iPromoter-RNN: Sigma Promoter Identification and Classification in *E. coli* Bacteria from Genomic Data using Sequence model Approach

RIFAT RAHMAN, Department of CSE, BUET
NIBIR MANDAL, Department of CSE, BUET
SHAMSUZZOHA BAYZID, Department of CSE, BUET

Gene transcription is a very important phenomenon in organisms' cell and promoters, localized proximal to the transcription start sites of genes, are responsible for the initiation of gene transcription. As individual promoters often differ from the consensus at one or more positions, it is so much challenging to predict promoters accurately. In this study, we present a computational tool, called iPromoter-RNN, for identifying and classifying sigma factor dependent promoters in *E. coli* bacteria. We have considered five features where three of them are formulated by NLP-based approach word2vec (skip-gram) model and rest two features are formulated from structural properties of DNA sequence. iPromoter-RNN is a branched deep neural network based model where all five features are fed into both LSTM and convolutional layers and concatenated to dense layer. We also perform comparative analysis of the performance of our tool with respect to other state of the art works. Our implementation has also shown desirable result in the independent test dataset. This tool can facilitate the discovery of both general and specific types of promoters in the post-genomic era.

Additional Key Words and Phrases: Sigma promoter, Gene transcription, Sequence model

ACM Reference Format:
Rifat Rahman, Nibir Mandal, and Shamsuzzoha Bayzid. 2020. iPromoter-RNN: Sigma Promoter Identification and Classification in *E. coli* Bacteria from Genomic Data using Sequence model Approach. In *Proceedings of* . ACM, New York, NY, USA, 12 pages.

1 INTRODUCTION

Promoters are short regions near gene. Every prokaryotic species have promoters that contain specific hexamaers [2, 6] and a purine at the transcription start site (TSS). RNA polymerase must bind near the promoter for DNA transcription. *Escherichia coli* bacteria, containing prokaryotic cells, has different sigma factors in the RNA polymerase. These sigma factors are dependent on gene and environment. As RNA polymerase must bind near the promoter at the time of DNA transcription and sigma factors are residing in RNA polymerase, promoter sequence in DNA can be distinguished by depending on sigma factors. These kinds of promoters can be called sigma factor dependent promoters or sigma promoters. Different sigma factors in *E. coli* bacteria have different functions. For example, σ^{70} factor is responsible for transcription of most of the genes [8]. There are some sigma factors like σ^{32} and σ^{24} that are related to respond heat shock [19]. σ^{38} is liable for responding stress at the time of the transition from exponential growth phase to the stationary phase of *E. coli* [11]. Again, σ^{54} is accountable for nitrogen metabolism [10] and σ^{28} for flagellar genes [10].

Promoter is responsible for initiating transcription of specific genes. So, it is essential to build automatic identifier and classifier to detect & classify promoters simultaneously. Promoters may have both intra and inter class differences and similarities in terms of consensus sequences. Molecular techniques for promoter identification or classification is costly in terms of time and money. So, computational methods are needed for identification and classification tasks [22]. Again Promoters normally differ from the consensus at one or more positions. Hence, it is a challenging task to predict promoter accurately. Again, scarcity of proper data and imbalanced dataset have made the task of classification and identification so much complicated.

In this study, we are identifying whether it is sigma factor dependent promoters or not in the perspective of *E. coli* bacteria. Also we are classifying the sigma factor dependent promoters into six different categories. Here, we are taking DNA sequences with 81 nucleotide length as input. Next, we are identifying and classifying sigma promoters from that sequences.

There have been several computational models regarding the identification and classification of sigma promoters in different species. Coelho et al. [4] proposed LibSVM based model, BacSVM+ for promoter prediction in *Bacillus subtilis*. Work of Silva et al. [5] identified σ^{28} and σ^{54} dependent promoters in *E. coli* bacteria by implementing a neural network and taking DNA duplex stability as feature. Umarov et al. [23] built CNN based classifier for promoter identification in *E. coli*. Liu et al. [16] implemented iPromoter-2L where random forest classifier was used. Zhang et al. [24] proposed MULTiPly for the same task and Amin et al. [1] claimed CNN based model, iPromoter-BnCNN as the state-of-the-art study related to sigma factor dependent promoter identification and classification in *E.coli* bacteria.

