Viral Sequence Classification with Deep Learning

Filip Trajkovski Emanuele Pase Ingeborg M. Klemsdal

[1]Department of Information Engineering, University of Padova , Padova, Italy.
19th June 2024

Abstract

This project explores and extends the concepts introduced in the Viraminer[1] paper through the development and evaluation of alternative models. Viraminer proposed a novel approach for mining viral sequences from metagenomic data using a sequence-based clustering method. Our study builds upon this foundation by introducing several alternative models, designed with the aim of enhance performance, scalability, and accuracy in viral sequence identification.

1 Introduction

Human bodies are made up of cells, and in each of these cells, there is a nucleus that contains the genetic recipe that codes for all the proteins used to build the body, DNA. This genetic material consists of four bases A, T, C, and G. In the double-stranded helix of DNA, these bases form pairs: A-T and C-G. Knowing this, all the information is then retained when looking at a single strand and the order in which the bases are lined.

The body contains not only cells but has a diverse population of microorganisms, of which some are viruses. These viruses play a vital part in human well-being and are called the human virome. For a long time medical research has shown the danger of certain viruses entering the body and introducing deceases. Because of this, it is important to be able to differentiate between the human virome and unknown viruses that may pose a danger [2].

One suggestion of how we could utilize Deep Learning to solve the problem related to the "unknown" viral samples was ViraMiner. A research article from 2018 suggested a neural network architecture to tackle this. They use a two-branched model of Convolutional neural network(CNN) on raw metagenomic contigs to identify viruses across different human samples [1]. Already their architecture received great results with an area under the ROC-curve (Receiver operating characteristic) of 0.923.

In our day and age, AI technologies and techniques are rapidly expanding. That is why this project hopes to explore more options on how to create the architecture for the classification problem and see if there is a way to improve upon the results from ViraMiner. We will look at different neural networks and compare them against each other.

2 Methods

2.1 Dataset

The dataset for the ViraMiner-project was created from samples belonging to different, human patient groups. According to the paper, the data was created by 19 NGS(Next-generation sequencing) experiments. This technique is used for DNA sequencing and variant/mutation detection[3]. After this, they analyzed the sequences using PCJ-BLAST. more specified details can be found in the ViraMiner paper.

When finalizing the dataset, they decided to divide the sequences into equal parts of 300bp and the remaining base pairs would be removed. This meant that a viral sequence of 650bp would be made into two viral sequences of 300bp and the 50bp in the end would be removed [1].

2.2 ViraMiner

The entirety of the architecture for each branch in ViraMiner can be appreciated in 6. Here we can see the Frequency model and the Pattern model, both convolutional neural networks(CNNs)

