DeepBiG: A Hybrid Supervised CNN and Bidirectional GRU Model for Predicting the DNA Sequence

Chai Wen Chuah*[1], Wanxian He[2], De-Shuang Huang[3]
Guangdong University of Science and Technology, Dongguang, Guangzhou, China[1]
Guangxi Key Lab of Human-machine Interaction and Intelligent Decision
Guangxi Academy of Sciences, Nanning, Guangxi, China[2]
Eastern Institute for Advanced Study, Eastern Institute of Technology, Ningbo, Zhejiang, China[3]

***Abstract*—Understanding the deoxyribonucleic acid (DNA) sequence is a major component of bioinformatics research. The amount of biological data increases tremendously. Hence, there is a need for effective approaches to handle the critical problem in the general computational framework of DNA sequence prediction and classification. Numerous deep learning languages can be used to complete these tasks compared to manual techniques that have been followed for ages. The aim of this project is to employ effective approaches for pre-processing DNA sequences and using deep learning languages to train the sequences for making judgments, predictions, and classifications of DNA sequences into known categories. In this study, the pre-processing methods include k-mers and tokenization. We employ a novel hybrid deep learning algorithm that combines convolutional neural networks and is followed by bidirectional gated recurrent networks. This combination can capture dependencies within the genome sequence, even in large datasets with a lot of noise. The proposed model is compared with existing widely used models and classifiers. The results show that the proposed model achieves a good result with an accuracy of 82.90%. The dataset consists of 44,391 labeled DNA sequences obtained from the Encode project.**

Keywords*—DNA sequencing; deep learning; convolutional neural networks; bidirectional gated recurrent; k-mer; tokenizing

I. Introduction

Deoxyribonucleic acid (DNA) is unique. It contains a list of genetic codes which look likes no order random letters of adenine (A), cytosine (C), guanine (G), and thymine (T). Eventually, it is organised into little chunks that carry a set of meaning instructions for how to build and maintain body. The little chunks in known as genes. Most genes are alike to each other, only a small number of genes are slightly different between people, that resulted the uniqueness physical features. Genes instruct cells how to make proteins. As we need protein to repair cells and make a new ones for growth and maintenance of tissues. Our body proteins suppose is a constant state of turnover. Nevertheless, errors happen during the journey from genes to protein, it can develop into unhealthy genes and cause cells abnormal in growing.

To discovery these abnormal ChIP-seq is relying on experimental analysis the structures of DNA binding sequences [3], [4], [5]. These experimental analysis usually is time consuming [6], [7]. With quick expansion in the amount of genomic DNA, there is a need for efficient methods in predicting ChIP-seq allows the binding sites of transcription factors (TF). Hence, deep learning and machine learning are widely applied in predicting the DNA sequence binding specificities.

Deep learning techniques have accomplished exceptional outcomes in computer vision [8], [9], natural language processing [10], [11], bioinformatics [12], [13] and image analysis [14], [15]. Methods based on convolutional neural networks (CNN) [16] and recurrent neural networks (RNN) [17], [18] like gated recurrent unit (GRU), long short-term memory networks (LSTM) have been proposed to analyse and predict genome DNA. These techniques have been improved to generate autonomous prediction at learning process that spot specific trends and patterns to make better decisions based on the given data.

DeepBind [19] is pioneer CNN with single convolution layer, pooling operation and fully connected network. The design demonstrates a promising result to predict the sequence specification of DNA and ribonucleic acid (RNA) binding. This has inspired the following research like DeepSHR [20], DeepSEA [21] and Dilated [22]. KEGRU [23] uses a bidirectional gated recurrent (BiGRU) unit with k-mer sequences to find RNA protein binding sites. This method allows mining long dependencies of the sequences and thus achieves good performance in binding sites. DanQ [24] is a hybrid CNN + bidirectional LSTM (BiLSTM) model that applies the capabilities of CNN in extracting DNA features and BiLSTM in handling long range dependencies in order to obtain good performance.

Despite all these studies, there is a gap in finding a fair comparison which deep learning architectures perform well in detecting DNA sequences. As some methods use one-hot to code the DNA sequences, some use k-mer. One-hot is mutual orthoganal, it ignores the DNA sequence dependencies. k-mer overcomes the issue of one-hot by adjoining DNA sequences. Hence, this paper considering k-mer for data pre-processing in finding the dependency between the genome patterns and possibility independence for the underlying genome. Next, tokenize the k sequences to prepare a bag of vocabulary for deep learning process. This research, we aim to propose hybrid deep learning with CNN and BiGRU (DeepBiG) to classify Chromatin Immunoprecipitation Sequencing (ChIP-seq) data from lymphoblastoid cells (GM12878) and K562 chronic myelogenous leukemia (CML) obtained from the Encyclopedia of DNA Elements (ENCODE). CNN consists of extraction and representation capabilities. BiGRU allows capture long-range dependencies and thus obtains good performance. Next, DeepBiG is compared with different deep learning architectures with the same parameters and same dataset. Noted that, the ability in identifying the specific ChIP-seq can significantly improve our understanding on the epigenetic mechanism of the disease, thus promoting precision in drug discovery [1], [2].