Most of the studies implemented classifiers for one or two sigma factor dependent promoters except [1, 16, 24]. The sensitivity and specificity of promoter classification have shown opposing behavior for iPromoter-2L [16]. Again, the limitation of MULTiPly [24] is the selection of the basic features to work with. Furthermore, most of the literature have merged multiple binary classifiers for identification and classification which is time consuming [1].

Our *objective* is to implement a computational tool that will identify and classify sigma factor dependent promoters in *E. coli* bacteria automatically. We also perform comparative study among other state-of-the-art works.

The *contributions* of this work are: (i) representing polymers as vectors using word2vec (skip-gram) model, (ii) implementing sequence model for multi-class classification, (iii) identifying potential motif using Bi-LSTM based attention mechanism.

We implement a branched sequence model where we utilize word2vec (skip-gram) for representing polymers (e.g., trimer, tetramer, and pentamer) in DNA sequences which is a novel approach. We also consider physicochemical properties of dimer and trimer as features. Then we feed the vectors into LSTM layers, and the physicochemical info into CNN. We also implement these five branch individually and measure the performance. Our model can attain 86% accuracy in identifying promoters, and 91.54% accuracy in classifying different sigma promoters.

Our work *differs* from the existing works because we have used word2vec model to represent the polymers as vectors which can hold the syntactical characteristics of the polymer. Again, we applied sequence-based neural network model for identification and classification.

The rest of the paper is organized as follows. In Section 2 we have reviewed several related works related to sigma factor dependent promoter identification and classification with respect to *E. coli* bacteria. Section 3 describes our proposed methods. Section 4 depicts our experimental analysis. Finally, we conclude this work and provide future direction in Section 5.

2 LITERATURE REVIEW

Different computational methods have been established for identifying promoter or non-promoter, and classifying the categories of promoters. Many authors used traditional machine learning models. For promoter prediction in *Bacillus subtilis*, Coelho et al. [4] implemented BacSVM+ by utilizing LibSVM library. Lin et al. [14] also used SVM model to identify only σ^{28} and σ^{54} factor dependent promoters in *E. coli* bacteria. Based on pseudo k-tuple nucleotide composition (PseKNC) they developed iPro54-PseKNC. Another of their study [15] used pseudo nucleotide composition for feature extraction and applied SVM in order to identify σ^{70} promoters in prokaryotic species.

Liu et al. [16] implemented iPromoter-2L where they extracted physiochemical properties by multi-window-based PseKNC and utilized them as features. Authors used random forest classifier for classification. Rahman et al. [18] developed iPromoter-FSEnfor for identifying bacterial σ^{70} promoter using feature subspace based ensemble classifier and achieved an impressive accuracy of 86.32%. Zhang et al. [24] implemented MULTiPly where they extracted both local (k-tuple nucleotide composition, dinucleotide based auto covariance) and global information (bi-profile Bayes and KNN feature encodings), and performed F-score feature selection method to identify the best unique type of feature prediction results. They used SVM classifier for classification.

Several studies showed neural network based approaches for promoter classification task. For example, Silva [5] used DNA duplex stability as feature of multilayer perceptron model to identify σ^{28} and σ^{54} dependent promoter in *E. coli* bacteria. For classifying enhancer-promoter, Li et al. [13] applied a deep feature selection (DFS) model. For selecting features, He et al. [9] used PSTNPSS (Position-specific trinucleotide propensity based on single-stranded characteristic) and PseEIIP(Electron-ion potential values for trinucleotides) features. Another study used multiple windowing and minimal features. Umarov and Solovyev [23] implemented CNN based architecture on the same promoter type in *E. coli* bacteria. In iPromoter-BnCNN, Amin et al. [1] combined local features related to mono-mernucleotide sequence, tri-mernucleotide sequence, di-merstructural properties and tri-merstructural properties through the use of parallel branching.

