# Analysis of Prostate Cancer DNA Sequences Using Bi-direction Long Short Term Memory Model.

Yusuf Aleshinloye Abass[1], Steve A. Adeshina[2], Nwojo Nnana Agwu[3], Moussa Mahamat Boukar[4]
Department of Computer Science, Nile University of Nigeria[1,2,3,4]
yusuf.abass@nileuniversity.edu.ng[1],steve.adeshina@nileuniversity.edu.ng[2]
nagwu@nileuniversity.edu.ng[3], musa.muhammad@nileuniversity.edu.ng[4]

2021 16th International Conference on Electronics Computer and Computation (ICECCO) | 978-1-6654-0945-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICECCO53203.2021.9663839

***Abstract*— Machine and deep learning-based models are the emerging techniques in addressing prediction problems in biomedical data analysis. DNA sequence prediction is a critical problem that requires huge attention in the biomedical domain. These techniques have been shown to provide better accurate results when compared to the conventional regression-based models. Prediction of the gene sequence that leads to cancerous diseases such as prostate cancer is very crucial. Identifying the most important features in a gene sequence is one of the most challenging tasks and extracting the components of the gene sequence that can give an insight into the kind of mutation in the gene is very important, it will lead to effective drug design and promote the new concept of personalized medicine. In this work we have extracted the exons in the various prostate gene sequence that was used in the experiment, we built a bi-LSTM model using a k-mer encoding for the DNA sequence and one-hot encoding for the class label. The bi-LSTM model was evaluated on different classification metrics. Our experimental results show that the model prediction offers a training accuracy and validation accuracy of 95 percent and 91 percent respectively.**

**Keywords—Deep learning, DNA sequence, k-mer, Prediction, Bi-LSTM**

## I. INTRODUCTION

Prostate cancer is one of the most common cancers in men and its global incidence is rising [1]. Diagnosis results have shown that more than 80% of the men that are diagnosed have a non-metastatic disease. Prostate cancer is the world's third most commonly diagnosed types of cancer after lung and breast cancer and the fifth cancer-specific death in men [2]. Studies have shown that around 1,106,349 patients will be diagnosed with prostate cancer by 2023 in the nine major markets (9MM) (United States, France, Germany, Italy, Spain, United Kingdom, Japan, Brazil, and Canada) [3]. The study includes a 10 year epidemiological forecast for the diagnosed incident cases of prostate cancer segmented by age (from age 40 to age=85) in these markets. Prostate cancer research has focused on Prognosis, Diagnosis, and Prediction of prostate cancer outcomes through the use of statistics and Artificial Intelligence (AI). Computer-Based learning models have become a predominant area of research in prostate cancer in recent times.

The deep neural models mainly include Convolutional Neural Network (CNN), Recurrent Neural Network (RNN and its variants), and Stacked Auto-Encoders (SAE). CNN is known to have a convolutional filter, followed by a non-linearity, a sub-sampling, and a fully connected layer that recognizes the final classification. The early adoption of CNN in genomics was in the area of Computer Vision [4]. The adaptation of CNN from computer Vision to genomics was made possible by assimilating a window of genomic sequence as an image. The ability to use CNN for sequence analysis was first demonstrated in the case of genomic text in [5]. The redesigning of RNN to exploit sequential information of input data with the cyclic connection among building blocks like perceptron's or Long Short-Term Memory unit (LSTM). RNN architecture is known for handling sequence information over time, this attribute is very important in sequence prediction tasks [6].

In this work, we explore the predictive capabilities of a variant of Recurrent Neural Network (RNN) known as the bi-directional Long Short Term Memory (Bi-LSTM) in deep learning architecture for prostate cancer gene sequence using a publicly available dataset. We use the bi-directional Long Short Term Memory (LSTM) as the choice of LSTM because research has shown that the bi-LSTM outperforms the LSTM model in model prediction. The LSTM is a type of recurrent neural network with a more complex computational unit that leads to better performance.

