IJICIS, Vol.24, No.1, 89-104
DOI: 10.21608/ijicis.2024.279692.1332

International Journal of Intelligent Computing and Information Sciences

https://ijicis.journals.ekb.eg/

INTEGRATION OF DEEP LEARNING MODELS FOR ENHANCED CLASSIFICATION OF VIRAL DNA SEQUENCES ACROSS SPECIFIC VIRUSES AND VIRAL FAMILIES

Ahmed El-Tohamy*
Information Systems Department,
Faculty of Computer and Information
Sciences, Ain Shams University,
Cairo, Egypt
ahmed.eltohamy@cis.asu.edu.eg

Huda Amin Maghawry
Information Systems Department,
Faculty of Computer and Information
Sciences, Ain Shams University,
Cairo, Egypt
huda_amin@cis.asu.edu.eg

Nagwa Badr
Information Systems Department,
Faculty of Computer and Information
Sciences, Ain Shams University,
Cairo, Egypt
nagwabadr @cis.asu.edu.eg

Received 2024-03-27; Revised 2024-03-27; Accepted 2024-03-30

Abstract: *The field of genomic bioinformatics is continually challenged by the need for precise classification of viral DNA sequences. The challenge of accurately classifying viral sequences is crucial for the development of diagnostic and therapeutic strategies for any viral outbreaks. This study presents a comprehensive approach integrating two distinct deep learning models, namely the Genetic Algorithm (GA) optimized Convolutional Neural Networks (CNN) hybrid model and the CNN-Extreme Learning Machines (ELM) model aiming to enhance the classification of viral DNA sequences across specific viruses and viral families.*

A comprehensive data preprocessing strategy is employed, wherein both datasets undergo k-mer, label, and one-hot vector encoding. This allows for a uniform and comparative analysis across different models and datasets. When the optimized GA-CNN is applied to the more generic viral family dataset, it demonstrates a good adaptability with an accuracy of 95.88% achieving a higher result than the CNN-ELM. In contrast, the CNN-ELM, when tested on the specific virus dataset, maintains robust feature extraction capabilities, faster training time but lower than the optimized GA-CNN model achieving an accuracy of 92.7%.

A comparative analysis of training times is also employed in this study. The CNN-ELM model shows a notable efficiency, with a 34% faster training time compared to the GA-CNN. Moreover, when both models are applied to the new generic dataset, a comparative study with other deep learning models is conducted. Remarkably, the GA-CNN outperforms other models, achieving the highest classification accuracy of 95.88%.

Keywords: *Genomic Bioinformatics, Viral DNA Classification, GA-CNN, CNN-ELM, Deep Learning, Data Preprocessing, Encoding Methods, Training Efficiency, Comparative Analysis.*

***Corresponding Author**: Ahmed El-Tohamy

Information Systems Department, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt

Email address: ahmed.eltohamy@cis.asu.edu.eg

1. Introduction

Viruses are microscopic agents capable of infecting a wide range of organisms. They have always been the subject of extensive research due to their impact on health and ecology. These entities are characterized by their simple structure and complex behavior. They are also known for their ability to replicate only within living cells [1,2]. The challenge they pose to public health, particularly evident in the wake of recent global pandemics, underscores the urgent need for advanced research in virology. Such a challenge was very prominent during the recent COVID-19 pandemic at which the world struggled greatly. Genomic sequencing has emerged as a powerful tool in understanding these pathogens [3]. This enables scientists to delve deeper into the genetic structure and function of these viruses.

The rise of high-throughput sequencing technologies has revolutionized the study of viral genomes [3]. These advancements have led to the accumulation of vast amounts of viral genetic data, stored in public databases such as the National Center for Biotechnology Information (NCBI) [4]. Access to such comprehensive genomic information has opened new avenues for research, particularly in the fields of diagnostics, vaccine development, and antiviral therapies. However, the sheer volume and complexity of this data presents significant analytical challenges, necessitating the development of sophisticated computational tools.

In response to these challenges, machine learning and deep learning approaches have gained prominence [5,6,7]. These methodologies are known for their ability to handle large datasets and extract meaningful patterns. They are particularly well-suited for genomic analysis. Deep learning, a subset of machine learning inspired by the structure and function of the brain, has further enhanced these capabilities.

The transformative impact of machine learning and deep learning techniques in the domain of genomic bioinformatics has been nothing short of revolutionary, particularly in virus classification. These advanced computational methods have shown an uncanny ability to unravel the intricacies of viral genetic sequences. By employing deep learning models, researchers have been able to identify nuanced patterns embedded within the DNA or RNA of viruses which was very challenging before [2],[8]. This capability not only facilitates precise classification of various viral strains but also enriches our understanding of their evolutionary trajectories and the complex web of diversity they exhibit.

The objective of this paper is to integrate two distinct models to analyze and enhance the classification of viral DNA sequences across both specific viruses and viral families. The first model is the Genetic Algorithm Optimized Convolutional Neural Network (GA-CNN) model which was used to classify viral DNA before. The success of the GA-CNN model can be attributed to its ability to optimize the neural network's weights effectively using genetic algorithms [10]. The GA-CNN model's design allows it to adapt to the unique challenges posed by viral DNA and RNA sequences. Viral genomes are often characterized by high mutation rates and genetic variability, presenting a moving target for any classification algorithm. The GA-CNN model, with its optimized CNN architecture, has demonstrated a remarkable ability to navigate these challenges. It effectively captures the subtle genetic variations and patterns that are key to distinguishing between different viruses.