Most of the literature have considered less than six sigma promoters for their classification models. Again, some studies only performed the identification between promoter and non-promoter. Shahmuradov et al. [21] developed a tool for predicting transcription start sites (TSSs) in five types of *E. coli* sigma promoters, such as σ^{24}, σ^{28}, σ^{32}, σ^{38}, and σ^{70} but not worked on *E. coli* sigma promoter classification. Again, iPromoter-2L showed conflicting behavior in the measurement of specificity and sensitivity. For instance, when classifying σ^{28}, σ^{32}, σ^{38}, and σ^{54}, iPromoter-2L showed that the specificity was higher than 99%, but the sensitivity was lower than 54%. Many works like MULTiPly implemented multiple binary classifier for overall classification because the benchmark dataset is imbalanced. The first sub-classifier of MULTiPly showed 85.24% accuracy in σ^{70} promoter type identification. The sensitivity and specificity was 87.27% and 86.57% respectively. The main limitation of MULTiPly was the selection of the basic features. Different combination of different heterogeneous features led to different prediction results. Effective selection of basic and essential features for the classification model is a difficult problem to solve. Through trial and error, the authors selected features that achieved satisfactory prediction performance.

3 METHODOLOGY

In this section, we provide an overview of our implementations. We have divided our methodology and results into five steps described as i) benchmark dataset selection (Subsection 3.1), ii) feature extraction from DNA sequence (Subsection 3.2), iii) model architecture (Subsection 3.3), iv) performance evaluation (Subsection 4.2), and v) public access to predictor (Subsection 4.4). Figure 1 depicts the workflow diagram of our proposed methodology and results.

3.1 Benchmark Dataset

We have collected our dataset from the RegulonDB database (Version 9.3) [7]. The dataset include the DNA sequences of *E. coli* bacteria with length of 81 bp and corresponding class (either non-promoter or any of six different sigma factor

,

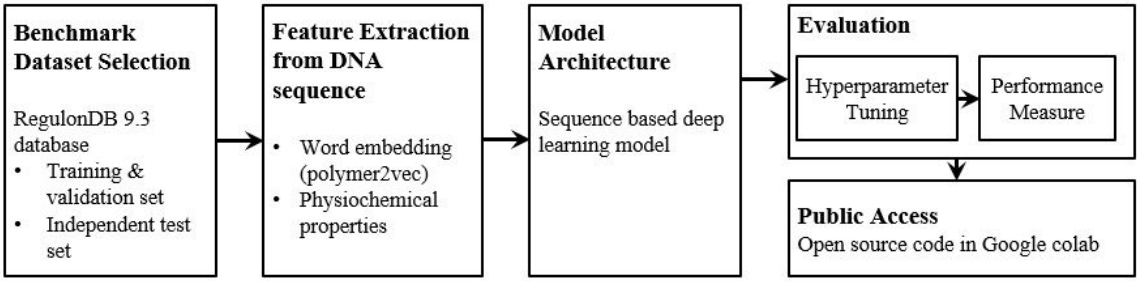

Benchmark Dataset Selection
RegulonDB 9.3 database
• Training & validation set
• Independent test set
Feature Extraction from DNA sequence
• Word embedding (polymer2vec)
• Physiochemical properties
Model Architecture
Sequence based deep learning model
Evaluation
Hyperparameter Tuning
Performance Measure
Public Access
Open source code in Google colab

Fig. 1. Work flow diagram of our approaches and results

dependent promoter). This is the benchmark dataset for promoter identification and classification. All other relevant studies such as iPromoter-2L [16], MULTiPly [24], and iPromoter-BnCNN [1] also used this benchmark dataset and took all six sigma promoters into consideration. All promoter samples of the used dataset are experimentally verified and there is no type of conflicting behavior in the dataset. Lin et al. [14, 15] randomly extracted non-promoter sequences from middle regions of long coding sequences and convergent intergenic regions in E.coli K-12 genome, which are also 81 bp long. We include these sequences in the non-promoter class. We also perform the task of redundancy reduction (no two samples of same class with pairwise sequence identity is greater or equal than 0.8) using CD-HIT software [12] on our dataset. Again, we use some recently included promoter samples (experimentally verified) from RegulonDB version 10.7 [20] as our independent test dataset. Our benchmark dataset and test dataset are totally disjoint. Table 1 represents the distribution of our benchmark dataset and independent test dataset. There have been total 5720 samples for training & validation, and 256 samples for testing.

