ISSN 2347 - 3983
Volume 9. No. 3, March 2021
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter20932021.pdf
https://doi.org/10.30534/ijeter/2021/20932021

Prediction of Eukaryotic Exons using Bidirectional LSTM-RNN based Deep Learning Model

Noopur Singh[1], Ravindra Nath[2], Dev Bukhsh Singh[3]
[1] Department of Biotechnology, Dr. A. P. J. Abdul Kalam Technical University, Lucknow-226021, India, noopursingh084@gmail.com
[2] Department of Computer Science, University Institute Engineering and Technology, Chhatrapati Shahu Ji Maharaj University, Kanpur-208024, India, rnkatiyar@gmail.com
[3] Department of Biotechnology, Institute of Biosciences and Biotechnology, Chhatrapati Shahu Ji Maharaj University, Kanpur-208024, India, answer.dev@gmail.com

ABSTRACT

Exon prediction has always been a challenge for computational biologist. Although there have been many advances in identification and prediction of exons by computational methods. The efficacy and efficiency of prediction methods need to be further improved using new parameters and algorithms. Moreover, it is essential to develop new prediction methods by combining already existing approaches that can greatly improve prediction accuracy. A eukaryotic gene contains several exons and introns that are separated by splice site junction. It is important to accurately identify splice sites in a gene. Splice sites regions are known, but computational signal prediction is still challenging due to numerous false positives and other problems. In this paper, a novel combination of Support Vector Machine and bidirectional LSTM-RNN based Deep Learning approaches has been applied to improve the efficiency and accuracy of exon prediction. The proposed method takes into account the conventional machine learning as well as the deep learning approach on predictive accuracy of eukaryotic exons.

Key words: Machine Learning, Deep Learning, SVM, Bidirectional LSTM, RNN, Splice Site, Intron, Exon

1. INTRODUCTION

Eukaryotes got a special feature from nature that makes them complex organisms, that is the mechanism of “splicing”. The diverse nature of proteins in eukaryotes are due to the mechanism of splicing only. One of the primary factors supporting protein diversity in eukaryotes is splicing. Through splicing, the stability of mRNA variants is also regulated. An important factor which depends on alternative splicing is the spatial localization of transcripts [1].

In eukaryotes, each gene consists of protein-coding regions (exons) and non-coding region (introns). In the DNA sequence, the intron starts with a donor splice site region GT and ends with an acceptor splice site AG. In order for a gene to be expressed as a protein, the new precursor messenger RNA (pre-MRNA) is modified by splicing the transcript, which removes the introns and joins them to the exons [2].

Precise identification of donor splice site (GT) and acceptor splice site (AG) from a genetic sequence is significant for transcriptome research and the diversity of expressed proteins. For solving this problem, we have applied both Machine Learning and Deep Learning methods. For machine learning, Support Vector Machine (SVM) and in deep learning, Bidirectional Long short-term memory Recurrent Neural Networks (LSTM-RNN) has been used.

2. MATERIALS AND METHODS

Machine Learning (ML) is a branch of computer programming that provides self-learning proficiencies for machines without explicit programming. ML algorithms are used widely in bioinformatics for classification, prediction and feature selection. ML methods are brilliant at solving problems such as differentiating DNA sequences and classifying DNA sequences. Now, ML in Bioinformatics has been developed as a substantial area with the introduction of deep learning [3].

Support Vector Machine (SVM) is a supervised machine learning method with a robust theoretical basis and high classification accuracy for many applications. SVMs can learn precise classifiers for linear inseparable data sets at the input space [4]. This is accomplished by selecting the appropriate kernel function to convert the input data into alternative feature, where it is easy to calculate the exact classification. Through learning the optimal separating hyperplane of this feature space, one can learn a non-linear classifier at the original input space [5].

Bidirectional Long short-term memory Recurrent Neural Networks (**LSTM-RNN)** is mainly a deep learning network model. Deep learning is composed of multiple layered neural networks. Recurrent neural network (RNN) is planned for taking information from sequences or time series data. They can take variable size inputs; variable size outputs and they really work nicely with DNA sequences or time series data. RNN is one such network model that has a combination of networks in loop. The networks in loop permit the information to stay. Each network in the loop takes input and information from previous network and performs the specified operation in turn produces output along with passing the information to next network [6]. Long Short Term Memory (LSTM) Networks of RNN are capable in learning such states. These networks are indeed planned to escape the long term dependency problem of recurrent neural networks [7]. LSTM is in fact adding of little more connections to RNN to rise the accuracy of the model.