## II. DEEP LEARNING MODELS

### A. Recurrent Neural Network

The Recurrent Neural Networks (RNN) is used for processing sequences data that evolves along the time axis. In a simpler version of RNN, the internal state $h_t$ represent the sequence seen until the previous time step (t – 1) and is used alongside the new input $x_t$.

$$h_t = \sigma(w_h x_t + u_h h_{t-1} + b_h) \quad (1)$$

$$y_t = \sigma(w_y h_t + b_y) \quad (2)$$

Where $w_h$ and $u_h$ are respectively the weight matrices for the input and the internal state, $w_y$ is

978-1-6654-0945-2/21/$31.00 ©2021 IEEE

the weight matrix for producing the output from the internal state, and the two b are bias vectors.

RNN is known to have some drawbacks. The limitation of RNN in the above formulation is that the entire time steps have the same weight and the input contribution in the hidden state is subjected to exponential decay. A variant of RNN was introduced in [7] with the name of Long Short-Term Memory (LSTM).

### *B. Long Short Term Memory*

There are several variants of RNN that were developed to address the drawback of the simple RNN. LSTM is a popular variant of RNN. Every unit of LSTM is associated with memory that is typically referred to as a cell. In the LSTM cell unit, three gates are used in regulating the memory. The gates are input gate ($i_t$), output gate ($o_t$), hidden state ($h_t$) and the forget gate ($f_t$). These gates help the LSTM in determining the information that needs to be added to the current call state ($c_t$) and the information that needs to be forgotten to update the cell state. The equations 3 to 8 below represent the flow from the current-cell state, previous cell state, and the next state [8].

$$f_t = \sigma\left(w_f * [h_{t-1,}x_t] + b_f\right) \quad (3)$$

$$i_t = \sigma\left(w_i * [h_{t-1,}x_t] + b_i\right) \quad (4)$$

$$\tilde{c}_t = \tan h\,(w_c * [h_{t-1,}x_t] + b_c \quad (5)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{c}_t \quad (6)$$

$$o_t = \sigma\left(w_o * [h_{t-1,}x_t] + b_o\right) \quad (7)$$

$$h_t = o_t * tan\, h(c_t) \quad (8)$$

The LSTM was specifically designed to overcome the challenges of vanishing gradient problems and is deemed efficient in capturing long-term dependencies [8].

### *C. Bi-directional LSTM*

The bi-directional LSTM captures the idea that the output of recurrent units at a time step not only depends on its past instances (past elements of the sequence) but also the future instances. The idea of such a network is developed by stacking 2 layers of LSTMs over each other, thus making the output dependent on the computation of the hidden states from both the LSTM layers as opposed to one as in the unidirectional LSTM network [9].

### *D. Softmax Layer*

The RNN and LSTM need a further layer to compute a prediction task. The softmax layer [10] is composed of k units, where k is the number of different classes. Each unit is densely connected with the previous layer and computes the probability that an element is of class k utilizing the formula.

$$softmax_k(x) = \frac{e^{w_k\, x+b_x}}{\sum_{l=1}^{k} e^{w_l\, x+b_l}} \quad (9)$$

Where $w_l$ is the weight matrix connecting the l-th unit to the previous later, x is the output of the previous layer and $b_l$ is the bias for the l-th unit.

Softmax is widely used in deep learning as a prediction layer because of the normalized probability distribution of its outputs, which proves particularly useful during backpropagation.

### *E. Character Embedding*

In this work, we evaluate the bi-LSTM model for predicting genomic sequences without providing prior information utilizing feature engineering. One method of achieving this is represented by the use of k-mer distribution. K-mer is unique subsequences of a particular length k from larger DNA sequences. K-mer representation of biological sequence is a feature discriminant between coding and non-coding region [11] and another form of encoding is the one-hot encoding [12].