Table 1. Distribution of benchmark and independent test dataset

Classes	Sub-classes	Benchmark Dataset		Independent Test Dataset	
		Number of samples	**Total**	**Number of samples**	**Total**
Promoter	σ^{70}-promoter	1694	2860	199	256
	σ^{24}-promoter	484		30	
	σ^{32}-promoter	291		13	
	σ^{38}-promoter	163		10	
	σ^{28}-promoter	134		4	
	σ^{54}-promoter	94		0	
Non-promoter	–	–	2860	–	0
Total			5720		256

3.2 Feature Extraction from DNA sequence

Machine can not understand molecular sequence, it can only understand numeric values. So, we extract feature from DNA sequences. Expressing biological sequences by preserving sufficient sequence-order information is challenging. We illustrate the biological sequences into vector representations in several ways. We describe the techniques in this subsection.

3.2.1 *Word2vec using skip-gram.*

Word2vec is a unsupervised shallow neural network based model which was developed by Mikolov et al. [17]. at Google.

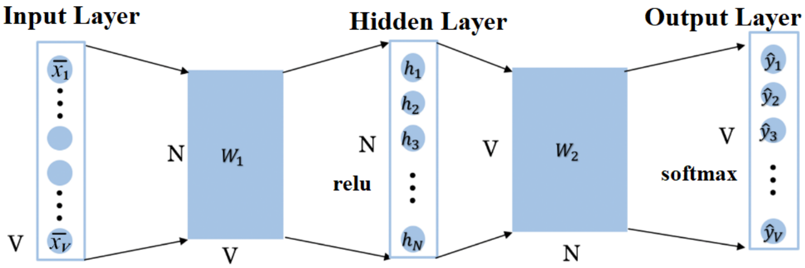


Fig. 2. Architecture of Word2Vec model

Figure 2 depicts the architecture of Word2Vec model. This model is a 2-layer neural network. The input of the model is one hot encoding vector of center word or the average one hot encoding vector of context words with dimension $V \times 1$. Here, V is called the vocabulary size which depends on the size of the polymers from input DNA sequence. For example, possible number of trimers is 64, so if we consider trimer, the value of V will be 64. In the same way, the value of V of tetramer and pentamer are 256 and 1024 respectively. The number of neurons of the hidden layer is same as dimension size, N. We take the dimension size of trimer, tetramer, and pentamer as 30, 120, and 300 respectively. The number of hidden nodes for output layer is equal to the vocabulary size, V. All the three layers are fully connected. The activation function, used in the hidden layer, is $relu$ and the activation function for the output layer is $softmax$. The dimension of the first weight matrix, W_1 is $N \times V$ and the second weight matrix, W_2 is $V \times N$. We transpose the first weight matrix and take average with the second one. Finally we get an average weight matrix that has dimension $V \times N$. Thus we learn different word embeddings by tuning dimension and sliding window size. So, from Figure 2, we are getting,

$$h = ReLU(W_1 X + b_1)$$

$$\hat{y} = softmax(W_2 h + b_2)$$

Here, W_1 & W_2 are weight matrices, b_1 & b_2 are bias matrices, and X is input.

We feed the DNA sequence from Subsection 3.1 to the word2vec (skip-gram) model by pretending polymers (e.g., trimer, tetramer, and pentamer) as words. 'Skip-gram' (SG) takes center word (polymer) as input and then predict the probabilities of the context words. So, this implementation takes one hot encoding vector of center word as input to predict context words.

