Paper Discussion Report

Lawrence Owusu, Jordan Sturtz, and Swetha Chittam

I. Summary

A paper authored by Gunasekaran, et al. (2021) uses deep learning methods to perform classification of viruses from their DNA sequences. Since DNA is composed of strings of nucleotides, the problem amounts to classifying viruses according to samples of nucleotide strings. The authors collect data from the public nucleotide sequence database, The National Centre for Biotechnology Information (NCBI) https://www.ncbi.nlm.nih.gov. They then encode this data using label encoding and kmer encoding. For each encoding type, they run three different deep learning models: CNN, CNN-LSTM, and CNN-Bidirectional-LSTM. The architectures of all three models start with embedding layers, then convolutional layers, then max pooling layers, and then from there diverge to either add LSTM layers or bidirectional LSTM layers before finishing with dense layers and a final output layer. The authors compare all six combinations of the two encoding methods with the three model types using several performance metrics.

II. Problem Statement

All DNA and RNA is composed of a string of nucleotides. A nucleotide refers to one of four compounds for DNA (adenine, cytocine, guanine, thymine) or four compounds for RNA (adenine, cytocine, guanine, uracil). For double-helix DNA or RNA, each nucleotide bonds with one and only one other nucleotide, forming what is called a base pair (Fig 1). Since these base pairs are fixed, then, a DNA or RNA sequence can be identified solely by one side of the double helix. Thus, every DNA virus can be identified by a single string of characters drawn from the set $\{A, C, G, T\}$ and every RNA virus can be identified by a string of characters drawn from the set $\{A, C, G, U\}$. The task, then, is to build highly accurate models to classify a virus from its DNA or RNA sample.

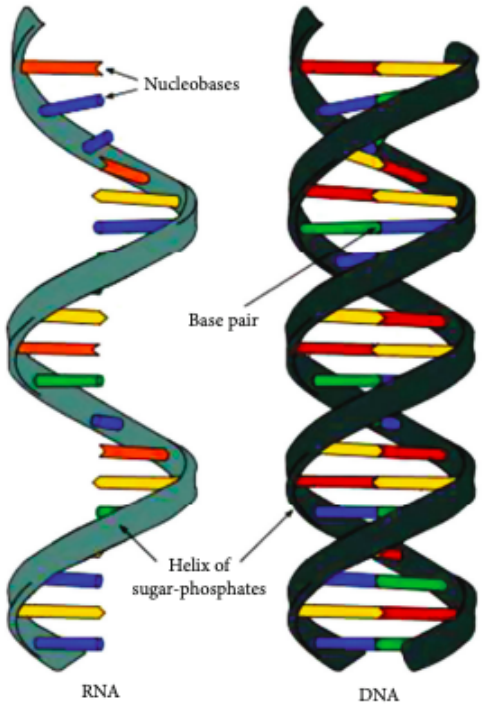

Fig. 1. Single or double-stranded DNA/RNA, borrowed from Gunasekaran, H., et al. (2020)

III. Related Work

A. Current Results on Proposed Problem

IV. Data Collection

The authors obtain complete genomic sequences from the National Centre for Biotechnology Information (NCBI) https://www.ncbi.nlm.nih.gov/. Sequence length ranges from 8 to 37971 nucleoids. They collected genomic sequences for six virus classes: COVID, MERS, SARS, Dengue, Hepatitus, and Influenza (Fig 2). Because the population of these viruses were unbalanced–for instance, there were 37272 samples of COVID and only 1418 samples of MERS–the authors opted to use Synthetic Minority Oversampling Technique (SMOTE) to get a more even distribution of all six classes in their dataset.

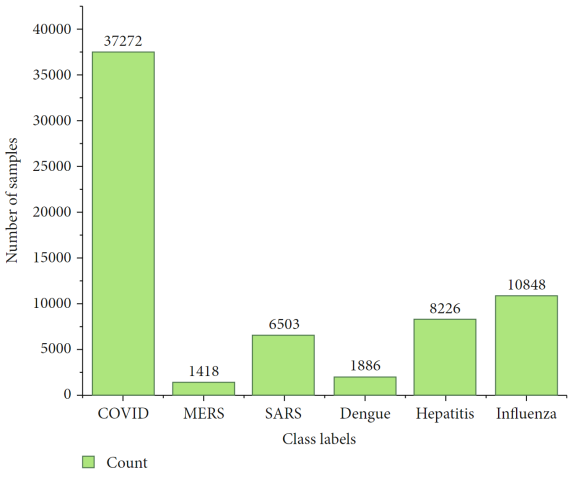

Fig. 2. Samples of virus classes retrieved from NCBI, borrowed from Gunasekaran, H., et al. (2020)