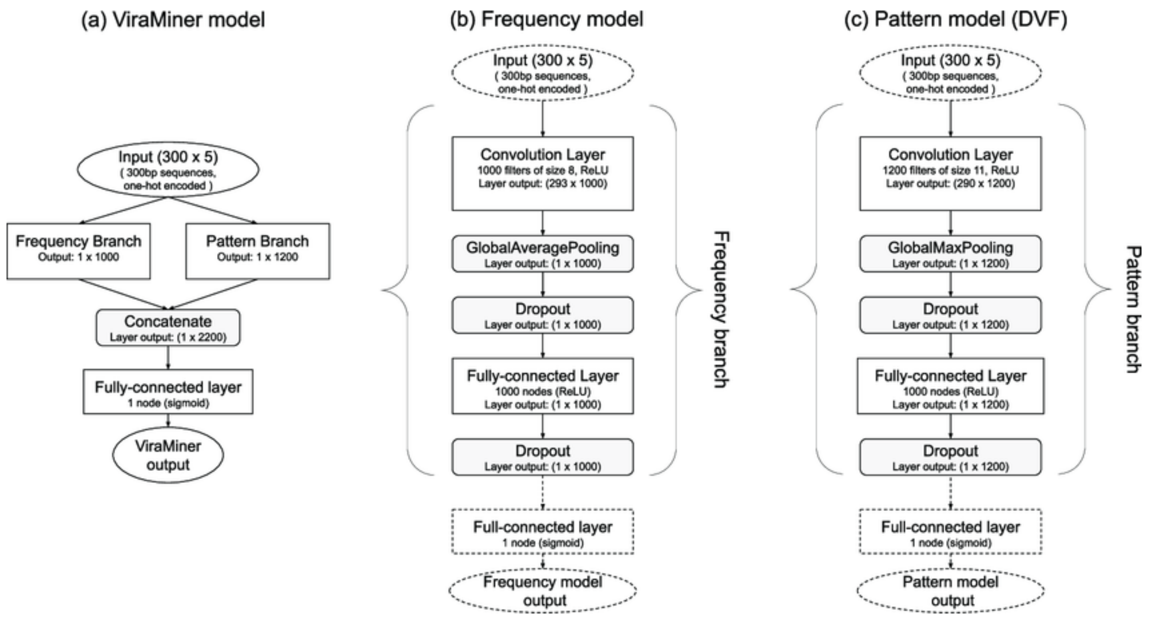
(a) ViraMiner model
Input (300 x 5) (300bp sequences, one-hot encoded)
Frequency Branch Output: 1 x 1000
Pattern Branch Output: 1 x 1200
Concatenate Layer output: (1 x 2200)
Fully-connected layer 1 node (sigmoid)
ViraMiner output
(b) Frequency model
Input (300 x 5) (300bp sequences, one-hot encoded)
Convolution Layer 1000 filters of size 8, ReLU Layer output: (293 x 1000)
GlobalAveragePooling Layer output: (1 x 1000)
Dropout Layer output: (1 x 1000)
Fully-connected Layer 1000 nodes (ReLU) Layer output: (1 x 1000)
Dropout Layer output: (1 x 1000)
Full-connected layer 1 node (sigmoid)
Frequency model output
Frequency branch
(c) Pattern model (DVF)
Input (300 x 5) (300bp sequences, one-hot encoded)
Convolution Layer 1200 filters of size 11, ReLU Layer output: (290 x 1200)
GlobalMaxPooling Layer output: (1 x 1200)
Dropout Layer output: (1 x 1200)
Fully-connected Layer 1000 nodes (ReLU) Layer output: (1 x 1200)
Dropout Layer output: (1 x 1200)
Full-connected layer 1 node (sigmoid)
Pattern model output
Pattern branch

Figure 1: ViraMiner Architecture

2.3 Our Models

In this subsection, we graphically represent the various architectures of the different models developed, in order to see the differences between each other and which one worked best.

