this work is applying a standard Genetic Algorithm (GA) to optimize each CNN layer in the models. The GA layer is used to optimize the weights in the CNN layer, which in turn improves the accuracy of the classification models [25,26,27]. Each model is trained and tested using three different encoding methods, label encoding, one-hot vector encoding, and finally using k-mer encoding. A summary of the proposed workflow with the models is shown in Fig. 3.

As demonstrated in Fig. 3, after the data preprocessing the GA layer is utilized to optimize the weights of the CNN layer. Then, the three models are used for the classification process.

This section demonstrates the classification methods in detail. In subsection 1, a detailed demonstration of the proposed genetic algorithm optimization layer will be presented. Following that, in subsections 2 and 3 the used classification models will be explained, respectively.

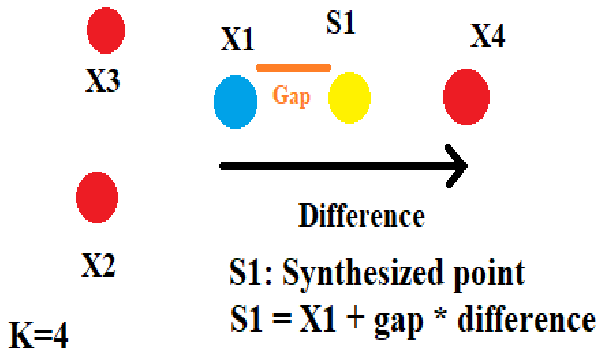

Fig. 1. Generation of Synthetic Data Points using ADASYN with k=4 as an Example and S1 Represents the Synthesized Point of the Minority Class where X_n Represents a Data Point.

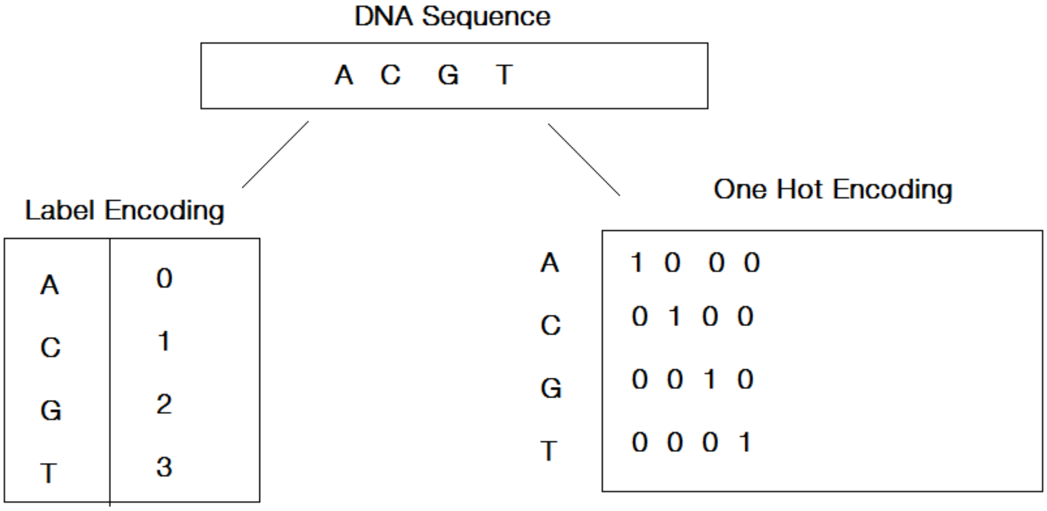

Fig. 2. Difference between One-Hot Encoding and Label Encoding for DNA Sequences.

1) Optimization layer using genetic algorithm (GA): Genetic algorithm [12] relies on biologically inspired operators including mutation, crossover, and selection to produce high-quality solutions to optimization and search problems. GA is mainly a heuristic approach for the optimization of search problems. It is used because the concern is about the optimization of the weights not how much time it takes. Thus, in this research, it is used to optimize the weights of the CNN layer.