## III. Materials and Methods

### *A. Data Collection*

The DNA/Genomic sequence of prostate cancer was obtained from the public nucleotide sequence database: "The National Centre for Biotechnology Information (NCBI)"[1]. The format of the DNA sequence data is the FASTA file. With the huge amount of human genome sequence now publicly available [13], researchers are mining these data to detect genetic variation with the hope of better understanding human diseases. Most genetic variations are in the form of Single-Nucleotide Polymorphisms (SNPs) and insertion/deletions of these non-synonymous SNPs are believed to be most frequently associated with disease phenotypes [14] as they may contribute pathological amino-acid substitutions or nonsense mutations in the protein product. In this research, the gene sequence that was obtained from the NCBI is the Ataxia-Telangiectasia Mutated (ATM), Fanconi Anemia Complementation Group

[1] www.ncbi.nlm.nih.goov

A (FANCA), Breast Cancer Gene1 and 2 (BRCA1 & BRCA2), Epithelial Cell Adhesion Molecule (EPCAM) and MutL Homolog1(MLH1).

### *B. Identification of Exonic Region*

DNA is the most important chemical compound in living cells, viruses, and bacteria. DNA is composed of various types of nucleotides namely Adenine (A), Cytosine (C), Guanine (G), and Thymine (T) [15]. The coding information for protein synthesis is only carried by a specific area of the DNA molecule called the gene [16]. The DNA is divided into gene and inter-genic species in the Eukaryotic cells. The gene is further divided into exons and introns, as shown in Figure 1.

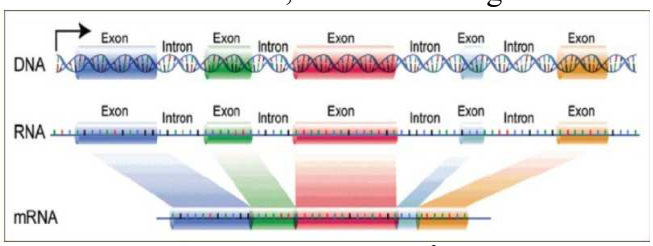


*Figure 1: Exon/Intron region for Eukaryotic*[2]

In recent times, there have been many digital signal processing (DSP) methods presented in the literature to identify the protein-coding regions, and most importantly, these methods are aimed at reducing background noise in the DNA sequences. In [17], a Fourier technique was used to analyze the three-base periodicity in genomic sequences and was able to determine exon regions based on the calculated power spectrum. In [18] a central frequency of $2\delta/3$ was used in removing the background noise from the DNA. In their work, a DNA sequence was passed through a notch filter and sliding window for Discrete Fourier Transform (DFT). In this work, the candidate Exons from the gene sequence mentioned in section 3.1 are extracted from the DNA sequence as shown in Figure 2 below:

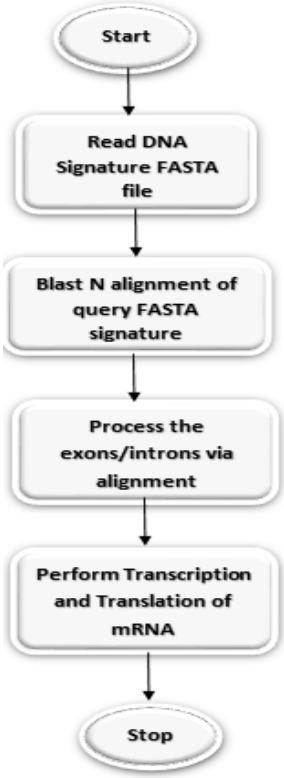


*Figure 2: Extraction of exons from FASTA file.*

The process of sequencing individual genes is usually performed at the exon level. First, the FASTA file containing the DNA sequence is read based on the gene of interest and this is obtained from the sequence database as discussed in the above section. Next, the corresponding genomic sequence is identified and retrieved. The exon/intron structure is determined once the genomic and mRNA sequence are obtained. The section of the mRNA is transcripted, hence the section of the mRNA that does not code for proteins is removed and the section that codes for protein is combined to a long chain of mRNA. Finally, the long chain of mRNA is translated.