The other model is the Convolutional Neural Network with Extreme Learning Machines (CNN-ELM) [11]. It harnesses the power of Convolutional Neural Networks (CNNs) for extracting key features from complex viral sequences. CNNs, known for their proficiency in pattern recognition [12,13]. In tandem with the CNN, the model employs Extreme Learning Machines (ELMs) for the classification phase. ELMs are renowned for their rapid learning capabilities, setting them apart from other machine learning algorithms [14,15]. This speed is particularly advantageous in the context of viral classification, where the ability to quickly process and classify large volumes of genetic data can be extremely valuable. This comprehensive approach aims to harness the strengths of each model, applying them to a wider range of datasets and encoding methodologies. Therefore, this study explores the adaptability and efficiency of these models in a broader genomic context. This study is structured in the following manner: Section 2 provides an overview of the relevant literature in the topic. The datasets utilized, along with the various preprocessing methods employed, are detailed in Section 3. Subsequently, Section 4 outlines the complete methodology used in this research. The findings from the experiments are presented in Section 4. The final section, Section 5, concludes the paper with a summary and takeaway and future experiments and work.

2. Related Works

Numerous studies have employed diverse models and techniques to effectively classify viral sequences. These approaches utilize a wide range of methodologies aimed at accurately classifying and distinguishing different viral genetic information.

Ming et al. [16] present HostNet, a deep learning framework that enhances virus-host prediction accuracy. HostNet utilizes a Transformer-CNN-BiGRU architecture and incorporates two modules for improved sequence representation. Experimental results demonstrate that HostNet outperforms existing deep learning methods in terms of host-prediction accuracy and F1 score. The framework shows promise in addressing challenges related to sparse and varying-length virus sequence data, making it a valuable tool for virus-host prediction in various biological contexts.

A study by Humayun et al. [17] introduced a novel method to classify Avian Influenza A viral (AIAV) sequences into subtypes based on the hemagglutinin (HA) and neuraminidase (NA) genes. By analyzing DNA sequence data and considering physicochemical properties, this approach employed machine learning techniques, comparing four classifiers: Naïve Bayes, Support Vector Machine (SVM), K-nearest neighbor (KNN), and Decision Tree. Among these classifiers, the Decision Tree model exhibited the highest accuracy, reaching 95%.

Another recent study by Alakus et al. [18] utilized DNA sequences of the viruses causing monkeypox and warts and were analyzed using a deep learning algorithm. The study consisted of four stages: obtaining the DNA sequences, mapping the sequences using various methods, classifying the mapped sequences with a deep learning algorithm, and comparing the performance of the DNA-mapping methods. The results showed an average accuracy of 96.08% and an F1-score of 99.83%, demonstrating the effectiveness of the deep learning model in accurately classifying the DNA sequences and distinguishing between monkeypox and warts based on their genetic information.

Tampuu et al. introduced ViraMiner [19], an innovative deep learning-based approach that utilizes Convolutional Neural Networks (CNNs) to detect patterns and pattern frequencies in unprocessed metagenomics contigs. By applying this dual approach, the researchers successfully tested their method on a vast dataset of human samples, specifically targeting viral sequences. The results demonstrated an impressive area under the ROC curve of 0.923, indicating the ability of ViraMiner to accurately identify viral sequences within raw metagenomic contigs derived from various human samples.

Wang et al. [20] presented an innovative approach that merges CNNs with ELMs to enhance feature extraction and achieve efficient classification of Synthetic Aperture Radar (SAR) images. The authors improved the conventional CNN model by replacing the Sigmoid activation function with the more effective Rectified Linear Unit (ReLU) activation function at which they employed ELM as a classifier. Through testing their method on the MSTAR database it achieved remarkable results, attaining 100% accuracy while maintaining fast execution time due to the ELM layer used.

Furthermore, in a previous study, the Genetic Algorithm (GA) optimized Convolutional Neural Network (GA-CNN) model was introduced [9]. This model represents a synergy of two powerful computational concepts: the robust feature extraction inherent in Convolutional Neural Networks (CNNs) and the efficiency of optimization provided by Genetic Algorithms (GAs). The GA-CNN model, as detailed in previous research, has successfully harnessed these capabilities to analyze and classify specific viral sequences with remarkable accuracy [9].

El-Tohamy et al. [9] developed an optimized Convolutional Neural Network (GA-CNN) for classification of viral genomes. It was also enhanced to outperform previous work of Gunasekaran, Hemalatha, et al. [21] by introducing more virus labels. It used the GA to optimize the CNN weights and the optimized ADASYN for preprocessing. This approach combined the powerful feature extraction capabilities of CNNs with the optimization strength of GAs. The GA-CNN model was adeptly applied to a dataset of specific viral sequences using 3 encoding methods label , one-hot and k-mer encoding, achieving the highest classification accuracy of 94.88% using label encoding . El-Tohamy et al. [11] also introduced Convolutional Neural Network with Extreme Learning Machines (CNN-ELM) model with the generic diverse virus family dataset collected. This model achieved a notable accuracy of 94.54% using k-mer encoding for classifying a broad spectrum of viral families. The CNN-ELM model uniquely integrated CNNs for in-depth feature extraction from viral sequences with the rapid and efficient classification capabilities of ELMs. This architecture proved particularly effective in handling the diversity present in viral genomes with a confusion matrix proving powerful performance across all the 10 major viral families in the dataset. The objective of this study is to integrate the GA-CNN and CNN-ELM models presented by El-Tohamy et al. in [9] and [11]. The methodology involves a comprehensive comparative application and evaluation of different encoding techniques: k-mer, label, and one-hot vector to explore the potential results and integrate the models with both specific virus and diverse viral family datasets.