The standard GA progression originally proceeds as follows:

- The population's initialization.
- Evaluating each member's fitness.
- Choosing parents to create children for the next generation.
- Parental cross-over to create offspring.
- Randomly mutating the offspring.
- Keep evaluating, reproducing, and mutating until the loss function is optimized.

The following are the proposed steps involved in integrating the GA with the CNN:

- Randomization of initial values of each chromosome.
- Substituting the CNN weights with the values of the selected chromosome.
- Using the newly obtained weights to update the weights of the CNN.
- Calculating the fitness of the present offspring by subtracting the resultant output from the goal output sequence.
- Repeating the simulation for all members of the population.
- Using a roulette strategy for selecting the parents of the next generation.
- Crossover of the parents to produce new offspring.
- Mutating the offspring with a 1% probability of mutation.
- Repeat the previous steps until the evaluation metrics or loss function is optimized. A pseudocode of the proposed GA algorithm is shown in Algorithm 1.

Algorithm 1: Genetic Algorithm for CNN Optimization

```
Input:
Population Number, $n$
Iterations, $I$

Output: Global best configuration of CNN weights $O_{best}$

 Begin
Generation of population $n$
Random initialization of each chromosome in $n$
    Set counter = 0
    Compute the fitness function of each chromosome
    While (counter < $I$)
        Select chromosome pair using roulette
        Calculate the fitness of the current offspring
        Apply crossing over with 70% probability
        Apply mutation with 1% probability
        Replace old population with new population
        Save the current configuration of offspring
        Update $O_{best}$
        Increment counter
    End while
    Return The best solution of configuration $O_{best}$
```

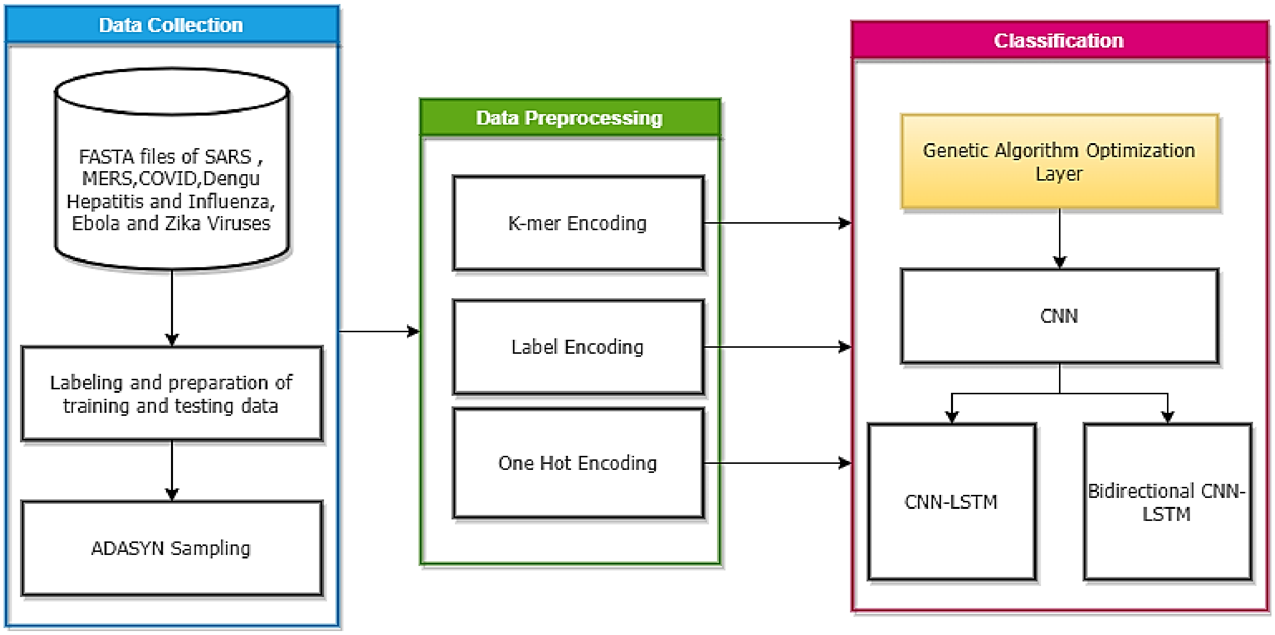
Data Collection
FASTA files of SARS , MERS,COVID,Dengu Hepatitis and Influenza, Ebola and Zika Viruses
Labeling and preparation of training and testing data
ADASYN Sampling
Data Preprocessing
K-mer Encoding
Label Encoding
One Hot Encoding
Classification
Genetic Algorithm Optimization Layer
CNN
CNN-LSTM
Bidirectional CNN-LSTM

Fig. 3. Summary of the Proposed Workflow.