For reducing the overfitting, we use 'adam' optimizer with learning rate as 0.01 and we evaluate 'categorical cross-entropy loss' for cost computation. We take the batch size as 256. Thus, each DNA sample (81 nucleotide length) is represented by 79×30, 78×120, and 77×300 size two dimensional matrices if we consider overlapping trimers, tetramers, and pentamers as input words.

3.2.2 *Physicochemical Properties.*

Physicochemical properties refers to dynamic DNA structure which are potential to change in conformation. It indicates specific characteristics of DNA like rigidity, stability, curvature, etc. PseKNC (Pseudo K-tuple nucleotide composition) [3] tool is used to convert DNA sample dataset into polymers providing many choices of physicochemical combinations.

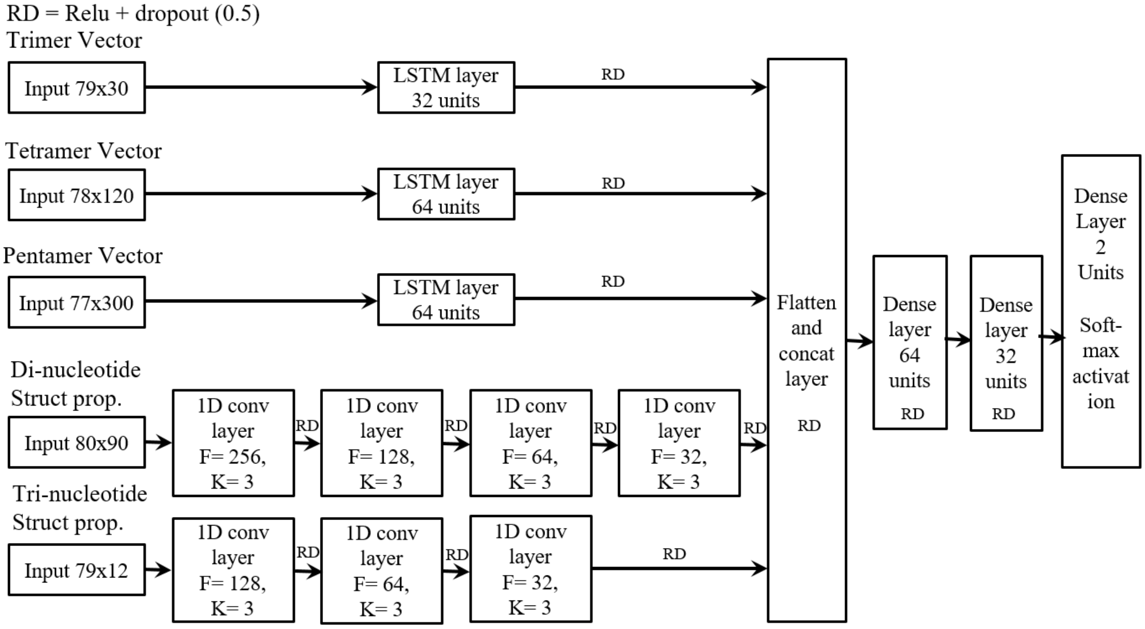


Fig. 3. Model architecture of ipromoter-RNN

This tool provides 90 physicochemical properties for every 16 possible dimers, and 12 physicochemical properties for every 64 possible trimers. Thus, each DNA sample (81 nucleotide length) is represented by 80×90, and 79×12 size two dimensional matrices if we consider overlapping dimers, and trimers as input words.

3.3 Model Architecture

We have implemented our model architectures into two different ways. First of all, we have implemented individual LSTM models where we take trimer, tetramer, and pentamer embeddings as features. We evaluate the performances of these three LSTM models. Then we implement a branched deep neural network where we use both LSTM and CNN. For physicochemical properties 3.2.2, we chose two k-tuple nucleotide (e.g., dimer structural properties, and trimer structural properties) and feed into convolutional layers. Again, we concatenate the previous three LSTM models. As a result, we get a branched deep neural network architecture.