The exonic dataset is then labeled based on genetic type. Figure 3 shows a sample dataset with a genomic sequence and class label.

| | Sequence | Type | Label |
|---|---|---|---|
| 0 | CCGGAGCCCGAGCCGAAGGGCGAGCCGCAAACGCTAAGTCGCTGGC... | ATM_201 | HIGH |
| 1 | ACAGTGATGTGTGTTCTGAAATTGTGAACCATGAGTCTAGTACTTA... | ATM_201 | HIGH |
| 2 | AAAGAAGTTGAGAAATTTAAGCGCCTGATTCGAGATCCTGAAACAA... | ATM_201 | HIGH |
| 3 | ATTTTTACAGAAATATATTCAGAAAGAAACAGAATGTCTGAGAATA... | ATM_201 | HIGH |
| 4 | GAGCACCTAGGCTAAAATGTCAAGAACTCTTAAATTATATCATGGA... | ATM_201 | HIGH |

*Figure 3: Sample dataset with genomic sequences, types, and labels.*

### *C. Data Preprocessing*

The task of preprocessing data is one of the most critical steps in most machine learning and deep

[2] https://pubmed.ncbi.nlm.nih.gov/24672762/

learning algorithm that involves numerical data rather than categorical data. There are many techniques available in converting categorical data to numerical. An encoding technique is a process of converting the categorical data of the nucleotide into numerical form. In this paper, label encoding and k-mer encoding are used to encode the DNA sequence. Each input DNA sequence label was converted into a matrix n · l by one-hot encoding, where n corresponds to the three labels High, Low, and Normal represented by binary vectors High = [1,0,0,0], Low = [0,1,0,0], and Normal= [0,0,0,1], respectively, and $l$ equals 4 which is the length of the k-mer. Figure 4 shows all k-mers generated and their concatenation to form a sentence.

| | Sequence | Type | Label | words |
|---|---|---|---|---|
| 0 | CCGGAGCCCGAGCCGAAGGGCGAGCCGCAAACGCTAAGTCGCTGGC... | ATM_201 | HIGH | [ccgg, cgga, ggag, gagc, agcc, gccc, cccg, ccg... |
| 1 | ACAGTGATGTGTGTTCTGAAATTGTGAACCATGAGTCTAGTACTTA... | ATM_201 | HIGH | [acag, cagt, agtg, gtga, tgat, gatg, atgt, tgt... |
| 2 | AAAGAAGTTGAGAAATTTAAGCGCCTGATTCGAGATCCTGAAACAA... | ATM_201 | HIGH | [aaag, aaga, agaa, gaag, aagt, agtt, gttg, ttg... |
| 3 | ATTTTTACAGAAATATATTCAGAAAGAAACAGAATGTCTGAGAATA... | ATM_201 | HIGH | [attt, tttt, tttt, ttta, ttac, taca, acag, cag... |
| 4 | GAGCACCTAGGCTAAAATGTCAAGAACTCTTAAATTATATCATGGA... | ATM_201 | HIGH | [gagc, agca, gcac, cacc, acct, ccta, ctag, tag... |

*Figure 4:K-mer encoding of DNA sequence*

## IV. Model Prediction

Given an exonic sequence of length $L_0$ , we split the sequence into k-mer of size k and extracted all subsequences of length k resulting in a k-mer sequence with length $L = (L_0 - k) + 1 + 1$ (9) In this work, we have used the bi-LSTM as the prostate cancer prediction model. The task of preserving the nucleotide information of the DNA sequence is based on the use of label encoding and k-mer techniques. Figure 5 shows the workflow for the experimentation using the k-mer to represent the features.