In the proposed GA algorithm, the chromosomes of the original GA reflect the CNN weights in GA. The population, which is made up of several chromosomes, is seeded at random. The number of weight vectors is represented by the number of chromosomes. The fitness function is the training set's accuracy. As a result, while using CNN, the optimization challenge entails maximizing the accuracy of the training set.

The first phase in the algorithm is the generation of the starting population. This is the first stage of the process. In the CNN model, the values of the hyperparameters are picked at random from the defined search spaces with the help of the python random module, which follows the uniform distribution. The fitness evaluation is the next phase. The validation accuracy and the average of the model's five highest training accuracy were both considered in the trials as fitness functions. The highest accuracy represents the highest fitness. The selection method employed is the roulette wheel. Then, the crossover and mutation stages proceeds. After the crossover occurs, the entire new generation gets mutated. Crossover is accomplished by picking hyperparameters between each parent at random in an equiprobable manner. Additionally, the parents are chosen equitably among the surviving. After forming a new generation, the procedure is repeated iteratively from the second step until the final condition is satisfied. The final condition in the context is the occurrence of a specified number of generations. The output of the algorithm is the configuration of the weight with the highest fitness.

In order to keep track of the GA configuration on each generation after evaluating the loss function, the complete parameters of the generation are saved in memory with its corresponding accuracy as well as the selected parents: a Boolean flag which represents if mutation occurs or not, the mutated individual if any and finally the crossing over Boolean flag.

2) Convolutional neural networks (CNN): In deep learning, Convolutional Neural Networks (CNN) [9] is a commonly used technique that may produce cutting-edge results for the majority of classification problems [9, 28, 29]. CNN not only works well in image classification, but it may also deliver accurate results when dealing with text data. CNN is mostly used to automatically extract the features from an input dataset, as opposed to machine learning models, which need the user to select the features from an input dataset. Text classification is processed using 1D CNN. CNN can only deal with numerical data. Therefore, the DNA sequence must be transformed into numerical values via one-hot encoding or label encoding. The CNN architecture extracts features from the input dataset through the use of a series of convolutional layers. After each convolutional layer, there is a maximum pooling layer, and the size of the derived features is lowered. This layer turns the words into a vector space model based on the frequency with which a word appears near other words in the text. For feature extraction, two convolutional layers with filters of 128 and 64 are used in the model, as well as a kernel of size (2 x 2) with ReLU as the activation function for the extraction of features. A max-pooling layer of size (2x2) is added to the feature map to minimize the overall size of the feature map. The softmax function [30] is utilized as the classification layer. In neural network models that predict a multinomial probability distribution, the softmax function is chosen as the activation function in the output layer. It produces an output that shows the probability of each class label. It can provide good results for multi-classification of DNA sequences. The CNN weights are already optimized due to the previous GA layer. Thus, the accuracy of the produced CNN layer is optimal for using it for the next models.