2.3.1 LSTM and Bi-LSTM

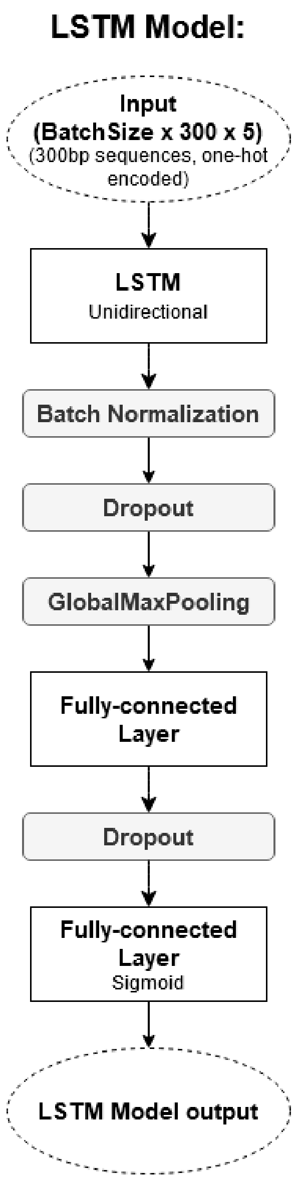

Figure 2: LSTM model Architecture

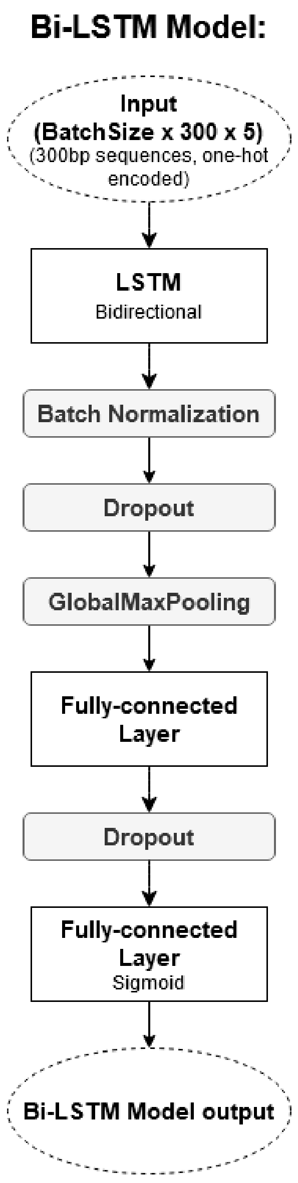

Figure 3: Bi-LSTM model Architecture

These architectures are chosen to ensure that the model can consider features related to the spatial position of sequences, as this is a fundamental characteristic of DNA. This is crucial both for encoding individual amino acids and for potential protein translation (with a specific three-dimensional structure).

2.3.2 GRU and Bi-GRU

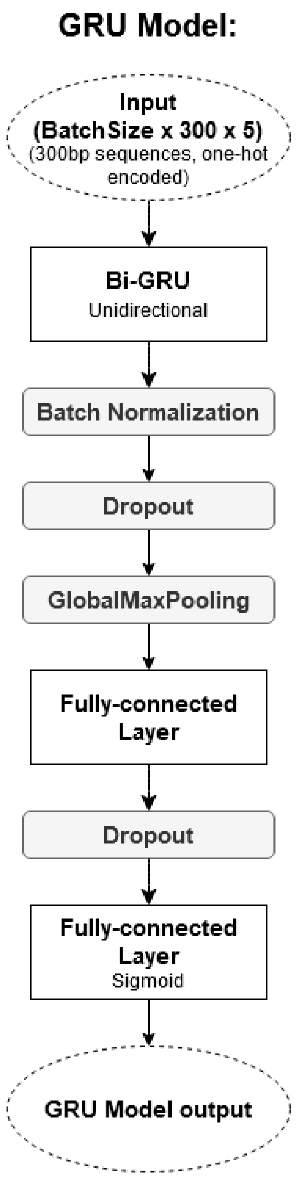

Figure 4: GRU model Architecture

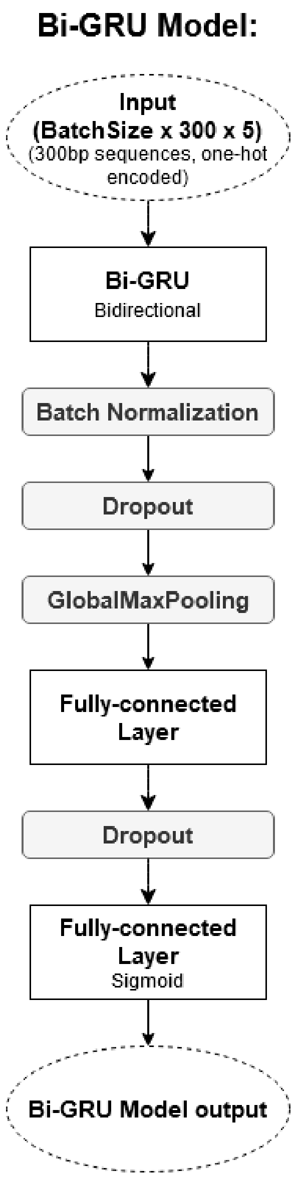

Figure 5: Bi-GRU model Architecture

These architectures are used for comparison with LSTM and BI-LSTM, as they are derived from the latter. The results of this comparison can help to better understand whether the gate mechanism that allows forgetting certain features can help reduce parameter interdependence (in addition to making the model less complex due to the fewer number of parameters).