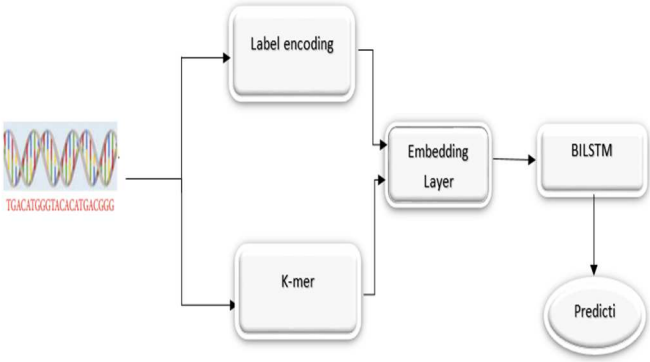


*Figure 5: Prostate cancer DNA sequence prediction model.*

In this work, the bi-directional LSTM which is a variant of the LSTM model is used for DNA sequence prediction. Table 1 shows the summary of the defined bi-LSTM deep learning model. After the model preparation, it was compiled with loss=categorical_crossentropy, Adam optimizer, and accuracy metrics to evaluate the prediction of the model. The architecture of the bi-directional LSTM is given in figure 7.

*Table 1: Hyperparameters used in the Bi-LSTM prediction model*

| Hyperparameters | Values |
|---|---|
| Epochs | 20 |
| Batch size | 32 |
| Architecture Function | Softmax |
| Training size | 80% |
| Validation | 20% |
| Learning rate | 0.0001 |

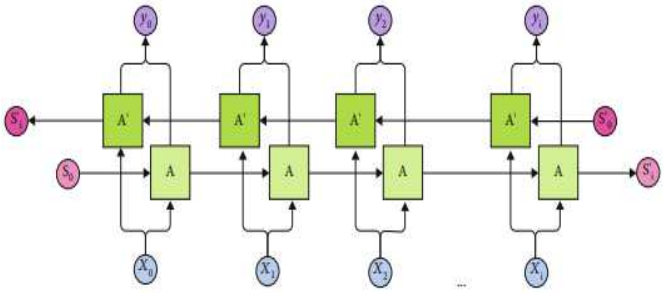

*Figure 6: Architecture of Bi-directional LSTM*

### A. Model Training

The input sequence as shown in Figure 4 is given as input to the sequence model. Each cell takes information as input from the previous cell, in the form of a hidden state vector and cell state vector, and combines it with the one-hot encoded vector. The output is the concatenation of the hidden and cell-state vectors. A softmax function is applied to obtain probability distribution. To reduce overfitting, early stopping is used to end training when the validation loss does not decrease.

## V. Experimental Result

The bi-LSTM model experimented with a system that has a configuration of 8000 MB of RAM. The dataset consists of 13,439 inputs and was divided into a training and testing set with a ratio of 80% and 20% respectively. The training set consists of 10,751 and the testing set consists of 2,687 samples. The size of each k-mer was four and the vocabulary size for each input was 256. In the training phase, the categorical cross-entropy function is used as a loss function. The goal of the loss function is to calculate the error between the

actual output and the target label, for which the training and update of the weight are done. The implemented model was all tested by varying the hyper-parameters like the number of the epoch, number of layers, and embedding dimensions. The classification models were all evaluated using different classification metrics like accuracy, precision, recall, and F1 score. The classification metrics were all calculated from the confusion metric by obtaining the True Positive Gene (TPGene), True Negative Gene (TNGene), False Positive Gene (FPGene), and False Negative Gene (FNGene). The formulae for the stated matrics are given below, Figure 8 shows the confusion matrix and the breakdown of the results obtained for the trained model.

$$Accuracy = \frac{\text{TPGene} + \text{TNGene}}{TPGene + TNGene + FPGene + FNGene}$$

$$Specificity = \frac{\text{TNGene}}{TNGene + FPGene}$$

$$Sencitivity = \frac{\text{TNGene}}{TPGene + FNGene}$$

$$Precision = \frac{\text{TPGene}}{TPGene + FPGene}$$

The computed accuracy for both the training and validation sets are 95 percent and 91 percent respectively.