3) CNN-LSTM and CNN-bidirectional LSTM layers: Long Short-Term Memory (LSTM) [11] is an RNN that can learn the long-term dependencies in a sequence. It is used in the prediction and classification of sequences [10,11,26]. It consists of a succession of cells, each of which has three gates: input, output, and forget. In this situation, the LSTM will only retain certain information and discard others. The LSTM output gate uses the sigmoid activation function and the tanh activation function to analyze the cell state to determine what value can be produced. After the convolutional layers, a 100-memory-unit LSTM layer is added to the model to help predict classification labels. The CNN output features are sent into the LSTM layer for classification. Hybrid models using CNN and LSTM are commonly used in NLP tasks to increase classification accuracy [9,10,11,29,31]. Text classification has been improved by using this hybrid model. With dropout layers and regularization approaches, the overfitting problem is minimized in the LSTM modeling process. DNA sequence classification is performed using a bidirectional LSTM/CNN hybrid model. The model employs a CNN for feature extraction and a bidirectional LSTM for classification. Then, CNN is sent into the Bidirectional LSTM as an input. DNA sequence classification makes use of a bidirectional LSTM/CNN hybrid model. For classification, the model relies on a bidirectional LSTM and CNN. The bidirectional LSTM has two RNNs, one for the forward sequence and one for the backward sequence [32].

IV. Experimental Results

The experiments were conducted on a machine using an NVIDIA 1660Ti GPU processor with a RAM size of 16GB. The CPU of the machine was Intel Core i5-8300H @2.30GHz with 4 Cores and 8 logical processors. The models were trained and tested using Tensorflow [33] in python. The dataset was divided into 60% training, 20% validation, and 20% testing using 10-fold cross-validation.

Before the classification phase, the GA was experimented on with different parameters. Several number of generations to end the GA optimization were used. The best results showed that using 12 generations as the specified number of generations yielded the best results. Several mutation probabilities were also used but the one that yielded the best results was a 1% rate of mutation. The rate of crossing-over used was 70%. The categorical cross-entropy function was used in the case of one hot encoding while binary cross-entropy was used with other embeddings as a loss function in the training phase. The error between the actual output and the goal label, on which the weights are trained and updated, is calculated using the loss function of the GA algorithm. A variety of hyperparameters, such as filter size, layer count, and embedding dimension, were used to evaluate the CNN, CNN-LSTM, and CNN-bidirectional LSTM models but the same architecture is used and the same hyperparameters as [18] in testing and evaluation to correctly compare the results. The embedding layer has 8 dimensions, which is the initial layer. If a word appears often next to other words, this layer transforms it into the vector space. This layer, which employs random weights, is responsible for figuring out how each word in the training dataset should be embedded. For feature extraction, a 2x2 kernel with ReLU as an activation function and two convolutional layers with 128 and 64-bit filters are added to the model. Adding a max-pooling layer of size reduces the feature map dimensions (2x2). Using the flatten layer, the feature maps are finally turned into single-column vectors. A thick layer with neurons 128 and 64 receives the output. The number of filters in each layer are 128, 64, and 32, respectively. The embedding dimension of 32 and a k-mer length of 6 are included in the filter's dimensions. The models were trained with 10 epochs each for each of the encoding methods. The resultant training accuracy for each model is shown in Table II.

The same LSTM and LSTM/CNN hybrid models are used in [18] to correctly compare the results and improve upon the currently existing model after adding the GA layer and using ADASYN for oversampling as well as increasing the dataset variability. Increasing the number of class labels in the dataset and the number of input sequences also contributed to the overall better performance of the models. The accuracy increased as compared to [18] by the introduction of the two new class labels for the Zika and the Ebola virus as well as the additional data collected for the rest of the class labels. Label encoding achieved the best accuracy in the CNN classification layer in both training and testing thus it would achieve the best results in the remaining layers. This is because the CNN layer is used as an input to both the CNN-LSTM layer and the CNN Bidirectional LSTM layer. The models were trained and tested using GA optimization and without using GA optimization. Results show that GA optimization yielded noticeably better results in all label, one-hot and k-mer encodings than the results without GA optimization. Testing results and the results of the experiments using GA optimization and the same experiment without using the optimization layer are shown in Table III.