3. Materials & Methods

3.1 Dataset Collection:

The current study leverages two distinct datasets. The Specific Viral DNA Dataset [9] and the viral family dataset [11]. These datasets are crucial for evaluating the adaptability and effectiveness of the models across various viral genetic contexts. The first dataset was obtained from the National Center for Biotechnology Information (NCBI). It includes complete DNA sequences of specific viruses, namely COVID, SARS, MERS, dengue, hepatitis, influenza, Zika, and EBOLA. Each sequence in this dataset is provided in a FASTA file format. The sequence lengths vary significantly, ranging from as few as 8 nucleotides to as many as 38,012 nucleotides. The total number of sequences amassed in this dataset is 86,637. The dataset exhibits an imbalance in class distribution, with certain viruses like COVID having a significantly higher number of samples compared to others like Zika and EBOLA.

This imbalance is addressed by employing the Adaptive Synthetic Sampling Approach (ADASYN) [22] to generate synthetic data for under-represented classes, enhancing the diversity of the dataset.

Table 1 provides a detailed distribution of each class label and the count of samples in each label.

Table 1: Specific Virus dataset description

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

The second dataset is a robust viral family dataset collected to be a generic dataset for the classification of a viral family instead of specific viruses. The dataset was collected from the NCBI public database. This dataset is designed to capture the genetic diversity and evolution of various virus families. It includes DNA sequences from ten major viral families: Coronaviridae, Flaviviridae, Togaviridae, Paramyxoviridae, Bunyaviridae, Rhabdoviridae, Filoviridae, Herpesviridae, Adenoviridae, and Reoviridae. These are the major 10 viral families each with entire genomes for the major viruses in each family. The dataset structure consists of sequences in FASTA format. This dataset provides a comprehensive collection of 50,000 viral genome sequences, equally distributed across the ten virus families. For each family, four representative viruses with the highest counts were selected to ensure a wide coverage of genetic diversity for each of the viral families. Each virus family in the dataset is represented by 5,000 sequences, ensuring an equal distribution and a representative sampling of the viral landscape. Table 2 summarizes the virus families and their representative viruses. Table 3 presents the average length of genome sequences for each virus family in the dataset, highlighting the range of genome sizes.

Table 2: Viral family dataset and their representative viruses

Virus Family	Representative Viruses
Coronaviridae	SARS-CoV-2, SARS-CoV, HCoV-OC43, and HCoV-229E
Flaviviridae	Dengue virus, West Nile virus, Zika virus, Yellow fever virus
Togaviridae	Chikungunya virus, Eastern equine encephalitis virus, Sindbis virus, Semliki forest virus
Paramyxoviridae	Measles virus, Mumps virus, Sendai virus, Newcastle disease virus
Bunyaviridae	La Crosse virus, Hantaan virus, Rift Valley fever virus, and Crimean-Congo hemorrhagic fever virus.
Rhabdoviridae	Rabies virus, Vesicular stomatitis Indiana virus, Piry virus, Hirame rhabdovirus
Filoviridae	Ebola virus, Marburg virus, Reston virus, Bundibugyo virus
Herpesviridae	Human herpesvirus 1, Human herpesvirus 2, Varicella-zoster virus, Epstein-Barr virus
Adenoviridae	Human adenovirus C, Human adenovirus D, Bovine adenovirus A, Canine adenovirus 1
Reoviridae	Rotavirus A, Bluetongue virus, Mammalian orthoreovirus type 3, Avian orthoreovirus

Table 3: Viral Family dataset sequence description

Feature	Description
File Format	FASTA
Sequence Type	DNA
Number of Entries	50,000 (10 families x 5,000 sequences)
Number of Families	10

Feature	Description
Sequences per Family	5,000
Header Line	Starts with '>' followed by a unique identifier and description
Sequence Data	Single-letter codes for nucleotides (A, C, G, T)
Line Length	60-80 characters per line

3.2 Data Preprocessing:

Data preprocessing is a crucial component in both machine learning and deep learning algorithms, significantly impacting the accuracy of the models used. DNA sequences are composed of continuous letter strings, differing from typical text data which is separated by spaces and includes distinct words or phrases [23]. Consequently, k-mer encoding [24] is employed to transform these DNA sequences into sequences of 'words', effectively maintaining the position of each nucleotide within the sequence. Additionally, two methods of vector encoding; one-hot vector encoding and label encoding; are applied to convert these sequences into numerical forms [24]. Unlike image data, which is typically processed in a two-dimensional numerical matrix format for CNN input, textual data like DNA sequences is characterized by a one-dimensional chain of characters. Therefore, this data requires conversion into numerical values for appropriate input into the deep learning models. Both datasets will undergo k-mer encoding. For each sequence, the DNA string is broken down into subsequences (k-mers) of length *k*. In line with the approach used by [11] which gave the highest accuracy among tried k values, a length of k=4 will be utilized. This process results in transforming each sequence into a collection of overlapping k-mers, which are then used as input features for the models. Additionally, label encoding will be applied to the datasets. In this method, each nucleotide (A, C, G, T) is assigned a unique numerical value, converting the sequences into arrays of integers. This encoding is less complex than k-mer encoding but still retains the sequential nature of the DNA, which is crucial for models like CNN-ELM that rely on pattern recognition in sequences. The third encoding method to be employed is one-hot vector encoding. Each nucleotide is represented by a binary vector, ensuring a distinct representation.
For example, Adenine (A) could be encoded as [1,0,0,0], C as [0,1,0,0], and so on.
One-hot encoding and label-encoding was not experimented with before in [11] which is important to test their resulting performance as well and to be used for comparative analysis.
As a result, the preprocessing steps for both the datasets utilized in this study are as follows:

A) Data Cleaning:

Both datasets undergo an initial cleaning process by removing any non-DNA characters, gaps, or ambiguous bases, leaving only the four nucleotide bases (A, C, G, T).

B) Encoding Methods:

A key preprocessing step involves k-mer encoding of the sequences, with a length (k) set to 4. This process involves breaking down each DNA sequence into overlapping subsequences (k-mers) using a sliding window approach [25]. The sliding window, of size k=4, moves along the sequence, capturing each k-mer at every position, thus preserving local sequence patterns. The resulting k-mers are then hashed to unique integer values using the Python hash() function, transforming them into numerical features. In addition to k-mer encoding, label encoding and one-hot vector encoding are applied to the sequences. Label encoding assigns a unique number to each nucleotide, transforming the sequence into an array of integers. One-hot vector encoding provides a binary vector representation for each nucleotide, offering a distinct and comprehensive numerical representation.

C) Dataset Splitting:

The final preprocessing step involves splitting the data into training, validation, and testing sets.
This division is critical for evaluating the models' performance and for avoiding overfitting.
A standard split of 70% training, 15% validation, and 15% testing will be used. The processed datasets will be utilized to train and test the GA-CNN and CNN-ELM models. Each model will be evaluated on both the specific virus dataset and the viral family dataset, following the respective encoding methods.

3.2 Classification Methods:

3.2.1 Genetic Algorithm (GA) CNN Optimization

The Genetic Algorithm is a heuristic search and optimization technique inspired by the process of natural selection in biological evolution [10]. In the context of neural network optimization, GA is employed to CNNs by optimizing their weights and hyperparameters [10],[26]. This optimization aims to improve the model's accuracy in classifying viral DNA sequences. The GA optimization process generally works as follows:

1) Population Initialization:
In the initial step, a population of potential solutions (chromosomes) is generated. Each chromosome represents a specific configuration of the CNN, comprising weights and hyperparameters.
The initial population is created randomly, ensuring a diverse range of solutions.

2) Fitness Evaluation:
Each chromosome is assessed based on its fitness, which is determined by the performance of the CNN configuration it represents.
The fitness function is typically the accuracy or the loss function of the CNN when applied to a validation dataset. The fitness score reflects how well the chromosome's CNN configuration can classify the viral DNA sequences.

3) Selection Process:
Chromosomes are selected for reproduction based on their fitness scores. Selection methods like roulette wheel selection or tournament selection are utilized, which prioritize chromosomes with higher fitness but also maintain diversity in the gene pool.

4) Crossover and Mutation:
Crossover (recombination) and mutation are genetic operators used to generate new offspring (solutions). Crossover combines the features of two parent chromosomes to create offspring, introducing new solution variations. Mutation randomly alters parts of a chromosome, enabling the exploration of new regions in the solution space.

5) Iterative Process:
The GA undergoes multiple iterations or generations, with each generation consisting of the selection, crossover, and mutation processes. Throughout these generations, the population of chromosomes evolves, ideally leading to an increase in the average fitness of the population.

6) Convergence and Solution:

The GA concludes when it reaches a predefined stopping criterion, such as a maximum number of generations or a plateau in fitness improvement. The best-performing chromosome at the end of the GA process is selected as the optimal solution. This chromosome's CNN configuration is then used for the final model deployment.

The GA optimizes the CNN by finding the best possible set of weights and hyperparameters that result in the highest classification accuracy. This optimization process is crucial as it fine-tunes the CNN to better capture and learn from the complex. The fitness function used in this study is a combination of classification accuracy and a loss function measure. A higher fitness score indicates a more accurate and reliable CNN model for classifying viral sequences. After the fitness evaluation of each generation, the GA algorithm updates the CNN weights and hyperparameters according to the best solutions found. This process is repeated iteratively, with the CNN model undergoing continuous refinement with each GA iteration. After the fitness evaluation of each generation, the GA algorithm updates the CNN weights and hyperparameters according to the best solutions found. This process is repeated iteratively, with the CNN model undergoing continuous refinement with each GA iteration. Table 4 Summarizes the GA layer used in this study.