The rest of this paper is organized as follows: Section II provides the experimental processes which include data pre-processing, deep learning algorithms. Section III presents the proposed model - DeepBiG. Evaluation matrices is shown in Section IV. Section V shows the experiment results and discusses the finding. Finally, Section VI concludes the paper.

II. Materials and Methods

The dataset includes 22832 labelled ChIP-seq data GM12878 lymphoblastoid cell and 21559 labelled ChIP-seq K562 chronic myelogenous leukemia (CML) cell. GM12878 is generated by Epstein–Barr virus which may cause infectious mononucleosis. K562 is one of the immortalized myelogenous leukemia cell that may cause for cancers of the blood cells. The TF of GM12878 consists ELK1 (5084 ChIP-seq) and SP1(17748 ChIP-seq). The TF of K562 consists ARID3A (9526 ChIP-seq) and CTCFL (12033 ChIP-seq). The dataset are obtained from Encode which has been processed and been provided in [19]. There are 44391 DNA cells in total and with no missing labelled. All the DNA sequences are labelled as 0 or 1. GM12878 sequences are labelled as 0 while K562 sequences are labelled as 1. Noted that the rational to choose the number of ChiP-seq is almost balance is to ensure model may perform the unbiased prediction. Saying that if given the ChIP-seq, the probability to perform the prediction manually towards the given ChIP-seq either is either GM12878 cell or K562 is about 50%.

A. Data Pre-processing

The data pre-processing steps are to transform the dataset into a uniform format that can be understood by the learning algorithms include k-mers and text tokenizer. k-mers is the common method for tokenizing the genome that splitting the long DNA sequence into k length biological sub-sequences [25], [26]. As shown in Table I, there are five k-mers where we can tokenize the sequence "AGGTCCGGGTCT". The five different k-mers will result different tokens and hence affect the performance of the language models. The k-mers range is between two until six are chosen as 1-mer will not provide any useful DNA sequence relation and accuracy prediction after 6-mers is decreased.

TABLE I. Example Biological Sub-sequences Generated by k-mers and the Number Sequence Index

v

k-mers	Biological k-mers sub-sequences	Distinct Sequence
2	AG GG GT TC CC CG GG GG GT TC CT	16
3	AGG GGT GGC TCC CCG CGG GGG GGT	64
4	AGGT GGGC GTCC TCCG CCGG CGGG GGGT	256
5	AGGTC GGTCC GTCCG TCCGG CCGGG CGGGT	1024
6	AGGTCC GGTCCG GTCCGG TCCGGT CCGGTC	4096

Tokenizer is essential to boost the performance of the natural language processing model. Firstly, creates a bag of "vocabulary" of ChiP-seq by transforming the splitting block of sequence based on k-mers into integer. For example, the bag of "vocabulary" for 2-mers ChIP-seq with 16 number of distinct sequence is 'gg'-1, 'cc'-2, 'gc'-3, 'ct'-4, 'ag'-5, 'tg'-6, 'ca'-7, 'tc'-8, 'ga'-9, 'tt'-10, 'aa'-11, 'cg'-12, 'gt'-13, 'ac'-14, 'at'-15, 'ta'-16. Next, converts the long ChIP-seq into integer based on the bag of "vocabulary", such that given the sequence "AGGTCCGGGTCT", the tokenizing output based on 2-mers is (5, 1, 13, 8, 2, 12, 1, 1, 13, 8, 4).

B. Convolutional Neural Networks (CNN)

Convolutional neural networks is deep neural networks that widely applied at the artificial intelligence research fields such that bioinformatics [27], [28], visual imagery [29] and natural language processing [30]. The design of CNN is composed of three layers, there are convolutional, pooling and fully connected layers. The layers of convolutional and pooling are designed to adaptive learn local information of original features, then extract and represent the spatial hierarchies features from low to high patterns through several feature maps and kernels. The layer of fully connected performs classification that maps the extracted features into final output. More specifically, a convolutional layer and pooling layer computes [31],

$$\begin{aligned} convolutional(X)_{i,k} &= Relu(\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} W^{k}_{mn} X_{i+m,n}) \\ pooling(Y)_{ik} &= max(Y_{iP,k}, Y_{iP+1,k}, ..., Y_{iP+P-1,k}) \end{aligned} \tag{1}$$

where X is the input matrix, i is index of output position, k is the filter index, W^k is an MxN matrix. Y is the output of convolutional layer. P is the pooling.