The authors are graduate students at NCA&T

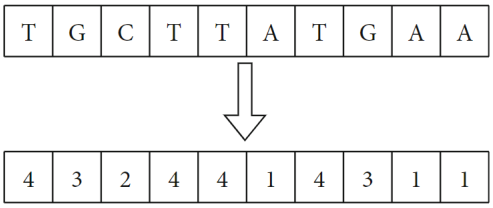

Fig. 3. Label encoding example, borrowed from Gunasekaran, H., et al. (2020)

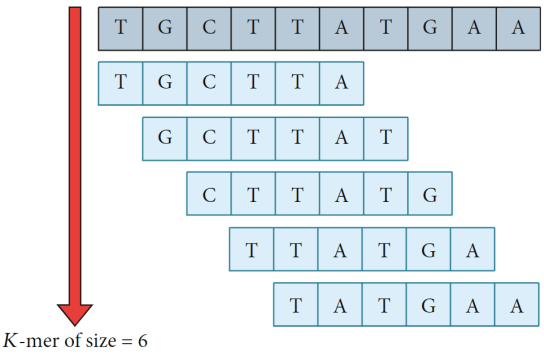

Fig. 4. Kmer encoding example, where k=6

V. Data Preprocessing

The authors encoded the data in two different formats for comparative analysis. In the first approach, they use label encoding, which replaces each nucleoid by a unique index value, preserving positional information (Fig 3). In the second approach, they used kmer encoding, which generates all kmers from a sequence and forms an English-like sentence onto which natural language processing techniques can be applied (Fig 4).

Once encoded, in both cases the input data is one-hot encoded and then fed into the first layer of the models, which is an embedding layer.

VI. Proposed Models

In this paper, the authors use three different classification models to perform multi-classification of the DNA sequence: CNN, CNN-LSTM and CNN-bidirectional LSTM. All three models share the same initial layers: an embedding layer, convolutional layers, and max pooling layers. The raw data is first either label encoded or kmer encoded and then one-hot encoded to be fed into the embedding layer. The results of the embedding layer are then fed into the convolutional layers, each of which has a corresponding max pooling layer. From here, the three model types diverge. For the CNN model, the results are fed into the final dense layers and output layers. The LSTM and bidirectional LSTM models differ only in that they add LSTM or bidirectional LSTM respectively before the final dense layers.

TABLE I
Hyper parameters of CNN model

Layers	Units	Filters	Kernel	Act. Func	Output Shape	Params
Embedding	8				(None, 1000, 8)	128
Conv 1D		128	2*2	ReLu	(None, 1000, 128)	3200
Max Pool			2*2		(None, 500, 128)	0
Conv 1D		64	2*2	ReLu	(None, 500, 64)	24640
Max Pool			2*2		(None, 250, 64)	0
Flatten					(None, 16)	0
Dense	128				(None, 128)	2176
Dense	64				(None, 64)	8256
Dense	6			Softmax	(None, 6)	390

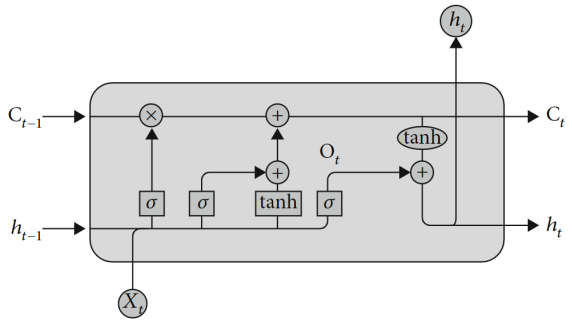

Fig. 5. LSTM Cell Architecture. x is a multiplicative gate, + is additive, borrowed from Gunasekaran, H., et al. (2020)

A. CNN

CNN is a popular deep learning technique that is widely used for feature extraction and to solve any classification problems. CNN can be used for any kind of datasets like text, image, or video datasets. The 2D CNN is often used for feature extraction from image datasets and 3D CNN is used for feature extraction from video datasets. The 1D CNN auto extracts the features from the input dataset and can therefore be used for text classification. Here in this paper, authors are using 1D CNN for feature extraction of the DNA sequence data. The proposed CNN model architecture has two 1D convolutional layers, each followed by a max pooling layer to reduce the feature map dimensions from the previous layers. Next, a flatten layer is used to convert feature maps to a single column vector followed by multiple dense layers. The output is passed to the final dense layer with softmax activation function to perform multi-classification.

