Continual Learning for Multiclass Classification of Viral Genomic Sequences

Vito Nicola Losavio [*] [1]

Abstract

Viral genomics is essential for understanding infectious diseases, yet traditional methods struggle with the complexity of viral sequences. This project leverages deep learning for multiclass classification of viral DNA and RNA sequences, integrating continual learning to keep models accurate over time.

Using the ”ncbi-virus-complete-dna-v230722” dataset from Hugging Face, we focus on four virus families after preprocessing: Geminiviridae, Hepadnaviridae, Orthomyxoviridae, and Circoviridae. The dataset is balanced using SMOTEN, resulting in 60,732 samples.

We developed a CNN-LSTM model, employing continual learning by dividing the dataset into ten batches. This approach allows the model to adapt to new data while retaining knowledge, ensuring high accuracy and relevance.

Our model demonstrates robust performance, proving suitable for real-time viral classification on devices with limited resources, maintaining high accuracy and adaptability to evolving viral genomes.

The code is available at the following link: GitHub

1. Introduction

In recent years, viral genomics has become crucial in studying infectious diseases, enabling a deeper understanding of the diversity and characteristics of viruses. Classifying viral DNA and RNA sequences is essential for identifying and characterizing new viruses, monitoring their evolution, and predicting potential outbreaks. However, the complexity and variability of viral genomic sequences pose significant challenges to traditional classification methods.

Machine learning offers powerful tools to address these challenges by handling large volumes of data and extracting complex patterns. Deep learning models, in particular, have shown remarkable success in multiclass classification tasks for genomic sequences. Despite their effectiveness, these models often require continual adaptation and improvement to remain accurate over time, especially given the rapid evolution of viral genomes.

Continual learning (Delange et al., 2021), is a methodology that allows models to continuously learn from new data without forgetting previously acquired knowledge. This approach is particularly beneficial for the classification of viral sequences, as it enables the model to adapt to new variants and evolving genomic information. By incorporating continual learning, the model can maintain high performance and relevance in the face of dynamic and ever-changing data.

This project aims to develop a deep learning model for the multiclass classification of viral DNA and RNA sequences, integrating continual learning techniques to ensure the model remains up-to-date and accurate. The goal is to develop a model that accurately classifies viral genomic sequences and can be trained on devices with limited resources.

In this report we will analyze:

1) The composition of the dataset and how it was analyzed trying to reduce the complexity of the project.

2) The model developed.

3) The training techniques used for the conduct of the classification.

4) The result obtained

2. State of Art

In recent years, the classification of DNA and RNA sequences has benefited significantly from advances in deep learning. Numerous studies have explored the use of convolutional neural networks (CNN) and recurrent neural networks (RNN), showing how these models can capture complex nucleotide dependencies and improve predictive performance.

[*] Equal contribution [1] Department of Computer Science, University of Bari Aldo Moro, Bari, Italy. Correspondence to: Vito Nicola Losavio <v.losavio5@studenti.uniba.it>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Our starting point for the development of this task is the paper by Ahmed El-Tohamy et al.(El-Tohamy et al., 2022) who have developed several deep learning architectures for the classification of viral sequences, these have developed architectures formed by CNN and LSTM with a genetic algorithm, starting from an unbalanced dataset oversampled combining everything with an approach of automatic selection of characteristics to overcome the challenges in manual extraction of these characteristics.

In addition to the work we took as a starting point, there are several works that vary in complexity, a work not very recent is the work of Tampuu et at (Tampuu et al., 2019), which they proposed ViraMiner, a deep learning model designed to identify viral genomes in human samples through raw DNA sequences. ViraMiner uses an architecture that combines branches of Pattern and Frequency, which extract information about patterns and frequencies of patterns respectively. These branches are trained separately and then combined to obtain a complete model that has shown excellent performance in human metagenomic data testing. A more complex computationally approach is proposed by Shiraj et al. (Shiraj & Yousuf, 2024) who developed a BERT(Devlin et al., 2019)-based transformer called virusBERT.

2.1. Continual Learning

The field of viral genomics deals with the study of viral DNA and RNA sequences, which are known for their high mutation rates and rapid evolution. This dynamic nature poses a significant challenge for classification models, as the introduction of new viral strains can quickly render a previously accurate model obsolete. Continual learning addresses this issue by allowing models to update and incorporate new data, ensuring that they remain relevant and accurate as new viral sequences emerge.

For example Alharbi et al.(Alharbi et al., 2023) use continual learning to overcome the computational and time-consuming limitations of deep learning models, allowing them to add new classes without having to start from scratch with model training. In addition, it has been used in conjunction with few-shot learning methods to address the classification of plant diseases in agriculture, allowing for better results with a limited amount of training data.

3. Material

The starting dataset is the Hugging Face dataset ”ncbi-virus-complete-dna-v230722” which contains 2.661.137 lines.