C. Gated Recurrent Unit (GRU)

Gated recurrent unit network [32] is a deep learning machine learning that to process and to uncover the underlying relationship for a given sequences of data. The data can be text, speech, video and images. Human brain activates the process of acquiring information, forgetting and memory. The memory can be long or short. The long-term memory is applied if the matters are important. Otherwise, we tend to forget, which called short-term memory. GRU is modeled like a human brain which consists the process of reset and update. The reset process help captures short-term dependencies in sequence. The update process help captures long-term dependencies in sequence. The inputs are h_{t-1} and x_t. The reset and update processes consist of two gates to manage the cell state's information. The two gates we denoted it as z_t and r_t, where z_t is the reset gate and r_t is update gate. W_z and W_r are two weight matrices as well as $b_z and b_r$ are two bias value corresponding to these two gates. The two gates are composed by a sigmoid neural net layer (σ) and a pointwise multiplication operation. The candidate hidden state we denoted it as $\hat{C}_t$. Here, $tanh$ function is activated. The t_t is the hidden state. The value of z_t is either close one or close zero. Old state is retained, if value of z_t is closed to one. Otherwise, new latent state t_t.

Noted that, these layers are repeating and form a chain. The gate structures and cell states are calculated as follows [32]:

$$\begin{aligned} z_t &= \sigma(W_z x_t + V_z h_{h-1} + b_z) \\ r_t &= \sigma(W_r x_t + V_r h_{h-1} + b_r) \\ \hat{C}_t &= tanh(W_C x_t + V_C(r_t.h_{t-1}, x_t)) \\ t_t &= z_t.h_{t-1} + (1 - z_t).\hat{C}_t \end{aligned} \tag{2}$$

D. Bidirectional GRU (BiGRU)

Bidirectional GRU [33] accomplishes the training without the limitation of using input information just up to a present future frame. It predicts the sequence for each class using finite sequence based on the context of elements of past and future. One can see the two GRUs are executed parallel, one is forward and another one is backward. Eq. 3 and 4 shows the calculation of BiGRU that takes L inputs and H number of hidden units. The final output of BiGRU is based on the hidden BiGRU forward and backward values [33].

$$a_h^t = \sum_{l=1}^{L} x_l^t.w_{lh} + \sum_{h',t>0}^{H} b_{h'}^{t-1}.w_{h'h} \tag{3}$$

$$a_t^h = \theta_h(a_t^h) \tag{4}$$

III. The Proposed Model (DeepBiG)

CNN architecture consists of convolutional layer, pooling layer and fully connected layer. The proposed model is composed with the modified CNN architecture by replacing the fully connected layer with bidirectional GRU (BiGRU), the first two layers are remained, which is shown in Fig. 1. The rational of this design that remains the CNN first two layers to shorten the training time while maintains the accuracy during data processing by generalizing ChIP-seq patterns. Next, replacing fully connected layer with the BiGRU is to deal with the past and present order dependency information in the ChIP-seq which may efficiently characterize the highly complex order of ChIP-seq.

The first layer is a convolutional layer which is constructed with 32 filters, five kernels with rectified linear units (relu) as the activation function. During the training phase, the filters and kernels read the input matrices with same weights, produces different strengths of signals and extracts the correlation ChIP-seq patterns.

The second layer is a max pooling layer to improve the reliability and performance in term of time for the proposed model. It summarizes the feature maps so that the model will not need to be trained by maximizing the output signals of each kernel along the entire sequence.

The third layer is BiGRU to process the filtered correlated ChIP-seq with its own interpretation by considering the context of elements of past and future into its hidden state. The interpretation is further propagated to the next GRU block. Once the nucleotide is remarked, the last block of GRU makes the final decision for the goodness of the probe.

The last layer is a non-linear transformation with sigmoid activation. The sigmoid activation will produce a value between 0 and 1. This value represents the probability of a binding preference of each probe. In this case, 0 is GM12878 ChIP-seq, 1 is K562 ChIP-seq.

The proposed model is implemented based on Keras library. The experiment is undergo three phases: training, validation and testing. For the training phase, the experiment will randomly train 50% of the dataset and 25% dataset is validated at validation phase. Then, remaining 25% dataset is tested at testing phase. Early stopping is applied for overfitting. Lastly, the performance for the models are evaluated. The model is simulated on on graphical processing units (GPU) with Intel(R) Core (TM) i9-10980XE CPU@ 3.00GHz, 128GB random access memory and 1T hard disk.

IV. Performance Metrics

A confusion metric is used to assess the performance of the models on the data as shown in Table II. True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are four assessment elements formulated in the confusion matrix table. TN is both predict and actual events fall on GM12878 ChIP-seq. The sequence is not K562 ChIP-seq. FP is prediction is the model incorrectly classified GM12878 ChIP-seq as K562 ChIP-seq. FN is the prediction is GM12878 ChIP-seq but the actual is K562 ChIP-seq. TP is both predict and actual events fall on K562 ChIP-seq that are correctly classified by the model.