Table 4: Summary of the GA optimization layer

Step	Description	Parameters and Values
Population Initialization	Generate initial population of chromosomes	50
Fitness Evaluation	Evaluate fitness of each chromosome	CNN accuracy
Parent Selection	Select chromosomes for reproduction	Selection Method: Roulette wheel
Crossover	Combine features of parent chromosomes	Crossover Probability: 70%
Mutation	Introduce random changes in offspring	Mutation Probability: 1%
Iteration	Repeat selection, crossover, and mutation	
Convergence	Algorithm concludes	Fitness stagnation method

3.2.2 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are deep neural networks predominantly used in the field of machine learning, especially for analyzing image data and are used and highly effective in other types of structured data analysis, including time-series and genetic sequence data [9],[27,28]. In the context of viral DNA sequence classification, CNNs serve a critical role. CNNs automatically extract features from input sequences without the need for manual feature engineering. They are adept at identifying patterns and motifs within the viral DNA sequences, crucial for accurate classification. In the GA-CNN-LSTM model [9], CNN acts as the initial layer responsible for feature extraction. The extracted features are then passed to the LSTM layers for sequence modeling, capturing the temporal dynamics within the viral sequences. The GA optimizes the CNN part of the model, tuning its weights and hyperparameters to improve classification accuracy. The CNN-ELM model employs CNN as the feature extraction layer. The features extracted by CNN are fed into the ELMs for classification. This combination leverages the strength of CNN in feature extraction and the efficiency of ELM in classification, making the model both powerful and fast. Table 5 shows a summary of the CNN layer utilized as a feature extraction layer in the CNN-ELM architecture. Table 6 shows a summary of the CNN layer utilized in the GA-CNN architecture.

Table 5: Summary of the CNN layer utilized as a feature extraction layer in the CNN-ELM architecture.

Layer	Parameters
Input Layer	K-mer encoded of DNA sequence
Conv1	filters=32, kernel=4
Pool1	pool_size=2
Conv2	filters=64, kernel=4
Pool2	pool_size=2
Flattening	
FC1	neurons=256
Dropout	rate=0.5

Table 6: Summary of the CNN layer utilized in the GA-CNN architecture.

Layer	Description
Input Layer	1D input of encoded DNA sequences
Convolutional Layer 1	128 filters, Kernel size: 2x2, Activation: ReLU
MaxPooling Layer 1	Pool size: 2x2
Convolutional Layer 2	64 filters, Kernel size: 2x2, Activation: ReLU
MaxPooling Layer 2	Pool size: 2x2
Fully Connected Layer	Feature integration
Output Layer	Softmax, Multinomial probability distribution for each specific virus

3.2.3 Extreme Learning Machines (ELMs)

Extreme Learning Machines (ELMs) represent a distinct and efficient category of feedforward neural networks known for their speed and simplicity [29,30]. ELMs are significantly faster than traditional neural networks due to their non-iterative training process, which involves a one-step learning approach. This speed is crucial in handling large datasets, a common scenario in genomic studies. In the CNN-ELM model, the role of ELM is to classify viral DNA sequences based on features extracted by the CNN layer. This integration capitalizes on the strength of CNNs in feature extraction and the efficiency of ELMs in classification. The ELM serves as the final classification layer, receiving the high-dimensional feature vectors processed by the CNN and efficiently mapping them to the output classes. The ELM in the CNN-ELM model consists of a hidden layer with a predefined number of neurons. The activation function used is typically sigmoid or Gaussian, enabling the network to handle the non-linear relationships in the data. The model's output layer corresponds to the number of virus families or specific virus types in the classification task. The selection of Extreme Learning Machines in this study is a strategic choice driven by the need for a fast, efficient, and accurate classification method in the realm of viral DNA sequence analysis. Table 7 shows the summary of the ELMs layer.

Table 7: The summary of the ELM layer used.

Layer	Description
Input Layer	256 inputs from the CNN feature extraction layer
Hidden Layer	128 Neurons
Output Layer	10 neurons with softmax activation function
Layer	Description

3.2.4 CNN-LSTM and CNN-Bidirectional LSTM Layers:

LSTM (Long Short-Term Memory) networks are a type of Recurrent Neural Network (RNN) specifically designed to learn long-term dependencies in sequence data [31]. In the context of viral DNA sequence classification, LSTM layers are used to predict classification labels based on the features extracted by the CNN layers. The integration of LSTM with CNN allows the model to capture both the local features (through CNN) and the sequential dependencies (through LSTM) in the DNA sequences. The bidirectional LSTM (BiLSTM) extends the LSTM capability by processing the sequence data in both forward and reverse directions, providing a more comprehensive understanding of the sequence context. The same architecture and hyperparameters are used as in [9].

A complete system architecture showing a summary of the classification methodology is shown in Figure 1.