Table I shows the hyper parameters for the proposed CNN model

B. CNN-LSTM

Long-short term memory (LSTM) is a recurrent neural network that can learn long term dependencies in a sequence and hence can be used for sequence classification or prediction. The LSTM model consists of three gates: forget gate, input gate and output gate (Fig 5). For the hybrid CNN-LSTM model, the authors add an LSTM Layer with 100 LSTM memory units between the final convolutional layer and the final dense layers.

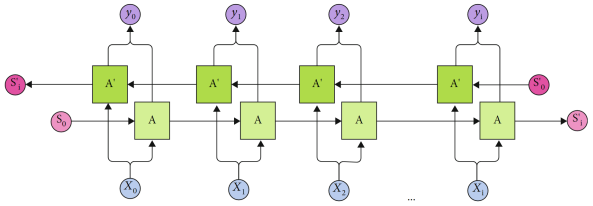

Fig. 6. Bidirectional LSTM Model Architecture, borrowed from Gunasekaran, et al. (2020)

TABLE II
BEST HYPERPARAMETERS

Parameters	Values
Size of Filter	2*2
Training Batch Size	100
Training Epochs	10
Embedding Dimensions	32
K-mer Size	6

C. CNN-Bidirectional-LSTM

The bi-directional LSTM has two RNN's: One to learn dependencies in forward direction and the other to learn dependencies in backward direction (Fig 6). For the hybrid CNN-LSTM-Bidirectional model, the authors add the bidirectional LSTM layers after the final convolutional layers and before the final dense layers.

VII. RESULTS AND DISCUSSION

The DNA sequence dataset used in this paper consists of 66,153 inputs which is divided into training, validation and testing sets with a ratio of 70%, 10% and 20% respectively. The training set consists of 46307, and the validation set consists of 6615, and the testing set consists of 13231 samples. The maximum sequence length is 2000, and the vocabulary size is 8972. The proposed models CNN, CNN-LSTM, CNN-Bi-LSTM models are tested by varying different hyperparameters listed in Table I. The authors use the grid-search cross-validation technique to find the best parameters of the model. The best parameters are listed in Table II.

The classification models are evaluated using different classification metrics like accuracy, precision, recall, F1 score, sensitivity, and specificity by obtaining the confusion matrix for both k-mer and label encoding techniques. The confusion matrix holds the values of true positives (TP), true negatives (TN), false positives (FP), and false negatives(FN). Based on these values, classification metrics are calculated and compared. Table III shows the formulas for calculating these classification metrics.

A. Model Comparison

Figure 7 shows the accuracy comparison for all three models for training and testing data using label and k-mer encoding. The testing accuracies for label encoding are less when compared to its training accuracy. For k-mer encoding testing accuracies are more significant than training. Therefore, the encoding technique plays an important role for achieving high accuracy.

TABLE III
FORMULAS TO CALCULATE PERFORMANCE METRICS

Metric	Formula
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Specificity	$\frac{TN}{TN+FP}$
Sensitivity	$\frac{TP}{TP+FN}$
Precision	$\frac{TP}{TP+FP}$

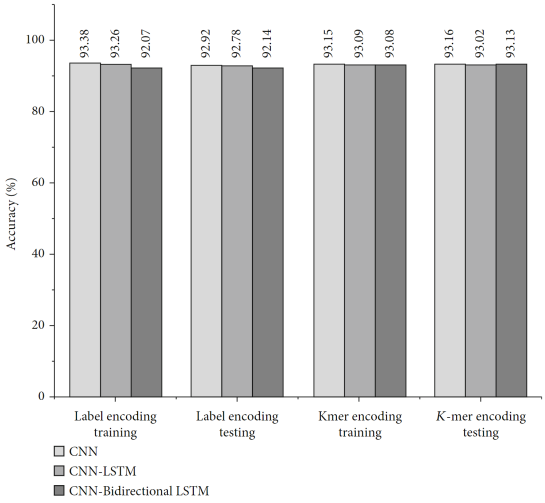

Fig. 7. Bidirectional LSTM Model Architecture, borrowed from Gunasekaran, et al. (2020)

Figure 8 shows the training and validation accuracies for all three models and both encodings. From Figure 8, we can observe that the accuracy of the model remains same after 10 epochs for all models except CNN-LSTM using k-mer encoding. This model has unstable accuracy as each epoch increases. CNN and CNN-Bi-LSTM have high accuracies of 93.16%, 93.13% respectively, when compared to LSTM.