TABLE II. Confusion Matrix

	GM12878 seq	K562 seq
GM12878 seq	True Negative (TN)	False Positive (FP)
K562 seq	False Negative (FN)	True Positive (TP)

With this matrix, one may evaluate the performance of the design model based on accuracy, precision, recall, and F1-score are as follows.

1) Accuracy: Accuracy refers to how close a measurement is to the accepted value. As shown in Eq. 5 [34], the accuracy is the proportion of correct predictions for both true positive and true negative. High accuracy requires high precision and high trueness.

$$Accuracy(A) = \frac{TP + TN}{TP + FP + FN + TN} \tag{5}$$

2) Precision: Precision refers to positive predictive value. As shown in Eq. 6 [34], the precision is the fraction of correct predictions among the true and false positive (such as correct predict the DNA sequence is K562 ChIP-seq). High precision requires high trueness.

$$Precision(P) = \frac{TP}{TP + FP} \tag{6}$$

3) Recall: Recall refers to sensitivity of the model in capturing true positive value. As shown in Eq. 7 [34], the recall value is the fraction of correct predictions among the actual positive value.

$$Recall(R) = \frac{TP}{TP + FN} \tag{7}$$

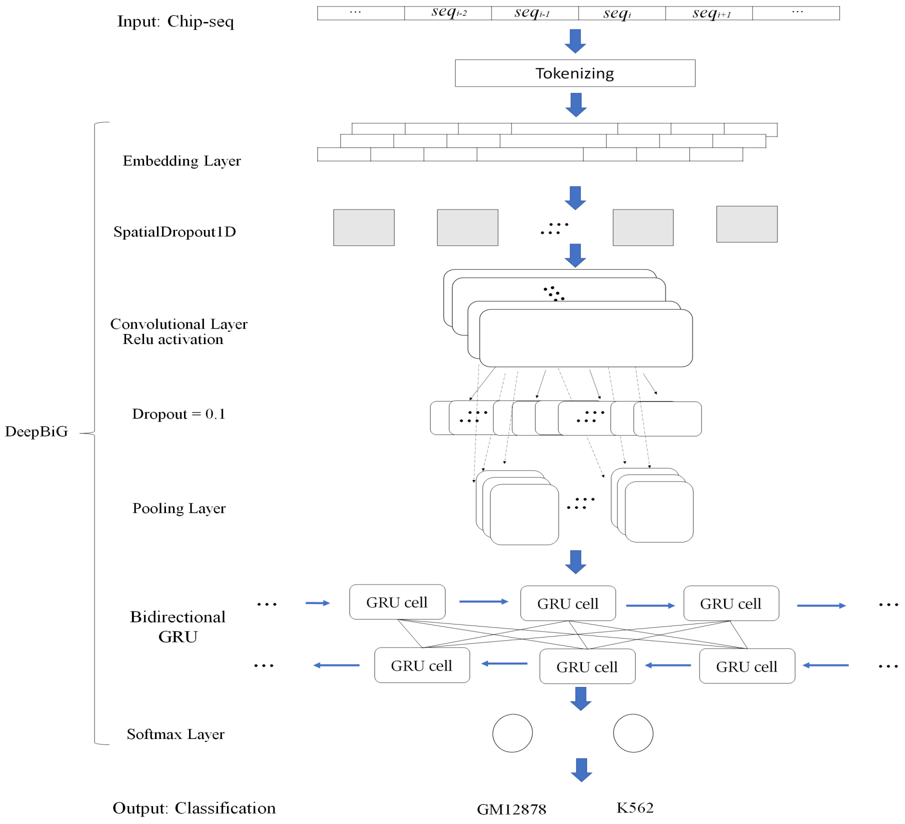

Fig. 1. Proposed Model - DeepBiG.

4) F1-Score: F1-score refers to seek for balance of the precision and recall. As shown in Eq. 8 [34], F1-score measures is there any uneven class distribution.

$$F1 - Score(F1) = \frac{2 * P * R}{P + R} \tag{8}$$

V. Result and Discussion

The results and discussion consist tables of performance metric in percentage that include accuracy, precision, recall and F1-score as well as model training time in minutes. There are four types of performance metrics comparison: 1) The comparison k-mers spectra with each being tokenized before being trained based on DeepBiG. 2) The comparison performance metric with difference combination of activation functions. 3) The comparison performance metric with different types of models and classifiers. 4) The comparison performance metric with different datasets.