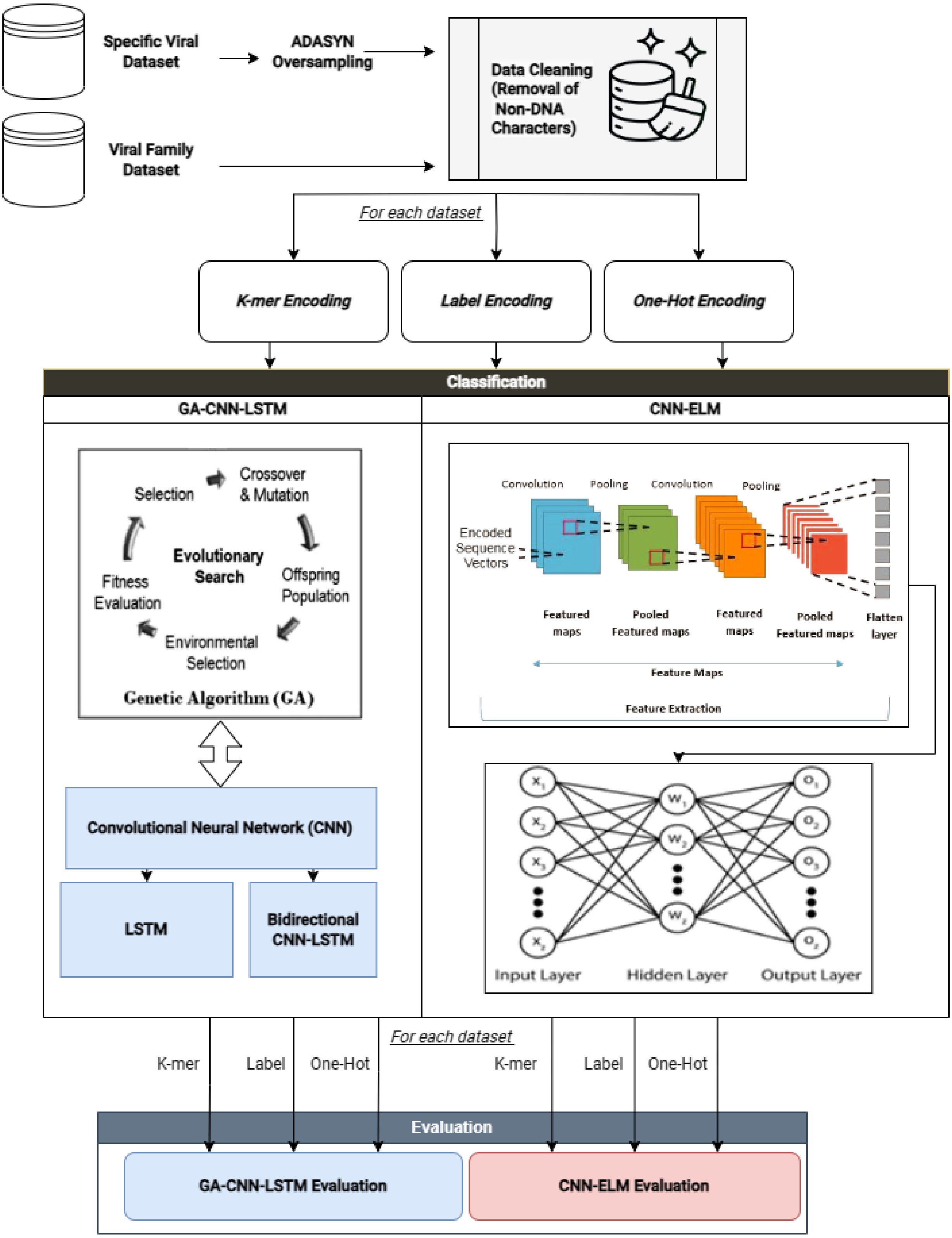

Figure. 1: System architecture of the proposed classification methodology

4. Experimental Results

For the experiments, a NVIDIA GeForce GTX 3060 graphics card with 12GB of VRAM. The processing power was supplied by an Intel Core i7-11390H processor, supported by 64GB of RAM. The software environment utilized Python 3.8 and TensorFlow 2.4 were used mainly for constructing and training the deep learning models, favored for their flexibility and ease of use in neural network applications. The datasets were divided following a standard split ratio of 70% for training, 15% for validation, and 15% for testing. This division was used to ensure a balanced approach to model training and evaluation. The Genetic Algorithm (GA) optimization parameters included a population size of 50, a crossover rate of 70%, a mutation rate of 1%, and a total of 50 generations and selection used roulette wheel method. First, three different encoding methods are evaluated using GA-CNN and CNN-ELM models across the two datasets. Table 8 and 9 show the resultant accuracies for both datasets respectively for the different encoding methods.

Table 8: Encoding Methods Comparison using GA-CNN Model

Encoding Method	**Accuracy on Specific Viral Dataset (%)**	**Accuracy on Viral Families Dataset (%)**
k-mer (k=4)	**93.45**	**95.88**
One-hot Vector	91.30	94.80
Label Encoding	90.25	94.20

Table 9: Encoding Methods Comparison using CNN-ELM Model

Encoding Method	**Accuracy on Specific Viral Dataset (%)**	**Accuracy on Viral Families Dataset (%)**
k-mer	**92.70**	**94.54**
One-hot Vector	90.60	93.80
Label Encoding	89.55	93.20

The results displayed in Tables 8 and 9 indicate that k-mer encoding consistently achieved the highest accuracy across both datasets for both models. Therefore, the models will be analyzed further based on k-mer encoding as our main encoding methodology for the rest of the results. The GA-CNN model was rigorously tested on the Viral Families Dataset, consisting of 50,000 sequences. The primary objective was to assess the model's adaptability and classification accuracy on a diverse range of viral families. During the training phase, the GA-CNN model exhibited a learning curve, achieving a peak training accuracy of 97.5%. On the testing set, which comprised 15% of the entire dataset, the GA-CNN model maintained a consistent performance with an accuracy of 95.88%. The genetic algorithm's optimization of CNN weights played a pivotal role in this adaptability, enabling the model to effectively handle the broader and more complex nature of the dataset. Table 10 shows the confusion matrix of the model across all the viral families.