TABLE II. Training Accuracy of the Proposed Method

ENCODING METHOD	**CLASSIFICATION METHOD**		
	CNN	**CNN-LSTM**	**CNN Bidirectional LSTM**
Label Encoding	**95.12%**	**94.36%**	**93.82%**
One-Hot Encoding	94.57%	93.89%	93.22%
K-mer Encoding	94.51%	94.21%	93.55%

TABLE III. Comparison of Classification Models with and without GA Optimization Layer using Different Encoding Methods

ENCODING METHOD	CLASSIFICATION METHOD					
	CNN		CNN-LSTM		CNN Bidirectional LSTM	
	Using GA Optimization	Without Using GA Optimization	Using GA Optimization	Without Using GA Optimization	Using GA Optimization	Without Using GA Optimization
Label Encoding	**93.51%**	92.92%	**93.27%**	92.78%	**93.20%**	92.14%
One-Hot Encoding	**93.77%**	93.16%	**93.54%**	93.02%	**93.44%**	93.13%
K-mer Encoding	**93.51%**	92.92%	**93.27%**	92.78%	**93.20%**	92.14%

With the addition of the GA optimization layer, label encoding, one hot encoding and k-mer encoding achieved an accuracy of 94.88%, 94.33% and 94.05%, respectively using the CNN model. Using CNN-LSTM, label encoding, one hot encoding and k-mer encoding achieved an accuracy of 94.42%, 93.51% and 93.9%, respectively. Finally utilizing the CNN-LSTM Bidirectional model, the accuracies were 93.74% for label encoding, 93.01% for one-hot encoding and 93.37% for k-mer encoding. On the other hand, without using the GA optimization layer the accuracy for each model was considerably less. CNN achieved an accuracy of 93.22%, 93.50% and 93.54% for label encoding, one hot encoding and k-mer encoding, respectively. Using CNN-LSTM model, label encoding achieved an accuracy of 93.5%, one-hot encoding achieved an accuracy of 91.59% and k-mer encoding showed an accuracy of 92.16%. Finally, CNN Bidirectional model achieved an accuracy of 91.35%, 92.16% and 92.46% for label encoding, one hot encoding and k-mer encoding, respectively. Among all the three encoding techniques, label encoding is shown to achieve the best results overall with the introduction of the GA layer and without using it.

In order to compare the results with [18], the two additional class labels Zika and EBOLA viruses were removed from the dataset and then the dataset was experimented on. Thus, the experiment was carried on using ADASYN for oversampling and the addition of the GA optimization layer in comparison with [18] who used SMOTE and the hybrid model without the addition of the GA layer. The resultant accuracies are shown in Table IV. Furthermore, only label encoding and k-mer encoding is demonstrated for comparison as in [18].

By comparing the results of k-mer encoding using GA and introducing two new class labels to the dataset and ADASYN oversampling method, the proposed method is proved to give better accuracy results than the previous model used by [18]. The best results from [18] were achieved using k-mer encoding. In the proposed method in this study the resulting accuracy using k-mer encoding were 94.05% using CNN, 93.9% using CNN-LSTM and 93.37% using CNN Bidirectional LSTM. Whereas it previously resulted in 93.16% using CNN, 93.02% using CNN-LSTM and 93.13% using CNN-Bidirectional LSTM without GA optimization and using SMOTE oversampling with less dataset sequences and less class labels. Thus, the proposed method achieved best accuracy using k-mer encoding in comparison to [18]. It also achieved the best overall classification accuracy of 94.88% using label encoding. The training and validation loss curves for the three encoding methods are shown in Fig. 4.

The accuracy curve shows that label encoding achieved the best training and testing results overall among all the three used encoding methods. One hot encoding showed similar results for both training and testing in CNN Bidirectional LSTM but better training accuracy using CNN and LSTM. Utilizing ADASYN resulted in better results in the overall training accuracy due to the optimized oversampling of the dataset in the minority class labels such as Zika and Dengue. As a limitation, improving the accuracy by introducing the optimization layer leads to an increase in computational time. Moreover, the generated synthetic dataset in the oversampling method might have some fuzzy class boundaries.