A. Performance Comparison with Different k-mers

Table III shows the performance metric for k-mers spectra to evaluate genome assemblies. Noted that Class 0 is GM12878 and Class 1 is K562. Time is model training time in minutes. There are 2-mers, 3-mers, 4-mers, 5-mers and 6-mers. The accuracy increases from 2-mers to 4-mers and decreases after 4-mers. The simulations show 4-mers outperforms compare with others k-mer spectra with only 63 minutes in model training and 82.90% accuracy in predicting the ChIP-seq either belong to GM12878 or K562. However, the F1-score, the weighted average of precision and recall for 3-mers is better with only 1% different between class 0 and class 1 compare with 4-mers with 2% difference. But, the training time for 3-mers is double compare with 4-mers.

TABLE III. Performance Results in Term of Accuracy, Precision, Recall and F1-score for DeepBiG Model

k-mers	Time	A (%)	Class	P(%)	R(%)	$F1$(%)
2	99	81.70	0	78	90	84
			1	87	73	79
3	133	82.70	0	85	81	83
			1	81	84	82
4	63	82.90	0	82	86	84
			1	84	80	82
5	91	79.60	0	82	78	80
			1	78	81	79
6	67	77.30	0	84	70	76
			1	72	86	78

Fig. 2 displays the accuracy and loss for 4-mers during the training and validation phases with the highest accuracy is 88.17% and 82.80%, respectively. Early stopping at epochs 8 as overfitting occurred. This is one of the limitation of the design but it provides better accuracy compare with other models or classifiers as shown in Table V. Therefore, 4-mers encoding DeepBiG is chosen for the remaining experiments.

B. Performance Comparison with Different Activation Functions

Table IV shows the simulation results for different combination activation functions like relu, softmax and tanh. These

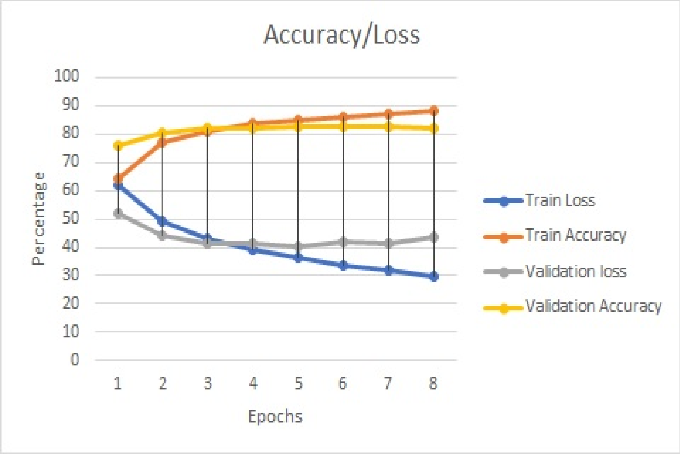

Fig. 2. DeepBiG using 4-mers, the training / validation's accuracy and loss.

activation functions are generally applied in deep learning models. The result shows that activation softmax for the last layer is preferable as it compatible with the adam optimizer and categorical cross-entropy loss. The accuracy for models where last layer is softmax activation is more than 80%. Relu and tanh are not suitable to be placed at last layer as vanishing gradient problem, in this simulation, class 0 is on higher side as the number of dataset is larger compares with class 1, then the gradient will be near zero. This has resulted no learning during backpropagation for class 1 as weights is updated with really small values. Noted that the simulation dataset for Class 0 is GM12878 and Class 1 is K562. Time is model training time in minutes.

DeepBiG is using relu activation at CNN layer and softmax activation at last layer. The performance in term of accuracy is higher almost 2% compares with relu and tanh activation in CNN layer. DeepBiG training time is double faster compares to softmax activation and 6 minutes quicker compares to tanh activation.

TABLE IV. Comparison Performance Results in Term of Accuracy, Precision, Recall and F1-score for Different Types of Models

Models	Time	A (%)	Class	P(%)	R(%)	$F1$(%)
Relu* - Softmax$^+$	63	82.90	0	82	86	84
			1	84	80	82
Softmax* - Softmax$^+$	134	81.00	0	86	76	81
			1	77	86	81
Tanh* - Softmax$^+$	76	80.60	0	81	81	81
			1	80	80	80
Softmax* - Tahn$^+$	34	51.40	0	51	100	68
			1	0	0	0
Softmax* - Relu$^+$	39	51.40	0	51	100	68
			1	0	0	0
Relu* - Relu$^+$	34	51.40	0	51	100	68
			1	0	0	0

C. Performance Comparison with Different Existing Predictors

Table V compares the performance in term of accuracy, precision, recall and F1-score between the deep learning models and machine learning classifiers. The dataset for Class 0 is GM12878 and Class 1 is K562. Time for these simulation is recorded in minutes. The deep learning models are bidirectional GRU (BiGRU), bidirectional long short-term memory (BiLSTM), CNN and the combination models. The machine learning classifiers are Naïve Bayes (NB), K-Nearest Neighbors Algorithm (KNN) and Random Forest (RF). Each method is simulated using the same dataset as stated and the dataset undergoes the pre-processing process as shown in Section II. The parameters for deep learning models are similar with DeepBiG which using dropout ratio of 0.1, kernel number is 5, cell number is 10, epochs is 15 and batch size is 64. The parameters for machine learning classifiers are varied for each others. NB smoothing parameter is set between the range of 0.1, 1, 10, 100 and 1000. In the KNN classification, the number of neighbors to be used in this simulation is in the range of 2, 5, 8, 10 and 15. For RF, the number of trees in the forest is set in the range of 10, 25, 30, 50, 100 and 200. The maximum depth of the tree is in the range of 2, 3, 5, 10 and 20. The minimum number of samples require to be at a leaf node is in the range of 5, 10, 20, 50, 100 and 200.