The CNN-ELM was utilized on the specific viral dataset as well for comparative analysis and to test its accuracy on these viruses. In the training phase, the CNN-ELM model demonstrated a training accuracy of 93.6%. This indicates a strong learning ability but slightly lower compared to its performance on the viral families dataset. The model achieved a validation accuracy of 92.70%. With a loss value of 0.18, the CNN-ELM model maintained a good level of precision. However, the slightly increased loss compared to its performance on the viral families dataset hints at challenges the model faces when dealing with more specific and potentially varied viral sequences. The CNN-ELM model exhibited a remarkable efficiency in terms of training time. Despite its slightly reduced accuracy on the specific viral dataset, the model maintained a swift training pace. The entire training process, including epochs, was completed significantly faster than the GA-CNN model, underlining the ELM's inherent advantage in rapid model training. With an accuracy of 92.7% on the specific viral dataset and 94.54% on the viral family dataset, the model successfully balanced between speed and precision. This aspect is particularly noteworthy as it demonstrates the model's potential in scenarios where rapid training is crucial, without sacrificing classification accuracy much. Table 11 shows the comparative time analysis of both models with both datasets.

Table 10: The confusion matrix of the GA-CNN Model on the Viral Family dataset across all the viral families

Virus Family	1	2	3	4	5	6	7	8	9	10
1.Coronaviridae	**733**	4	0	4	4	3	4	6	0	2
2.Flaviviridae	13	**715**	6	6	0	6	0	5	0	6
3.Togaviridae	0	0	**719**	1	0	3	1	8	1	0
4.Paramyxoviridae	5	4	4	**726**	0	0	7	6	7	4
5.Bunyaviridae	1	4	0	0	**734**	0	9	3	8	2
6.Rhabdoviridae	0	0	8	3	5	**712**	2	5	9	0
7.Filoviridae	1	0	4	4	7	0	**731**	0	9	5
8.Herpesviridae	1	2	1	9	7	5	4	**716**	6	0
9.Adenoviridae	5	3	0	4	3	1	0	4	**727**	5
10.Reoviridae	5	0	6	7	6	2	8	4	4	**721**

Table 11: Comparative Analysis of Training Times of both models on the datasets

Model	Specific Viral Dataset		Viral Families Dataset	
	Training Time per Epoch	Total Training Time	Training Time per Epoch	Total Training Time
GA-CNN	~15 minutes	~25 hours	~18 minutes	~30 hours
NN-ELM	**~10 minutes**	**~16.5 hours**	**~12 minutes**	**~20 hours**

A comparative study was conducted to assess the performance of various deep learning models in classifying viral DNA sequences. This analysis included state-of-the-art models specifically tailored for sequence data, such as Recurrent Neural Networks (RNNs), Bidirectional Long Short-Term Memory networks (BiLSTMs), and Graph Neural Networks (GNNs). For GNNs, the viral DNA sequences were transformed into graph structures, wherein each nucleotide was represented as a node, interconnected through edges based on nucleotide relationships. This allowed GNNs to capture the intricate connectivity patterns inherent in viral sequences. Similarly, RNNs and BiLSTMs were applied to the sequence data, leveraging their ability to process sequential information effectively. Table 12 shows the comparative accuracy of these models on two distinct datasets; specific viral and viral families datasets; using k-mer encoding as it gave the highest accuracy as mentioned before.

Table 12: The comparative accuracy of these models on two distinct datasets - specific viral and viral families datasets using k-mer encoding

Model	Accuracy on Specific Viral Dataset (%)	Accuracy on Viral Families Dataset (%)
GA-CNN	**93.45**	**95.88**
CNN-ELM	92.70	94.54
RNN	85.11	83.60
BiLSTM	82.50	84.20
GNN	80.80	82.70

The resultant accuracy shows that GA-CNN achieved the highest accuracy among both the specific viral and viral families dataset with accuracies 93.45% and 95.88% respectively. This superior performance shows the efficacy of combining genetic algorithm optimization with the CNN-Bidirectional model. In contrast, traditional RNNs, BiLSTMs, and GNNs, while still providing good classification capabilities, fall short in comparison to the optimized weights of the GA-CNN model.

5. Conclusion & Future Work

The realm of genomic bioinformatics faces the challenge of accurately classifying viruses. This complexity is very complex when considering the vast array of specific viruses and diverse viral families. In this study, comprehensive exploration of two models the GA-CNN and the CNN-ELM applying them to distinct datasets - one comprising specific viruses and another encompassing various viral families that was previously used in the previous studies. The results demonstrated that the GA-CNN model achieved an accuracy of 95.88% on the viral families dataset. When applied to the specific virus dataset, the GA-CNN and CNN-ELM models achieved accuracies of 93.45% and 92.70%, respectively. These results highlight the efficacy of the GA-CNN model in optimizing feature extraction for accuracy and the CNN-ELM model's rapid training capabilities. Three encoding methods were evaluated: k-mer, One-hot Vector and Label Encoding. K-mer encoding methodology achieved the highest accuracy among all other methods on both models and datasets using k=4. Furthermore, a comparative training time analysis of both GA-CNN and CNN-ELM was performed which showed that the CNN-ELM is significantly faster than the more complex GA-CNN and gives a good classification accuracy still on both datasets. Finally, a comparative study was performed with other deep learning methods like RNN, BiLSTM and GNN which proved that GA-CNN outperformed and achieved the highest classification accuracies on both datasets as well.

Future work will focus on integrating additional advanced deep learning architectures and exploring more sophisticated optimization techniques. The goal is to further refine the classification accuracy and adaptability of these models. Also, analysis with other traditional methods such as BLAST and multiple sequence analysis might be explored further.