From Figure 3, we can observe total 5 branches. First three branches are taking polymer embeddings as input feature and they are fed into LSTM layers. Trimer, tetramer, and pentamer vectors are representing the evaluated vectors from word2vec (skip-gram) model. Again, the bottom most two layers are fed into several convolutional layers. From the Figure 3, K stands for the kernel size, and F stands for the total number of filters. Then there includes a flatten layer with separate branches. They concatenate to themselves. After the concatenation, two dense layer with 64 and 32 hidden units are added. Finally, a dense layer with 2 units and softmax activation function is added. All the internal layers are integrated with ‘relu’ activation function.

We choose all the values of hyperparameters, such as total number of hidden layer, hidden units per layer, activation function, learning rate, etc by fine-tuning. We choose our loss function as "categorical cross-entropy" and optimizer as "adam". We select our initial learning rate 0.01 and also utilize learning rate reduction mechanism.

We also implement each branch individually to find out the performance of each branch. At the same time we present the comparative analysis of different features. At the time of evaluating three LSTM models, we add only one dense layer after the flatten layer. The number of hidden units are 16, 32, and 32 for trimer, tetramer, and pentamer respectively.

At the time of sigma promoter classification task, we implement separate models for binary classification because the dataset is imbalanced. From Table 1, we can observe that there are total 1694 σ^{70} factor dependent promoters, whereas there are only 94 σ^{54} promoters. Because of these imbalance, a single multi-class classifier do not give any appropriate results. We divide the whole dataset into 6 parts, so that these kinds of issues can be fixed. The models used for different binary classification consist of the same model architecture.

Table 2. Use of six binary classifier to indicate the multiclass classification

Binary Classifiers	**Positive class**	**Number of samples**	**Negative classes**	**Number of smples**	**Total number of samples**
Model 1	Promoter	2860	Non promoter	2860	5720
Model 2	sigma 70	1694	Other promoters except sigma 70	1166	2860
Model 3	sigma 24	484	Other promoters except sigma 70 & 24	682	1166
Model 4	sigma 32	291	Other promoters except sigma 70, 24, & 32	391	682
Model 5	sigma 38	163	sigma 28 & 54	228	391
Model 6	sigma 28	134	sigma 54	94	228

4 RESULTS AND DISCUSSION

In this section, we present the comparative performance analysis of our proposed models. We also show comparisons with other state-of-the-art studies.

4.1 Experimental Setup

We use python programming language for all experiments. We do program on 'Google Colab'[1] platform. This platform provides both GPU and TPU services. We use Keras library with Tensorflow background for the implementation of deep learning based models. For data visualization and manipulation, we take support from some Python libraries including numpy, pandas, matplotlib, etc.

For implementing sequence based deep neural network based models, we use 80% of benchmark data for training and 20% for validation. Again, we run total 20 epochs. We also apply 5-fold cross validation, so that we can get the most accurate result.

[1] https://colab.research.google.com/notebooks/intro.ipynb

,

4.2 Performance Metrics

We have calculated accuracy (acc), sensitivity (Sn), Specificity (Sp), and Mathew's Correlation coefficient (MCC) to measure the performances of our implemented supervised models with other studies. The formulas of these performance metrics are:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Sn = \frac{TP}{TP + FN}$$

$$Sp = \frac{TN}{TN + FP}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

All symbols are directed towards binary classification. Here, TP, TN, FP and FN denote the number of true positive, true negative, false positive and false negative samples depending on model predicted label. Sn and Sp are also known as true positive rate and true negative rate respectively. Values related to accuracy, sensitivity and specificity lie in the range [0, 1] while for MCC score, the range is [-1, +1]. Higher value indicates better classification ability.

4.3 Result Analysis

4.3.1 Comparison among our implemented models.

We have conducted both individual experiments and the combined branched one.