Deep learning models outperform compare to machine learning classifiers in term of accuracy, average is 80%. Machine learning classifiers' accuracy in average 55%. The result demonstrates the performance in term of accuracy of the proposed DeepBiG model is the highest (82.90%) by comparing with other models and classifiers on the same dataset. It follows by CNN + BiLSTM model with 81% accuracy. The training time for CNN is the only requires only 6 minutes but the accuracy below 80%. The weakest performance is NB classifier with best parameter 1000 achieves only 54.10% accuracy.

We noted overfitting with the symbol of *. The simulation dataset is long sequence with combination of nucleobases, A,C,G, and T. For each long ChiP-seq, it might contains some irrelevant DNA information related with the TF. We named it as noisy data. The models like DeepBiG, BiGRU, BiLSTM, CNN and CNN+BiLSTM learn the noisy within the training data. This has caused the overfitting. Hence, two solutions are provided to overcome the overfitting by adding dropout in the model and early stopping during training phase.

D. Performance Comparison with Different Size of Datasets

To further assess the performance of DeepBiG, we conduct experiments on four different combination TF datasets using DeepBiG and CNN+BiLISTM as shown in Table VI. The dataset include GM12878-ELK1, GM12878-SP1, K562-ARID3A and K562-CTCFL. Class 0 is GM12878 and Class 1 is K562. Simulation time is recorded in minutes.

Based on the results, we find that when the dataset size is smaller, the accuracy rate for DeepBiG is above 82.9%. If the dataset is smaller, the noisy decreases, this has resulted the increase of accuracy rate. Hence, this has proven there are noisy for the simulation dataset in Table V indirectly.

From Table VI, one may find that the DeepBiG predicts well when the dataset is larger compares to CNN+BiLSTM model. DeepBiG predictions accuracy for dataset EC, SA and SC are 87.90%, 84.10% and 89.40% respectively. CNN+BiLSTM model predictions accuracy for dataset EC, SA

TABLE V. COMPARISON PERFORMANCE RESULTS IN TERM OF ACCURACY, PRECISION, RECALL AND F1-SCORE FOR DIFFERENT TYPES OF MODELS AND CLASSIFIERS

Models/Classifiers	Time	A (%)	Class	P(%)	R(%)	$F1$(%)
*DeepBiG	63	82.90	0	82	86	84
			1	84	80	82
*BiGRU	360	78.10	0	84	71	77
			1	73	86	79
*BiLSTM	628	80.30	0	80	83	81
			1	81	78	79
*CNN	6	79.20	0	80	79	80
			1	78	79	79
BiGRU+CNN	509	80.10	0	78	85	82
			1	82	75	78
BiLSTM+CNN	635	79.90	0	81	80	80
			1	79	80	79
*CNN+BiLSTM	99	81	0	87	74	80
			1	76	88	82
BiGRU+CNN+BiGRU	830	80.90	0	81	82	82
			1	81	79	80
BiLSTM+CNN+BiLSTM	1244	80.70	0	78	88	82
			1	85	73	79
BiGRU+CNN+BiLSTM	978	80.70	0	81	82	81
			1	80	79	80
BiLSTM+CNN+BiGRU	1261	79.60	0	77	86	81
			1	83	73	78
NB	34	54.10	0	56	56	56
			1	52	52	52
KNN	1826	57.10	0	58	60	59
			1	56	54	55
RF	291	61.90	0	61	74	67
			1	64	50	56

and SC are 87.40%, 83.80% and 88.80% respectively. DeepBiG has 0.5% more accurate compares to CNN+BiLSTM. But, for dataset EA with total 14610 labelled data, the prediction accuracy for DeepBiG is 0.5% less than CNN+BiLSTM. However, the overall training time for DeepBiG is faster compares to CNN+BiLSTM.