References

1. Matthews, R. E. F. "Classification and nomenclature of viruses." Intervirology 17.1 (1982): 199.

2. Wigginton, Krista Rule, and Tamar Kohn. “Virus disinfection mechanisms: the role of virus composition, structure, and function.” Current opinion in virology 2.1 (2012): 84-89.
3. Pareek, Chandra Shekhar, Rafal Smoczynski, and Andrzej Tretyn. "Sequencing technologies and genome sequencing." Journal of applied genetics 52 (2011): 413-435.
4. Sayers, Eric W., et al. "Database resources of the national center for biotechnology information." Nucleic acids research 49.D1 (2021): D10.
5. Yang, Aimin, et al. "Review on the application of machine learning algorithms in the sequence data mining of DNA." Frontiers in Bioengineering and Biotechnology 8 (2020): 1032.
6. Abd–Alhalem, Samia M., et al. "DNA sequences classification with deep learning: a survey." Menoufia Journal of Electronic Engineering Research 30.1 (2021): 41-51.
7. Lo Bosco, Giosuè, and Mattia Antonino Di Gangi. "Deep learning architectures for DNA sequence classification." Fuzzy Logic and Soft Computing Applications: 11th International Workshop, WILF 2016, Naples, Italy, December 19–21, 2016, Revised Selected Papers 11. Springer International Publishing, 2017.
8. Millán Arias, Pablo, et al. "DeLUCS: Deep learning for unsupervised clustering of DNA sequences." Plos one 17.1 (2022): e0261531.
9. El-Tohamy, Ahmed, Huda Amin Maghwary, and Nagwa Badr. "A Deep Learning Approach for Viral DNA Sequence Classification using Genetic Algorithm." International Journal of Advanced Computer Science and Applications 13.8 (2022).
10. Katoch, Sourabh, Sumit Singh Chauhan, and Vijay Kumar. "A review on genetic algorithm: past, present, and future." Multimedia tools and applications 80 (2021): 8091-8126.
11. El-Tohamy, Ahmed, Huda Amin Maghwary, and Nagwa Badr. " A Combined ELM and CNN Model Architecture for Accurate Viral Family DNA Classification". International Conference on Intelligent Computing and Information Systems (ICICIS) 2023.
12. Alzubaidi, Laith, et al. "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions." Journal of big Data 8 (2021): 1-74.
13. Albawi, Saad, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network." 2017 international conference on engineering and technology (ICET). Ieee, 2017.
14. Ding, Shifei, Xinzheng Xu, and Ru Nie. “Extreme learning machine and its applications.” Neural Computing and Applications 25 (2014): 549-556.
15. Liu, Nannan, et al. "ACO-KELM: Anti Coronavirus Optimized Kernel-based Softplus Extreme Learning Machine for Classification of Skin Cancer." Expert Systems with Applications (2023): 120719.
16. Ming, Zhaoyan, et al. “HostNet: improved sequence representation in deep neural networks for virus-host prediction.” BMC bioinformatics 24.1 (2023): 455.
17. Humayun, Fahad, et al. "Computational method for classification of avian influenza A virus using DNA sequence information and physicochemical properties." Frontiers in Genetics 12 (2021): 599321.
18. Alakus, Talha Burak, and Muhammet Baykara. "Comparison of Monkeypox and wart DNA sequences with deep learning model." Applied Sciences 12.20 (2022): 10216.
19. Tampuu, Ardi, et al. "ViraMiner: Deep learning on raw DNA sequences for identifying viral genomes in human samples." PloS one 14.9 (2019): e0222271.
20. Wang, Peng, Xiaomin Zhang, and Yan Hao. "A method combining CNN and ELM for feature extraction and classification of SAR image." Journal of Sensors 2019 (2019): 1-8.
21. Gunasekaran, Hemalatha, et al. "Analysis of DNA sequence classification using CNN and hybrid models." Computational and Mathematical Methods in Medicine 2021 (2021).

22. He, Haibo, et al. "ADASYN: Adaptive synthetic sampling approach for imbalanced learning." 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). Ieee, 2008.
23. Satam, Heena, et al. "Next-generation sequencing technology: Current trends and advancements." Biology 12.7 (2023): 997.
24. Heinis, Thomas, Roman Sokolovskii, and Jamie J. Alnasir. "Survey of information encoding techniques for dna." ACM Computing Surveys 56.4 (2023): 1-30.
25. Dietterich, Thomas G. "Machine learning for sequential data: A review." Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshops SSPR 2002 and SPR 2002 Windsor, Ontario, Canada, August 6–9, 2002 Proceedings. Springer Berlin Heidelberg, 2002.
26. Loussaief, Sehla, and Afef Abdelkrim. "Convolutional neural network hyper-parameters optimization based on genetic algorithms." International Journal of Advanced Computer Science and Applications 9.10 (2018).
27. Li, Zewen, et al. “A survey of convolutional neural networks: analysis, applications, and prospects.” IEEE transactions on neural networks and learning systems (2021).
28. Gomes, Ruither AL, and F. Murilo Zerbini. "ConCreT, a 2D convolutional neural network for taxonomic classification applied to viruses in the phylum Cressdnaviricota." Journal of Virological Methods 320 (2023): 114789.
29. Wang, Jian, et al. “A review on extreme learning machine.” Multimedia Tools and Applications 81.29 (2022): 41611-41660.
30. Huang, Guang-Bin, et al. "Extreme learning machine for regression and multiclass classification." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42.2 (2011): 513-529.
31. Malhotra, Pankaj, et al. "Long Short Term Memory Networks for Anomaly Detection in Time Series." Esann. Vol. 2015. 2015.