Table 3. Promoter identification performance analysis among different individual models

Feature	Model	Accuracy	Specificity	Sensitivity	MCC
Trimer embeddings	LSTM	0.796	0.78	0.81	0.59
Tetramer embeddings	LSTM	0.808	0.809	0.807	0.616
Pentamer embeddings	LSTM	**0.831**	0.786	**0.877**	**0.666**
Di-nucleotide structural properties	Deep CNN	0.763	**0.856**	0.669	0.535
Tri-nucleotide structural properties	Deep CNN	0.79	0.807	0.772	0.58

From Table 3, we can observe the comparative performance analysis of different individual models. We find that pentamer vectors using word2vec (skip-gram) with LSTM model is accuracy, sensitivity, and MCC score. Deep CNN with di-nucleotide physicochemical properties is giving highest score in specificity, but the difference between specificity and sensitivity is very high for this model. Tetramer embedding feature with LSTM is showing low difference between specificity and sensitivity.

Again, in the perspective of feature extraction, polymer embeddings are far better than the physicochemical properties. This indicates that the syntactical and positional characteristics of polymers are so much important factor in a DNA sequence. From Figure 4, we can observe that word embedding features (e.g., trimer, tetramer, and pentamer embedding) are giving higher accuracy, Mcc, and sensitivity than structural properties.

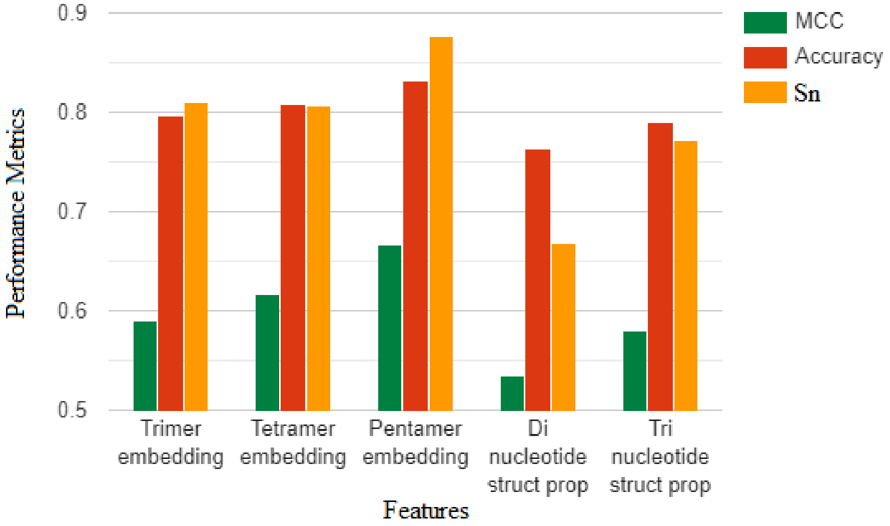


Fig. 4. Comparative analysis of different features

We also find significant performance when we concatenate all of individual models into a branched neural network. We have named this model as iPromoter-RNN. The accuracy, specificity, sensitivity, and Mcc score of iPromoter-RNN are 86%, 83.7%, 86.5%, and 0.719 respectively.

4.3.2 *Comparison with other related studies.*

We conduct comparative performance analysis of our implemented tool (iPromoter-RNN) with other related works. Table 4 depicts the comparison among other related studies.

Table 4. Promoter identification performance comparison using 5-fold cross-validation on benchmark dataset

Method	Accuracy	Specificity	Sensitivity	MCC
PCSF	0.748	0.707	0.789	0.498
vw Z-curve	0.803	0.828	0.778	0.61
Stability	0.78	0.795	0.766	0.562
iPro54	0.805	0.832	0.778	0.61
iPromoter-2L	0.817	0.842	0.792	0.634
MULTiPly	0.869	0.866	0.873	0.739
iPromoter-BnCNN	0.882	0.88	0.883	0.763
iPromoter-RNN	0.86	0.837	0.865	0.719

We also perform comparative analysis with the state of the art work (iPromoter-BnCNN) in classifying different types of sigma promoter. Table 5 depicts the comparative performance analysis between them. Table 6 represents the results of these two models on independent test dataset.