TABLE IV. Comparison of Classification Models with Gunasekaran, et al. [18] using GA Optimization and without the Addition of the 2 New Class Labels

ENCODING METHOD	CLASSIFICATION METHOD					
	CNN		CNN-LSTM		CNN Bidirectional LSTM	
	Proposed Model	Gunasekaran, et al. [18]	Proposed Model	Gunasekaran, et al. [18]	Proposed Model	Gunasekaran, et al. [18]
Label Encoding	**93.51%**	92.92%	**93.27%**	92.78%	**93.20%**	92.14%
K-mer Encoding	**93.77%**	93.16%	**93.54%**	93.02%	**93.44%**	93.13%

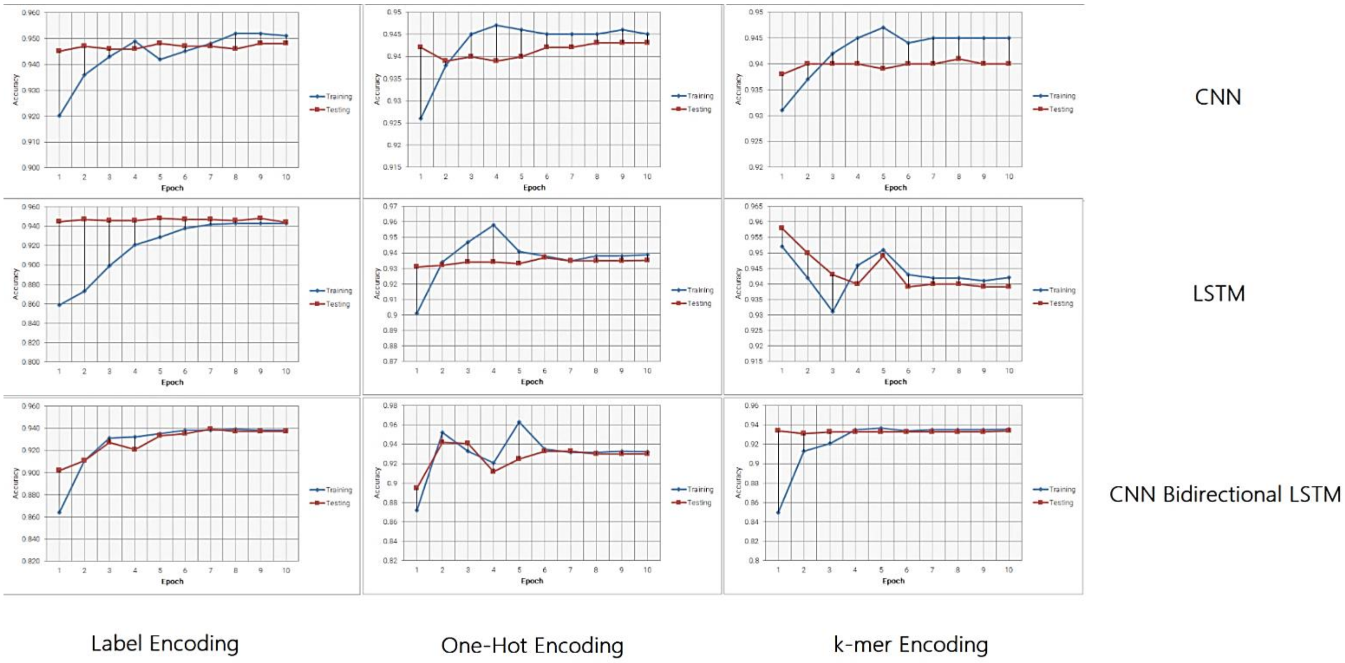

Fig. 4. The Resultant Training and Validation Loss Curves using GA Optimization and without using GA Optimization with all 3 Different Encoding Methods.