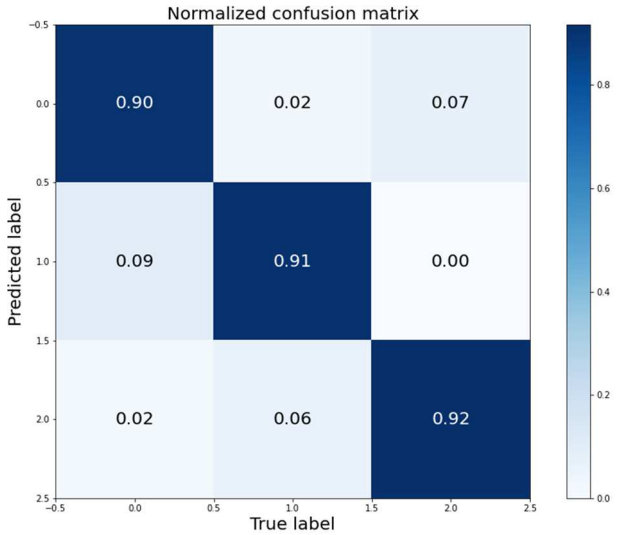


*Figure 8: Normalized Confusion Matrix for the model.*

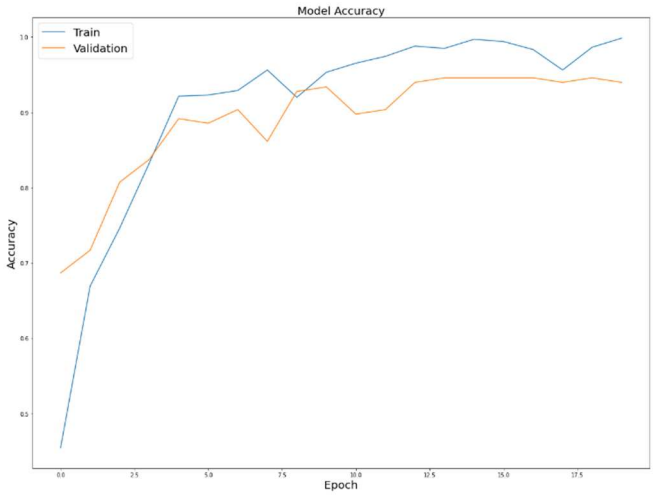


*Figure 9: Training and Validation accuracy.*

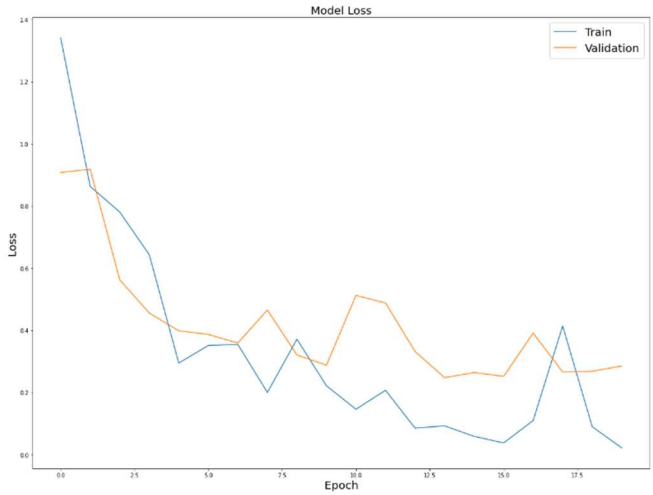


*Figure 10: Training and Validation loss.*

## VI. Discussion

The importance given to each of the stated metric evaluations differs across domains where they are applicable. In data science, a data scientist will look at precision and recall to evaluate the built model. In the field of medicine, specificity and sensitivity are used to evaluate the medical test. In real-life applications, both are very similar but different. Given the sets of positive sequences, sensitivity is a set of sequence that is defined by the positive model. The average of recall and precision values is the F1-score. Precision is the percentage of the positive words (k-mers) identified by the model in the sequence and the specificity values are derived from how well the model determines the negative cases in the sequence. The choice of bi-LSTM is based on the fact that it has been reported in [19] that the use of bi-LSTM outperforms the LSTM model. The overall training and test sets of the model have an accuracy of 95 percent in table 1 and 91 percent respectively. Figures 9 and 10 also show

the training and validation accuracy and loss based on the number of epoch used in the experiment.