2.1 Training Dataset Preparation

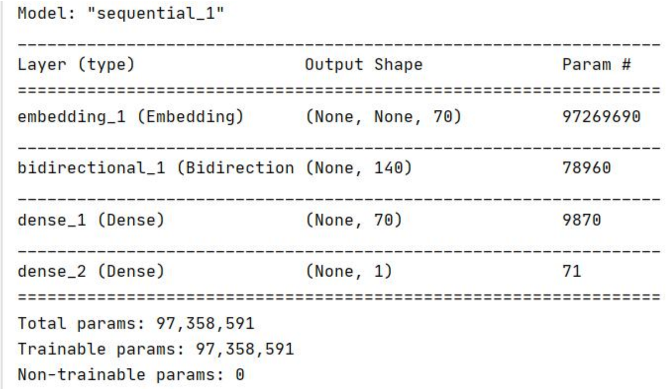
Introns always have two distinct nucleotides at either end. At the 5' end of the intron, the DNA nucleotides are GT [GU in the pre-messenger RNA (pre-mRNA)]; at the 3' end they are AG. These nucleotides are canonical splice site signals that are used in the prediction methods to predict exons and introns. The human genome sequences were downloaded from NCBI (https://www.ncbi.nlm.nih.gov/genome/?term=homo+sapiens) [8].

Then at first, the sense and antisense strand of the DNA sequences were translated into all the three reading frames using a program and the translated sequences are then searched in Pfam database [9] for protein domains. Considerable hits to the protein domains possessing e-value less than 0.01 were mapped against the open reading frames (ORFs), that comprise the primary set of domains supporting DNA sequences. Secondly, a SVM classifier based on composition is trained. All DNA sequences carrying a domain motif possessing e-value less than 10^{-50} used for training inputs for the coding sequence class and ORFs situated in between of these DNA sequences were used for training the non-coding ORF class. For input to the SVM classifier, all the ORFs were signified as vectors of sequence composition features.

A computer script to extract exon and introns from the coding sequences (CDSs) written using SVM. In prediction model, The average length of exon was kept 170 bp and that of intron was kept 5419 bp [10]. The extracted exons and introns were stored in two separate text files. These extracted exons and introns are categorized as the classes of the training dataset.

2.2 Preparing Deep Learning Model

The first step of preparation of the model is loading dataset. The exon and intron sequences were loaded from their text files to separate exon sequence list and intron sequence list respectively. Then kmer (length = 170) of each exon sequence was computed and the resulting kmer stored to exon_texts string variable. Similarly, kmer (length = 5419) of each intron sequence was computed and the resulting kmer stored to intron_texts variable. All the exons in exon_texts were labelled to 1 and all the introns in intron_texts labelled to 0. Then both the string variables, exon_texts and intron_texts were merged using tokenizer to merge_texts. The dataset was splitted into training and testing sample that comprise of 80% and 20% of the data respectively. Second step is compilation of the model. Here bidirectional LSTM-RNN sequential model has been used [11]. In the network layer, embedding with vocab size 97269690 was added. Then bidirectional LSTM with 70 inputs were added. A dense layer with 70 outputs were added and at last, an output layer that is a dense layer with 1 output and activation sigmoid were added. Figure 1 shows the summary of defined bidirectional LSTM-RNN deep learning model. After the model preparation, it was compiled with loss = binary_crossentopy, adam optimizer and accuracy metrics to evaluate the coefficient of the model.



```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 70)          97269690
_________________________________________________________________
bidirectional_1 (Bidirection (None, 140)               78960
_________________________________________________________________
dense_1 (Dense)              (None, 70)                9870
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 71
=================================================================
Total params: 97,358,591
Trainable params: 97,358,591
Non-trainable params: 0
```

Figure 1: Defined Bidirectional LSTM-RNN Deep Learning Model

2.3 Training Deep Learning Model

The prepared and compiled deep learning model was trained by 80% of the dataset with input X, output Y and numbers of 5 epochs.

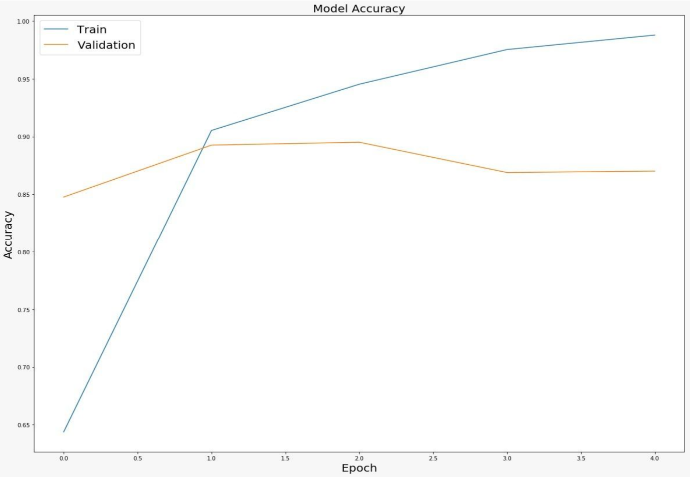


Figure 2: Model Accuracy versus Epoch

Figure 2 shows that the accuracy of the model is increasing with the number of epochs for both training and validation data i.e., the test data.