V. Conclusion

The classification of viral DNA poses a major challenge in recent years. The accurate classification of the DNA of pandemic viruses will greatly help in the production of vaccines and the identification of new pathogens. This study proposes an optimized method for the accurate classification of viral DNA utilizing genetic algorithm for optimization classification using a hybrid deep learning model. The proposed method uses a genetic algorithm to optimize the weights of the CNN model which enhances the overall classification accuracy. The study also utilizes ADASYN as an optimized dataset oversampling technique for the minority class labels. Three encoding techniques were experimented with which are label encoding, k-mer encoding, and one-hot encoding which was not used in previously proposed models. The experiments showed that the proposed optimization layer GA and ADASYN with the deep learning model outperformed previously proposed models on the same dataset in terms of classification accuracy. The models were then trained and tested with GA optimization and without GA optimization. The GA optimization drastically affected the accuracy of the models. As a result, label encoding was shown to achieve the best accuracy of 94.88% using CNN. Besides, k-mer encoding achieved an accuracy of 94.05% whereas the best results achieved by a previously proposed model were 93.16%. As a result, it is shown that the introduction of an optimization layer improved the overall classification accuracy. The introduction of more evolutionary or optimization algorithms in future research could improve the accuracy further. Furthermore, the use of an optimized oversampling technique yielded better overall accuracy. Therefore, by using ADASYN which is an optimized version of SMOTE yielded better results.

For future work, it is planned to introduce more viral DNA sequences in the training dataset and use other selection criteria for the GA selection algorithm which could further improve the accuracy of the classification. In addition, more optimization methods could be utilized.

References

[1] Trovato, M., Sartorius, R., D’Apice, L., Manco, R. and De Berardinis, P., 2020. Viral emerging diseases: challenges in developing vaccination strategies. Frontiers in Immunology, 11, p.2130.

[2] Gelderblom, H.R., 1996. Structure and classification of viruses. Medical Microbiology. 4th edition.

[3] Travers, A. and Muskhelishvili, G., 2015. DNA structure and function. The FEBS journal, 282(12), pp.2279-2295.

[4] Randhawa, G.S., Soltysiak, M.P., El Roz, H., de Souza, C.P., Hill, K.A. and Kari, L., 2020. Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: COVID-19 case study. Plos one, 15(4), p.e0232391.

[5] Liu, R., Qiao, M., Zheng, J. and Zhou, W., 2021. Analysis SARS-CoV-2 Genomes of G20 Areas on Phylogeny Tree, t-SNE based on Machine Learning.

[6] Yang, A., Zhang, W., Wang, J., Yang, K., Han, Y. and Zhang, L., 2020. Review on the application of machine learning algorithms in the sequence data mining of DNA. Frontiers in Bioengineering and Biotechnology, 8, p.1032.

[7] Abd–Alhalem, S.M., El-Rabaie, E.S.M., Soliman, N., Abdulrahman, S.E.S., Ismail, N.A., El-samie, A. and Fathi, E., 2021. DNA Sequences Classification with Deep Learning: A Survey. Menoufia Journal of Electronic Engineering Research, 30(1), pp.41-51.

[8] Deorowicz, S., 2020. FQSqueezer: k-mer-based compression of sequencing data. Scientific reports, 10(1), pp.1-9.

[9] Nguyen, N.G., Tran, V.A., Phan, D., Lumbanraja, F.R., Faisal, M.R., Abapihi, B., Kubo, M. and Satou, K., 2016. DNA sequence classification by convolutional neural network. Journal Biomedical Science and Engineering, 9(5), pp.280-286.

[10] Sharma, A., Lysenko, A., Boroevich, K.A., Vans, E. and Tsunoda, T., 2021. DeepFeature: feature selection in nonimage data using convolutional neural network. Briefings in bioinformatics, 22(6), p.bbab297.

[11] Nowak, J., Taspinar, A. and Scherer, R., 2017, June. LSTM recurrent neural networks for short text and sentiment classification. In

International Conference on Artificial Intelligence and Soft Computing (pp. 553-562). Springer, Cham.

[12] Katoch, S., Chauhan, S.S. and Kumar, V., 2021. A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications, 80(5), pp.8091-8126.

[13] Abd-El-Wahed, W.F., Mousa, A.A. and El-Shorbagy, M.A., 2011. Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems. Journal of Computational and Applied Mathematics, 235(5), pp.1446-1453.

[14] Hamdia, K.M., Zhuang, X. and Rabczuk, T., 2021. An efficient optimization approach for designing machine learning models based on genetic algorithm. Neural Computing and Applications, 33(6), pp.1923-1933.