## VII. Conclusion

In this work, we described some of the notable deep learning models. The dataset used was from The National Centre for Biotechnology Information (NCBI). The dataset was preprocessed by way of extracting the exons which are the proteins that coded with information. Both the one-hot encoding and k-mer encoding were used on the label and sequences respectively. We train and test the dataset using a bi-LSTM based on the fact that was reported in the discussion section. The implemented model has a high accuracy of 95 percent, another matrix such as precision, recall, and F1-score were also considered. The model's recall values are high in both the training and test sets, indicating that the model is very sensitive in recognizing the k-mers in both sets. We intend to combine multiple models for the same purpose of predicting prostate cancer DNA sequences in the future.

## Acknowledgment

The authors thank the National Information Technology Development Agency (NITDA) and Nile University of Nigeria (NUN) for supporting Y.A. Abass's studies in Nigeria. The authors thank the anonymous reviewers for their objective remarks and their suggestions on the paper.

## References

[1] H. E. Taitt, "Global trends and prostate cancer: a review of incidence, detection, and mortality as influenced by race, ethnicity, and geographic location," *American journal of men's health,* vol. 12, p. 1807–1823, 2018.

[2] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre and A. Jemal, "Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: a cancer journal for clinicians,* vol. 68, p. 394–424, 2018.

[3] K. Gohil, "Exciting therapies ahead in prostate cancer," *Pharmacy and Therapeutics,* vol. 40, p. 530, 2015.

[4] Y. A. Abass and S. A. Adeshina, "Deep Learning Methodologies for Genomic Data Prediction," *Journal of Artificial Intelligence for Medical Sciences,* 2021.

[5] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM,* vol. 60, p. 84–90, 2017.

[6] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of machine learning research,* vol. 12, p. 2493–2537, 2011.

[7] I. Goodfellow, Y. Bengio and A. Courville, Deep learning, MIT press, 2016.

[8] C. Olah, "Understanding lstm networks," 2015.

[9] H. P. Desai, A. P. Parameshwaran, R. Sunderraman and M. Weeks, "Comparative study using neural networks for 16S ribosomal gene classification," *Journal of Computational Biology,* vol. 27, p. 248–258, 2020.

[10] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing*, Springer, 1990, p. 227–236.

[11] M. Axelson-Fisk, "Comparative Gene Finding," in *Comparative Gene Finding*, Springer, 2010, p. 157–180.

[12] L. Fu, Q. Peng and L. Chai, "Predicting dna methylation states with hybrid information based deep-learning model," *IEEE/ACM transactions on computational biology and bioinformatics,* vol. 17, p. 1721–1728, 2019.

[13] E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh and others, "Initial sequencing and analysis of the human genome," 2001.

[14] S. Sunyaev, J. Hanke, A. Aydin, U. Wirkner, I. Zastrow, J. Reich and P. Bork, "Prediction of nonsynonymous single nucleotide polymorphisms in human disease-associated genes," *Journal of molecular medicine,* vol. 77, p. 754–760, 1999.

[15] D. Snustad and M. Simmons, *Principles of Genetics 2nd Edition John Wiley & Sons,* Inc, 2000.

[16] E. R. Dougherty and I. Shmulevich, Genomic signal processing and statistics, vol. 2, Hindawi Publishing Corporation, 2005.

[17] S. Tiwari, S. Ramachandran, A. Bhattacharya, S. Bhattacharya and R. Ramaswamy, "Prediction of probable genes by Fourier analysis of genomic sequences," *Bioinformatics,* vol. 13, p. 263–270, 1997.

[18] H. Saberkari, M. Shamsi, M. Sedaaghi and F. Golabi, "Prediction of protein coding regions in DNA sequences using signal processing methods," in *2012 IEEE Symposium on Industrial Electronics and Applications*, 2012.

[19] S. Siami-Namini, N. Tavakoli and A. S. Namin, "A comparative analysis of forecasting financial time series using arima, lstm, and bilstm," *arXiv preprint arXiv:1911.09512,* 2019.