TABLE VI. PERFORMANCE RESULTS IN TERM OF ACCURACY, PRECISION, RECALL AND F1-SCORE FOR DEEPBIG MODEL AND CNN+BILSTM WITH DIFFERENT DATASETS

Models	Time	A (%)	Class	P(%)	R(%)	$F1$(%)
DeepBiG-EA[1]	26	84.80	0	78	78	78
			1	88	88	88
DeepBiG-EC[2]	37	87.90	0	82	76	79
			1	90	93	92
DeepBiG-SA[3]	32	84.10	0	86	91	88
			1	80	72	76
DeepBiG-SC[4]	53	89.40	0	91	92	91
			1	87	86	87
CNN+BiLSTM-EA[1]	71	85.30	0	81	76	78
			1	87	90	89
CNN+BiLSTM-EC[2]	57	87.40	0	77	83	79
			1	92	89	91
CNN+BiLSTM-SA[3]	62	83.80	0	85	92	88
			1	82	69	75
CNN+BiLSTM-SC[4]	105	88.80	0	93	88	90
			1	84	89	87

EA - GM12878-ELK1 and K562-ARID3A with total dataset 14610.[1] . EC - GM12878-ELK1 and K562-CTCFL with total dataset 17117.[2] . SA - GM12878-SP1 and K562-ARID3A with total dataset 27274.[3] . SC - GM12878-SP1 and K562-CTCFL with total dataset 29781.[4]

VI. CONCLUSION AND FUTURE WORK

In this paper, the combination of k-mers encoding with tokenizing have been introduced for the data pre-processing phase. In the experiments, the DNA sequences are sized from 2-mers up to 6-mers are considered. The hybrid deep learning algorithms, we named it as DeepBiG is proposed with combination of CNN and BiGRU. DeepBiG is simulated and is analysed in terms of training time, accuracy, precision, recall and F1-score. The results reveal that the proposed 4-mers encoding DeepBiG gives better accuracy with 82.90% when compares with other deep learning models and machine learning classifiers. Although our model achieves better result, there is a limitation of DeepBiG is overfitting. Therefore, dropout and early stopping is adding into the model. There are open researches for improving this model which may still preserve the accuracy during learning, validation and prediction phases. For example, noise deduction during data pre-processing and reduces overfitting at model training phase.

ACKNOWLEDGMENT

This work was fully funded by Guangdong University of Science and Technology (Grant no. GKY2023BSQD-46). As well as cooperate with Program ASEAN Talented Young Scienties, GuangXi Academy of Science, National Natural Science Foundation of China (Grant nos. 62333018, U22A2039, and 62372318), respectively, and the Key Project of Science and Technology of Guangxi (Grant no. 2021AB20147), Guangxi Natural Science Foundation (Grant Nos. 2022JJD170019 & 2021JJA170204 & 2021JJA170199).

REFERENCES

[1] H. Yan, S. Tian, S. L. Slager and Z. Sun,*ChIP-seq in studying epigenetic mechanisms of disease and promoting precision medicine: progresses and future directions*, Epigenomics, 8(9), 1239-1258, 2016.

[2] Z. Zou, M. Iwata, Y. Yamanishi and S. Oki,*Epigenetic landscape of drug responses revealed through large-scale chip-seq data analyses*, BMC bioinformatics, 23(1), 1-20, 2022.

[3] C. D. Aimone, J. S. Hoyer, A. E. Dye, D. O. Deppong, S. Duffy, I. Carbone and L. Hanley-Bowdoin,*An experimental strategy for preparing circular ssDNA virus genomes for next-generation sequencing*, Journal of Virological Methods, 300, 114405, 2022.

[4] A. L. Bowes, M. Tarabichi, N. Pillay, and P. Van Loo,*Leveraging single-cell sequencing to unravel intratumour heterogeneity and tumour evolution in human cancers*, The Journal of Pathology, 2022.

[5] V. A. Sontakke and Y. Yokobayashi,*Programmable macroscopic self-assembly of DNA-decorated hydrogels*, Journal of the American Chemical Society, 144(5), 2149-2155, 2022.

[6] S. Roth, D. Ideses, T. Juven-Gershon and A. Danielli,*Rapid biosensing method for detecting protein–DNA interactions*, ACS sensors, 7(1), 60-70, 2022.

[7] E. Scaglione, G. De Falco, G. Mantova, V. Caturano, A. Stornaiuolo, A. D'Anna and P. Salvatore,*An experimental analysis of five household equipment-based methods for decontamination and reuse of surgical masks*, International journal of environmental research and public health, 19(6), 3296, 2022.

[8] A. Voulodimos, N. Doulamis, A. Doulamis and E. Protopapadakis,*Deep learning for computer vision: A brief review*, Computational intelligence and neuroscience, 2018.

[9] A. Haghighat and A. Sharma,*A computer vision-based deep learning model to detect wrong-way driving using pan–tilt–zoom traffic cameras*, Computer-Aided Civil and Infrastructure Engineering, 38(1), 119-132, 2023.

[10] L. Wu, Y. Chen, K. Shen, X. Guo, H. Gao, S. Li, J. Pei and B. Long,*Graph neural networks for natural language processing: A survey*, Foundations and Trends® in Machine Learning, 16(2), 119-328, 2023.