2.3.3 DEEP-CNN

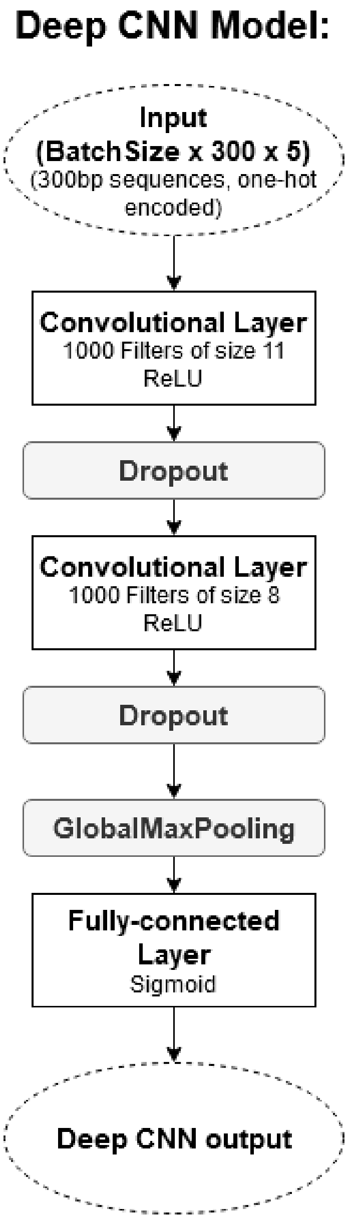

Figure 6: Deep-CNN model Architecture

This model is created using architectural features described in the ViraMiner[1] paper. It serves as a comparison with Viraminer's[1] final model and its two branches: frequency and pattern.

2.3.4 CNN-LSTM and CNN-Bi-LSTM Hybrid models

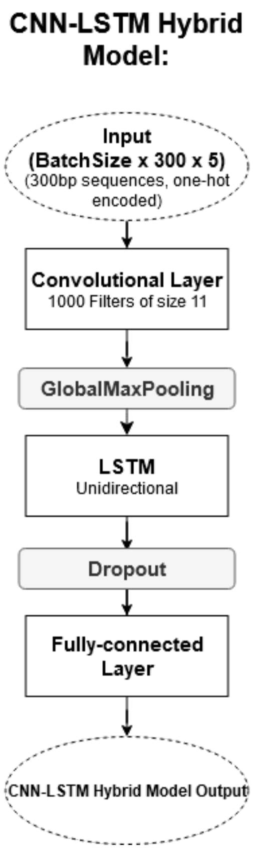

Figure 7: CNN-LSTM Hybrid model Architecture

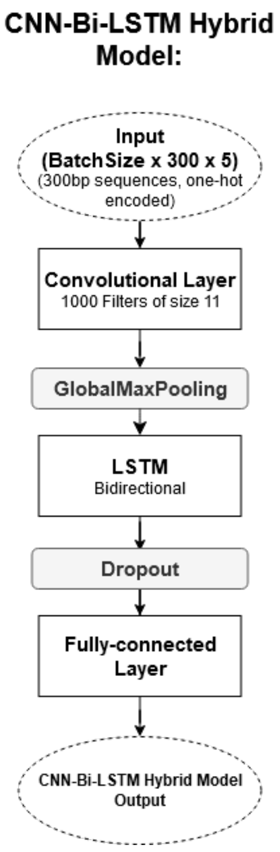

Figure 8: CNN-Bi-LSTM Hybrid model Architecture

These hybrid models fuse the architectural characteristics of the Deep CNN model with the LSTM model seen previously.

2.3.5 CNN-GRU and CNN-Bi-GRU Hybrid models

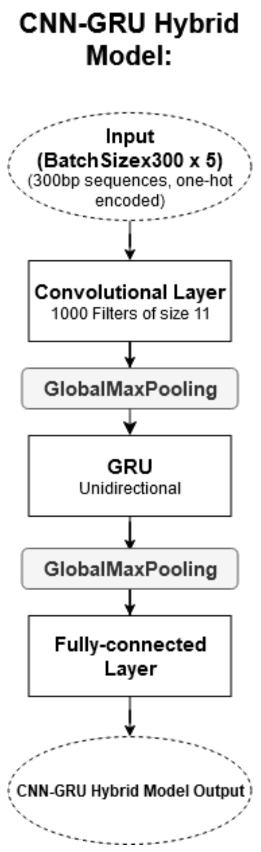
CNN-GRU Hybrid Model:
Input (BatchSizex300 x 5) (300bp sequences, one-hot encoded)
Convolutional Layer 1000 Filters of size 11
GlobalMaxPooling
GRU Unidirectional
GlobalMaxPooling
Fully-connected Layer
CNN-GRU Hybrid Model Output