The previously mentioned dataset contains over 100 different types of viruses and is composed of several features, among which we find the length of the sequence. We focus on this feature because in the first phase of preprocessing, we will remove all data that have a sequence length greater than 4000 elements.

That said we will have a dataset with a reduced number of elements, to further reduce the task, we will focus attention on the 4 viruses that appear most in this filtered dataset.

The designated viruses are: Geminiviridae, Hepadnaviridae, Orthomyxoviridae and Circoviridae. After this further filtering, we obtained a dataset containing 47.963 rows, but the dataset is slightly unbalanced.

To make the dataset balanced, an over-sampling process was carried out through SMOTEN(Chawla et al., 2002), to have an over-sampling dataset of 60.732 lines in total.

3.1. Multiclass Preprocessing

In carrying out the task, we focused on the sequence that represent the virus. That said, the different features were eliminated except for the feature target and a dataset was created containing 4.000 columns corresponding to the nucletotids in the sequence.

The next step was to encode the sequence in numerical values, to do this we went to replace the nucletide corresponding to ’A’ the value 1, to ’C’ the value 2, to ’G’ the value 3 and finally to’T’ the value 4 (Fig:1).

The new features are also scaled using a MinMaxScaler, going to bring the possible values between 0 and 1 in order to improve the classifications(de Amorim et al., 2023).

Since the sequences have a variable length depending on the sample, a padding of -1 is added that we will then macherare inside the model.

For feature targets, the LabelEncoder was used to encode the target features in numerical features.

3.2. Continual learning setup

After completing the preprocessing of the data, the dataset was divided into 10 distinct batches to establish a continual learning setting.

Initially, the entire preprocessed dataset was randomly shuffled to ensure each batch was representative of the overall dataset, thereby reducing bias and improving robustness. The shuffled dataset was then divided into 10 equal parts, creating 10 batches that underwent independent training, validation, and testing.

For each batch, the data was split as follows: 70% for the training set, which is used to learn the underlying patterns and features of the viral genomic sequences; 10% for the validation set, which helps in tuning the model’s hyperparameters and preventing overfitting by providing a checkpoint during the training process; and 20% for the test set, which is used to evaluate the model’s performance and gen-

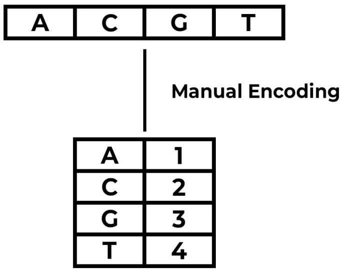

Figure 1. Encoding used for the development of the task

eralizability on unseen data.

This section has achieved the objective **1)**.

4. Models

The proposed model is a CNN-LSTM based model, composed of two CNN layers anticipated by a layer of Masking to ignore the added padding. The activation function used is the Relu, which in experimental value has good performance.

The CNN layers are alternated with a MaxPooling1D in order to take the subset of the data of maximum activation on a window composed of 2 elements halving the length.

Then we have an LSTM layer composed of 756 units. We tried different configurations of LSTM units, finding that the right compromise between computational cost and performance is the one proposed. Finally we find two layers Dense to make the final classification with a softmax activation considered the multiclass task.

Considering that we are working with very small numerical values and the inclination to the vanishing problem turns out to be high, as experimentally we were faced with this phenomenon, we used two optimizers, the first is a container of optimizers called LossScaleOptimizer with which you avoid underflow in intermediate gradients and Adamax (Kingma & Ba, 2017) which is a variant of Adam based on the infinite norm, the use of these optimizers allowed us to avoid underfitting the model.

This section has achieved the objective **2)**.

Training The model, as mentioned above, was trained on the different batches with the technique of early stopping, through which we saved additional computation time; we chose as patience a parameter equal to 1, going to monitor the loss in validation. This choice has been made in how much going to carry out various tests, it is noticed that in the moment in which the loss in phase of validation rises, the model tended to worsen the performances.

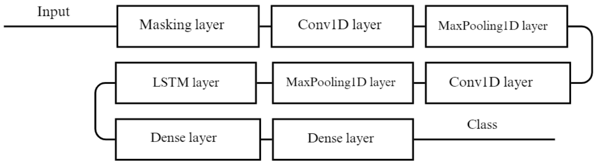

Figure 2. Model proposed

With this paragraph we achieved the objective **3)**.

5. Results and Conclusion

After having trained the models in the continual learning, a phase of test has been carried out taking 100 examples of each class and going to carry out the classification with the models that have obtained a greater accuracy during the phase of training.

We chose to evaluate the best model because continuing the training we could have encountered overfitting.