[15] Idrissi, M.A.J., Ramchoun, H., Ghanou, Y. and Ettaouil, M., 2016, May. Genetic algorithm for neural network architecture optimization. In 2016 3rd International conference on logistics operations management (GOL) (pp. 1-4). IEEE.

[16] Asim, M.N., Malik, M.I., Dengel, A. and Ahmed, S., 2020, July. K-mer Neural Embedding Performance Analysis Using Amino Acid Codons. In 2020 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[17] Humayun, F., Khan, F., Fawad, N., Shamas, S., Fazal, S., Khan, A., Ali, A., Farhan, A. and Wei, D.Q., 2021. Computational method for classification of avian influenza A virus using DNA sequence information and physicochemical properties. Frontiers in Genetics, 12, p.599321.

[18] Gunasekaran, H., Ramalakshmi, K., Rex Macedo Arokiaraj, A., Deepa Kanmani, S., Venkatesan, C. and Suresh Gnana Dhas, C., 2021. Analysis of DNA sequence classification using CNN and hybrid models. Computational and Mathematical Methods in Medicine, 2021.

[19] Bzhalava, Z., Tampuu, A., Bała, P., Vicente, R. and Dillner, J., 2018. Machine Learning for detection of viral sequences in human metagenomic datasets. BMC bioinformatics, 19(1), pp.1-11.

[20] Syahrani, I.M., 2019. Comparation analysis of ensemble technique with boosting (Xgboost) and bagging (Randomforest) for classify splice junction DNA sequence category. Jurnal Penelitian Pos dan Informatika, 9(1), pp.27-36.

[21] He, H., Bai, Y., Garcia, E.A. and Li, S., 2008, June. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence) (pp. 1322-1328). IEEE.

[22] Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, pp.321-357.

[23] Marques, Y.B., de Paiva Oliveira, A., Ribeiro Vasconcelos, A.T. and Cerqueira, F.R., 2016. Mirnacle: machine learning with SMOTE and random forest for improving selectivity in pre-miRNA ab initio prediction. BMC bioinformatics, 17(18), pp.53-63.

[24] Miao, Y., Liu, F., Hou, T. and Liu, Y., 2022. Virtifier: a deep learning-based identifier for viral sequences from metagenomes. Bioinformatics, 38(5), pp.1216-1222.

[25] Loussaief, S. and Abdelkrim, A., 2018. Convolutional neural network hyper-parameters optimization based on genetic algorithms. International Journal of Advanced Computer Science and Applications, 9(10).

[26] Chen, M., Yu, L., Zhi, C., Sun, R., Zhu, S., Gao, Z., Ke, Z., Zhu, M. and Zhang, Y., 2022. Improved faster R-CNN for fabric defect detection based on Gabor filter with Genetic Algorithm optimization. Computers in Industry, 134, p.103551.

[27] Bhandari, A., Tripathy, B.K., Jawad, K., Bhatia, S., Rahmani, M.K.I. and Mashat, A., 2022. Cancer Detection and Prediction Using Genetic Algorithms. Computational Intelligence and Neuroscience, 2022.

[28] Aoki, G. and Sakakibara, Y., 2018. Convolutional neural networks for classification of alignments of non-coding RNA sequences. Bioinformatics, 34(13), pp.i237-i244.

[29] Min, S., Lee, B. and Yoon, S., 2017. Deep learning in bioinformatics. Briefings in bioinformatics, 18(5), pp.851-869.

[30] Bridle, J., 1989. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. Advances in neural information processing systems, 2.

[31] Lugo, L. and Hernández, E.B., 2021. A Recurrent Neural Network approach for whole genome bacteria identification. Applied Artificial Intelligence, 35(9), pp.642-656.

[32] Mughees, N., Mohsin, S.A., Mughees, A. and Mughees, A., 2021. Deep sequence to sequence Bi-LSTM neural networks for day-ahead peak load forecasting. Expert Systems with Applications, 175, p.114844.

[33] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. and Kudlur, M., 2016. {TensorFlow}: a system for {Large-Scale} machine learning. In 12th USENIX symposium on operating systems design and implementation (OSDI 16) (pp. 265-283).