Figure 3 shows that the loss of the model decreases with the number of epochs for both training and validation data i.e., the test data.

That means, the number of epochs plays a significant role in the training and testing of the model. So, to increase the accuracy of the model it is very necessary to train the model by that much number of epochs to attain the maximum accuracy.

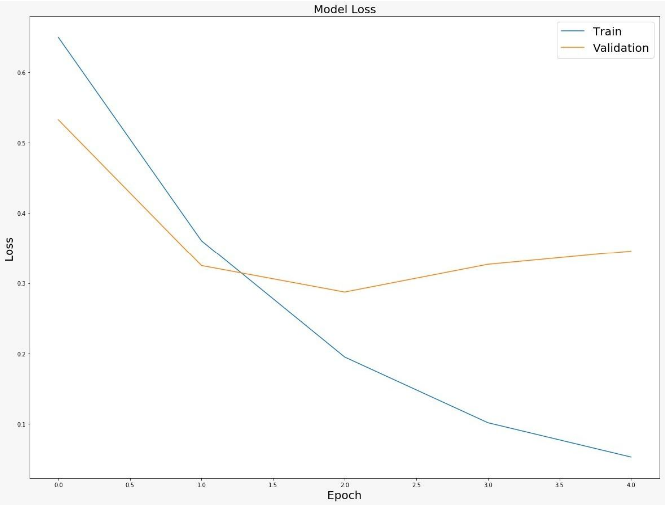


Figure 3: Model Loss versus Epoch

2.4 Testing Deep Learning Model

The deep learning model was tested by the 20% dataset with Test X (input test data), Test Y (output test data) and number of epochs 5

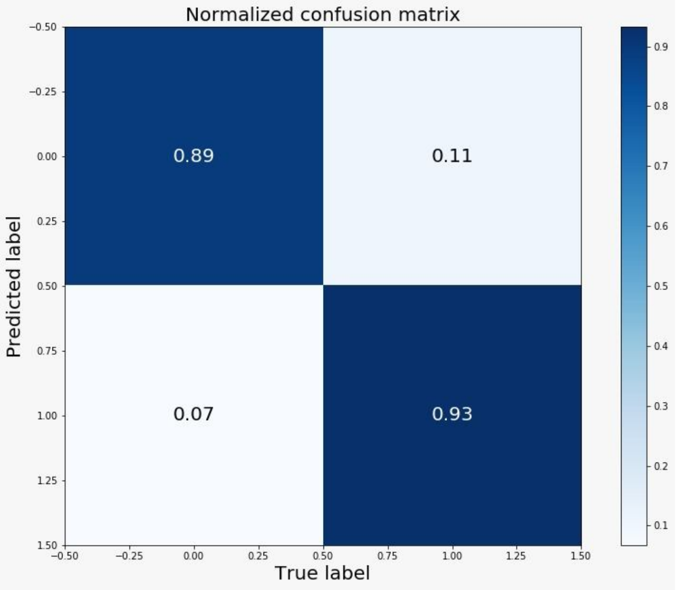


Figure 4: Confusion matrix for Tested Model

In Figure 4, the confusion matrix shows that the 89% of the tested model give true positives (TP) and 11% false positives (FP; 93% gives true negatives (TN) and 7% false negatives. The exon has been predicted and the probabilities of the model training result when compared to model test result for the same sequence the accuracy of the model comes out 96%. That is a great prediction result.

3. RESULT AND DISCUSSION

A total of 114403 CDSs were identified and predicted by the SVM classifier is 98.6% close to the annotated data of human genome CDSs that is 115987 (https://www.ncbi.nlm.nih.gov/genome/annotation_euk/Homo_sapiens/109.20201120/). These predicted CDSs were used to train and test data as mentioned in the methodology section.

Table 1: Prediction result of Exon and Intron by proposed approach and annotated data (A comparative study of predicted vs. annotated)

	Model Prediction	**Annotated data (NCBI)**
Exons	310479	323416
Introns	279188	290821

We selected human genome sequence to train, test and validate our model, as human genome contains a pretty good number of exons as well as introns. A four layered deep learning model was generated, embedding layer, hidden bidirectional LSTM layer, dense layer 1 and dense layer 2. The model was trained by 91552 sequences i.e., 80% of the total 114403 CDSs and was tested by 22881 sequences i.e., 20% of the 114403 CDSs. After training and testing the bidirectional LSTM-RNN based deep learning model, the model was validated using the benchmark data that is the annotated data of coding transcripts for exon and introns at NCBI database [12].