Figure 9: CNN-GRU Hybrid model Architecture

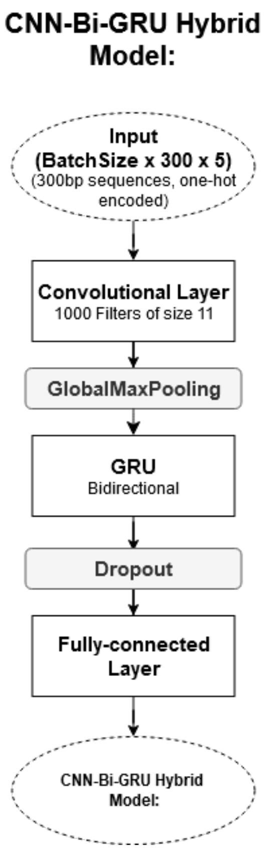

Figure 10: CNN-Bi-GRU Hybrid model Architecture

Similarly to the LSTM-Hybrid models, these models combine features of the Deep CNN model with GRU architectures.

3 Results and discussion

3.1 Learning rate analysis

In this section, we will examine how changes in the learning rate affect the accuracy of the models. The following tests were conducted by varying only the learning rate and the optimizer while keeping all other hyperparameters constant.

Optimizer:	**Method:**	**Pattern model (ROC)**	**Frequency model (ROC)**	**Final model (ROC)**
ADAM	Exponential decay (paper)	0.88803	0.87896	0.89079
	Exponential step decay	0.87513	0.85678	0.87637
	Linear	0.87206	0.82127	0.87217
	Linear step decay	0.86702	0.81869	0.86524
SGD (with momentum = 0.9)	Exponential decay	0.69795	0.52438	0.75627
	Linear decay	0.65736	0.51591	0.72212
SGD (with momentum = 0.75)	Linear decay	0.61254	0.51247	0.70584

Table 1: Learning rate analysis for different optimizers and methods

As we can see the Learning Rate and Optimizer used by Paper researchers, using the same architecture and the same hyperparameters, are optimal according to our tests. The values that come closest to their result are the ones from Exponential Step Decay LR with ADAM Optimizer.

3.2 Models

Model	**AUROC**
ViraMiner	0.923
LSTM	0.723
Bi-LSTM	0.851
GRU	0.810
Bi-GRU	0.876
Deep CNN	0.897
CNN-GRU Hybrid	0.890
CNN-Bi-GRU Hybrid	0.495
CNN-LSTM Hybrid	0.883
CNN-Bi-LSTM Hybrid	0.894

Table 2: Performance Metrics of Different Models

4 Conclusion

Based on the results obtained from our project, it is evident that we have explored several alternative models for viral sequence identification, each with varying degrees of success. While our primary objective was to improve upon the performance metrics established by Viraminer, our findings indicate a mixed outcome. Among the models evaluated, ViraMiner demonstrated the highest AUROC of 0.923, setting a strong benchmark in viral sequence mining from metagenomic data. Our efforts with LSTM, Bi-LSTM, GRU, Bi-GRU, Deep CNN, and various hybrid models such as CNN-GRU, CNN-Bi-GRU, CNN-LSTM, and CNN-Bi-LSTM yielded results that, while not surpassing ViraMiner, still showed substantial promise. In conclusion, while our project did not achieve superior results compared to ViraMiner, we have successfully developed and evaluated several models that demonstrate significant potential in enhancing the performance, scalability, and accuracy of viral sequence identification from metagenomic data.

5 Instructions

The Colab file provided includes links to shared Google Drive® folders. To ensure access from your personal workspace, it's recommended to create shortcuts to these shared folders in the "My Drive" section of Google Drive®. The necessary folders are:

- dataset: This folder contains the datasets already divided into training, validation, and test sets. Newly trained models will also be saved here to avoid overwriting the previously created ones;
- DeepLearningModels: This folder contains all the models already trained for 30 epochs.

List of Figures

List of Tables

Bibliography

[1] D. J. R. Tampuu A Bzhalava Z, ‘Viraminer: Deep learning on raw dna sequences for identifying viral genomes in human samples’, *PLOS ONE 14(9): e0222271*, 2019, Accessed on 20.05.2024.

[2] F. Liang G. Bushman, ‘The human virome: Assembly, composition and host interactions’, *Nat Rev Microbiol 19, 514–527*, 2021, Accesed on 20.05.2024.

[3] Q. D, ‘Next-generation sequencing and its clinical application’, *Cancer Biol Med. 2019 Feb;16(1):4-10*, 2019, Accessed on 31.05.2024.