[11] M. Anand, K.B. Sahay, M.A. Ahmed, D. Sultan, R.R. Chandan and B. Singh,*Deep learning and natural language processing in computation for offensive language detection in online social networks by feature selection and ensemble classification techniques*, Theoretical Computer Science, 943, 203-218, 2023.

[12] C. Xia and H. Shen,*Deep Learning Techniques for De novo Protein Structure Prediction*, Machine Learning in Bioinformatics of Protein Sequences: Algorithms, Databases and Resources for Modern Protein Bioinformatics, 3-27, 2023.

[13] Y. Li, M. Zeng, F. Zhang, F. Wu and M. Li,*DeepCellEss: cell line-specific essential protein prediction with attention-based interpretable deep learning*, Bioinformatics, 39(1), 2023.

[14] S. Chakraborty and K. Mali,*An overview of biomedical image analysis from the deep learning perspective*, Research Anthology on Improving Medical Imaging Techniques for Analysis and Intervention, 43-59, 2023.

[15] K.S. Kumar, A. Bansal and N.P. Singh,*Brain Tumor Classification Using Deep Learning Techniques*, Machine Learning, Image Processing, Network Security and Data Sciences: 4th International Conference, MIND 2022, 68-81, 2023.

[16] F. Manavi, A. Sharma, R. Sharma, T. Tsunoda, S. Shatabda and I. Dehzangi,*CNN-Pred: Prediction of single-stranded and double-stranded DNA-binding protein using convolutional neural networks*, Gene, 853, 147045, 2023.

[17] A.B. ÖNCÜL,*LSTM-GRU Based Deep Learning Model with Word2Vec for Transcription Factors in Primates*, Balkan Journal of Electrical and Computer Engineering, 11(1), 42-49, 2023.

[18] H. Luo, W. Shan, C. Chen, P. Ding, and L. Luo,*Improving language model of human genome for DNA–protein binding prediction based on task-specific pre-training*, Interdisciplinary Sciences: Computational Life Sciences, 15(1), 32-43, 2023.

[19] B. Alipanahi, A. Delong, M.T. Weirauch and B.J Frey,*Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning*, Nature biotechnology, 33(8), 831-838, 2015.

[20] S. Salekin, J.M. Zhang and Y. Huang,*A deep learning model for predicting transcription factor binding location at single nucleotide resolution*, 2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), 57-60, 2017.

[21] J. Zhou and O.G. Troyanskaya,*Predicting effects of noncoding variants with deep learning–based sequence model*, Nature methods, 12(10), 931-934, 2015.

[22] A. Gupta and A.M Rush,*Dilated convolutions for modeling long-distance genomic dependencies*, 2017.

[23] Z. Shen, W. Bao and D. Huang,*Recurrent neural network for predicting transcription factor binding sites*, Scientific reports, 8(1), 15270, 2018.

[24] D. Quang and X. Xie,*DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences*, Nucleic acids research, 44(11), 2016.

[25] P. EC. Compeau, P. A. Pevzner and G. Tesler,*How to apply de Bruijn graphs to genome assembly*, Nature biotechnology, 29(11), 987-991, 2011.

[26] B. Chor, D. Horn, N. Goldman, Y. Levy, and T. Massingham,*Genomic DNA k-mer spectra: models and modalities*, Annual International Conference on Research in Computational Molecular Biology, 571-571, 2010.

[27] Q. Zhu, X. Li, A. Conesa and C. Pereira,*GRAM-CNN: a deep learning approach with local context for named entity recognition in biomedical text*, Bioinformatics, 34(9), 1547-1554, 2018.

[28] A. Thakare, M. Bhende, N. Deb, S. Degadwala, B. Pant and Y.P. Kumar,*Classification of Bioinformatics EEG Data Signals to Identify Depressed Brain State Using CNN Model*, BioMed Research International, 2022.

[29] S. Lee, Z.J. Wang, J. Hoffman and D.H.P. Chau,*VisCUIT: Visual auditor for bias in CNN image classifier*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 21475-21483, 2022.

[30] N. Widiastuti, J. Hoffman and D.H.P. Chau,*Convolution neural network for text mining and natural language processing*, IOP Conference Series: Materials Science and Engineering, 665(2), 2019.

[31] Y. Bengio, P. Simard and P. Frasconi,*Learning long-term dependencies with gradient descent is difficult*, IEEE transactions on neural networks, 5(2), 157-166, 1994.

[32] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio,*Learning phrase representations using RNN encoder-decoder for statistical machine translation*, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.

[33] M. Schuster and K.K. Paliwal,*Bidirectional recurrent neural networks*, IEEE transactions on Signal Processing, 45(11), 2673-2681, 1997.

[34] MW. D. Powers, *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation*, Journal of Machine Learning Technologies,2(1), 37-63, 2011.