B. Model Performance Against State of the Art Methods

The authors compare their accuracy results to other state-of-the-art approaches to virus classification (Fig 9) Their results show that all three of their models show higher accuracy than that achieved by Nguyen, et al. (2016), Do, et al. (2020), and Zhang, et al. (2020).

VIII. CONCLUSION

REFERENCES

[1] H. Gunasekaran, K. Ramalakshmi, A. Rex Macedo Arokiaraj, S. Deepa Kanmani, C. Venkatesan, and C. Suresh Gnana Dhas. "Analysis of DNA Sequence Classification Using CNN and Hybrid Models." *Computational and mathematical methods in medicine*, 2021.

[2] N. G. Nguyen, V. A. Tran, D. L. Ngo et al., "DNA sequence classification by convolutional neural network," *Journal of Bio- medical Science and Engineering*, vol. 9, no. 5, pp. 280–286, 2016.

[3] D. T. Do and N. Q. K. Le, "Using extreme gradient boosting to identify origin of replication in Saccharomyces cerevisiae via hybrid features," *Genomics*, vol. 112, no. 3, pp. 2445–2451,2020

[4] X. Zhang, B. Beinke, B. Al Kindhi, and M. Wiering, "Comparing machine learning algorithms with or without feature extraction for DNA classification" 2020, http://arxiv.org/abs/2011.00485

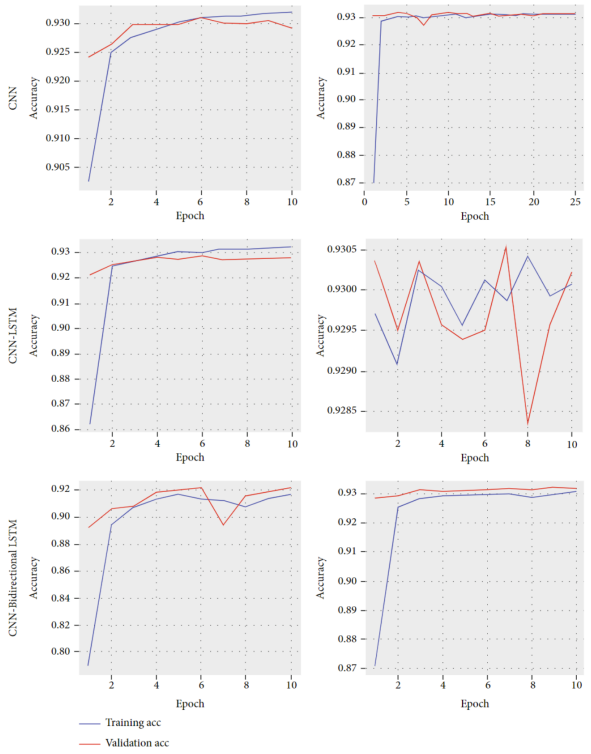

Fig. 8. Training / Validation accuracies for all six models, borrowed from Gunasekaran, et al. (2020)

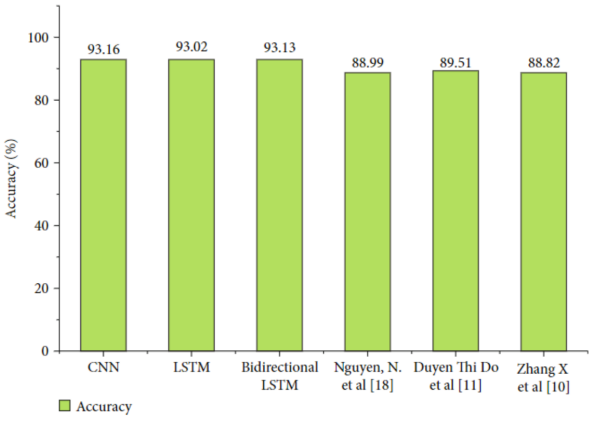

Fig. 9. Gunasekaran, H. et al, compared against state of the art methods, borrowed from Gunasekaran, et al. (2020)