As shown in the figure 3: the model trained on the unbalanced dataset, has found good results, despite the class: 'Orthomyxoviridae' has not been recognized by the model. This class belongs to one of the minority classes of the dataset. Consequently, it is assumed that the data in the training phase were contained in test fold or that the data belonging to this class were not significant or very similar to those belonging to the class 'Hepadnaviridae'.

In figure: 4 we find the results obtained from the balanced dataset. Analyzing them, we see that the best model turns out to be overfitted on the class 'Geminiviridae'. It is as-

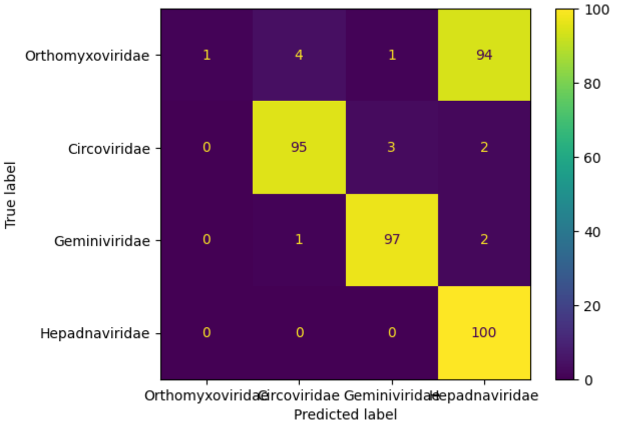

Figure 3. Result UnBalanced Best Model

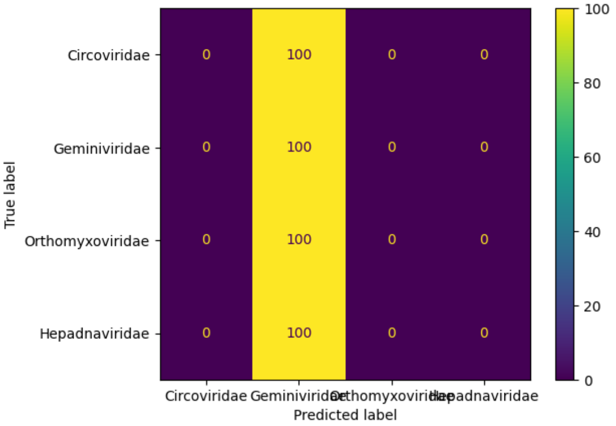

Figure 4. Result Balanced Best Model

sumed that in the fold in which it turns out to be more performant the model has a test set that contains more the class 'Geminiviridae'. This result turns out to be at an ambiguous theoretical level as having balanced the model, one expects a greater distribution of the data and a more accurate classification, considering this it is assumed that the cause of overfitting was the balancing of the dataset in which the synthetic data were found to be insignificant. This episode was found in the literature by Tariq Alkhalifah et al. (Alkhalifah et al., 2022), which although in a task very different from the proposal were found the same problem.

With this section we achieved the objective **4)**.

References

Alharbi, A., Khan, M. U. G., and Tayyaba, B. Wheat disease classification using continual learning. *IEEE Access*, 11:90016–90026, 2023. doi: 10.1109/ACCESS.2023.3304358.

Alkhalifah, T., Wang, H., and Ovcharenko, O. Mlreal: Bridging the gap between training on synthetic data and real data applications in machine learning. *Artificial Intelligence in Geosciences*, 3:101–114, 2022. ISSN 2666-5441. doi: https://doi.org/10.1016/j.aiig.2022.09.002. URL https://www.sciencedirect.com/science/article/pii/S2666544122000260.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, June 2002. ISSN 1076-9757. doi: 10.1613/jair.953. URL http://dx.doi.org/10.1613/jair.953.

de Amorim, L. B., Cavalcanti, G. D., and Cruz, R. M. The choice of scaling technique matters for classification performance. *Applied Soft Computing*, 133:109924, 2023.

Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. ISSN 1939-3539. doi: 10.1109/tpami.2021.3057446. URL http://dx.doi.org/10.1109/TPAMI.2021.3057446.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

El-Tohamy, A., Maghawry, H., and Badr, N. A deep learning approach for viral dna sequence classification using genetic algorithm. *International Journal of Advanced Computer Science and Applications*, 13, 01 2022. doi: 10.14569/IJACSA.2022.0130861.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Shiraj, T. B. and Yousuf, M. A. A study to classify virus genome through analyzing dna sequences using transformer model. In *2024 6th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, pp. 1275–1280, 2024. doi: 10.1109/ICEEICT62016.2024.10534520.

Tampuu, A., Bzhalava, Z., Dillner, J., and Vicente, R. Viraminer: Deep learning on raw dna sequences for identifying viral genomes in human samples. *PLOS ONE*, 14(9):1–17, 09 2019. doi: 10.1371/journal.pone.0222271. URL https://doi.org/10.1371/journal.pone.0222271.