Table 1 shows the results predicted by the proposed model and the actual statistics as per annotated data available in NCBI. The predicted exons and introns by the proposed deep learning model are 310479 and 290821 respectively. The predicted result is about 96% of the annotated data, and is very much close to real data of exon and intron as annotated in NCBI. The number of introns predicted by model are 279188 while intron count as per NCBI data are 290821. In both cases, proposed approach has reached to satisfactory level of prediction. The model used the canonical splice site signals to perform their task. The RNN architecture used was LSTM based. The accuracy of the model can be increased by increasing the number of epochs while training and testing.

Table 2 shows the test accuracy of the developed prediction model over other two methods for comparison that includes test accuracy of Deep Belief Network (DBM) [13] and Unidirectional LSTM [14]. Evidently, our proposed approach proved to be better than both the DBM and unidirectional LSTM methods.

Table 2: Test accuracy of DBM, Unidirectional LSTM, and proposed bidirectional LSTM-RNN approach

Model Type	**Accuracy**
Deep Belief Network (DBM) [13]	89%
Unidirectional LSTM [14]	82%
Bidirectional LSTM-RNN (proposed approach)	96%

4. CONCLUSION

In this paper, we have described an approach to identify and predict the exons and introns based on splice site junction signals of the predicted CDSs. The proposed model outperformed the existing alternatives in terms of accuracy. By keeping in mind, the accuracy level of our developed approach, we expect that it will be of great help in overcoming the limitations and challenges in the computational prediction of exons for eukaryotic DNA sequences. Further, this developed model can be used to train several other eukaryotic genomes for exon prediction. So that it can be proposed as a universal approach, or model for exon prediction of all eukaryotic genomes.

CONFLICT OF INTEREST

The authors declare no conflict of interest, financial or otherwise.

ACKNOWLEDGEMENT

Declared none.

REFERENCES

[1] Y. Li, H. Li-Byarlay, P. Burns, M. Borodovsky, G. E. Robinson, and J. Ma, **“TrueSight: A new algorithm for splice junction detection using RNA-seq,”** *Nucleic Acids Res.*, 2013, doi: 10.1093/nar/gks1311.

[2] M. Burset, I. A. Seledtsov, and V. V. Solovyev, **“Analysis of canonical and non-canonical splice sites in mammalian genomes,”** *Nucleic Acids Res.*, 2000, doi: 10.1093/nar/28.21.4364.

[3] K. G. Srinivasan, G. M. Siddesh, and S. R. Manisekhar, ***Statistical Modelling and Machine Learning Principles for Bioinformatics Techniques, Tools, and Applications*.** 2020.

[4] S. R. Sain and V. N. Vapnik, **“The Nature of Statistical Learning Theory,”** *Technometrics*, 1996, doi: 10.2307/1271324.

[5] T. T. Hastie, **“The Elements of Statistical Learning Second Edition,”** *Math. Intell.*, 2017.

[6] J. Ostmeyer and L. Cowell, **“Machine learning on sequential data using a recurrent weighted average,”** *Neurocomputing*, vol. 331, pp. 281–288, 2019.

[7] J. Kumar, R. Goomer, and A. K. Singh, **“Long Short Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model for Cloud Datacenters,”** 2018, doi: 10.1016/j.procs.2017.12.087.

[8] W. D. Law, R. L. Warren, and A. S. McCallion, **“Establishment of an eHAP1 human haploid cell line hybrid reference genome assembled from short and long reads,”** *Genomics*, 2020, doi: 10.1016/j.ygeno.2020.01.009.

[9] R. D. Finn *et al.*, **“Pfam: The protein families database,”** *Nucleic Acids Research*. 2014, doi: 10.1093/nar/gkt1223.

[10] M. K. Sakharkar, V. T. K. Chow, and P. Kangueane, **“Distributions of exons and introns in the human genome,”** *In Silico Biol.*, 2004.

[11] S. Hochreiter and J. Schmidhuber, **“Long Short-Term Memory,”** *Neural Comput.*, 1997, doi: 10.1162/neco.1997.9.8.1735.

[12] D. L. Wheeler *et al.*, **“Database resources of the National Center for Biotechnology Information,”** *Nucleic Acids Res.*, 2008, doi: 10.1093/nar/gkm1000.

[13] T. Lee and S. Yoon, **“Boosted Categorical Restricted Boltzmann Machine for Computational Prediction of Splice Junctions,”** 2015, vol. 37, pp. 2483–2492, [Online]. Available: http://proceedings.mlr.press/v37/leeb15.html.

[14] L. A. Uroshlev, N. V Bal, and E. A. Chesnokova, **“A Long Short-Term Memory Neural Network Used to Predict the Exon–Intron Structure of a Gene,”** *Biophysics (Oxf).*, vol. 65, no. 4, pp. 574–576, 2020, doi: 10.1134/S0006350920040259.