4.3.3 *Significance of physicochemical properties.*

There belongs to total 90 physicochemical properties for each of all 16 di-mers. Different physicochemical features

Table 5. Promoter classification performances of iPromoter-BnCNN (BnCNN) and iPromoter-RNN (RNN)

Performance Metrics	sigma24		sigma28		sigma32		sigma38		sigma70	
	RNN	BnCNN	RNN	BnCNN	RNN	BnCNN	RNN	BnCNN	RNN	BnCNN
Accuracy	0.927	**0.938**	0.944	**0.961**	0.9	**0.906**	**0.926**	0.916	**0.88**	0.873
Sensitivity	0.91	**0.933**	0.953	**0.978**	**0.919**	0.917	0.92	**0.949**	0.902	**0.91**
Specificity	**0.948**	0.941	**0.948**	0.936	0.877	**0.898**	**0.91**	0.893	**0.891**	0.822
MCC	0.859	**0.873**	0.85	**0.918**	0.89	**0.9**	**0.872**	0.833	**0.81**	0.737

Table 6. True positive and false negative results of iPromoter-BnCNN (BnCNN) vs iPromoter-RNN (RNN)

Performance Metrics	Promoter (256)		sigma24 (30)		sigma28 (4)		sigma32 (13)		sigma38 (10)		sigma70 (199)	
	RNN	BnCNN	RNN	BnCNN	RNN	BnCNN	RNN	BnCNN	RNN	BnCNN	RNN	BnCNN
TP	241	**245**	25	**28**	**2**	1	**10**	**10**	**5**	3	182	**187**
FN	15	11	5	2	2	3	3	3	5	7	17	12

have different impact on the trained model. From Table 7, we observe that MLP (Multi layer perceptron) with the "DinucleotideGC Content" values of 80 dimers as feature performs the best. Again, MLP with normalized "B-DNA twist" values of dimers as feature show the maximum accuracy and MCC. In the same way, CNN-based model works well with "Protein induced deformability" feature and RNN-based approach (LSTM) with "Hartman_trans_free_energy" feature. From Table 7, we can say that "Hartman_trans_free_energy" includes dynamic properties dependent on DNA sequence.

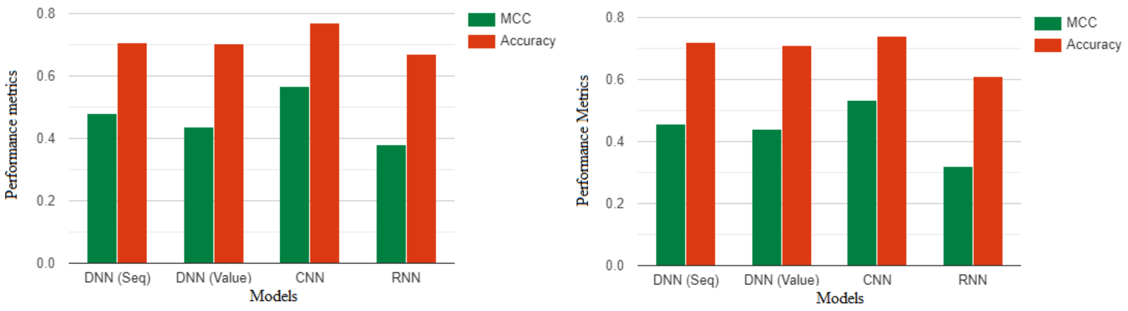
Table 7. Different neural network-based models Vs mostly significant physicochemical features

Physicochemical properties	**Model**	**Acc**	**Sn**	**Sp**	**Mcc**
Dinucleotide GC Content	MLP (Seq)	0.76	0.795	0.773	0.569
B-DNA twist	MLP (Value)	0.64	0.774	0.548	0.548
Protein induced deformability	CNN	0.77	0.757	0.816	0.575
Hartman_trans_ free_energy	RNN (LSTM)	0.76	0.708	0.798	0.508

Figure 5 depicts two different physicochemical (e.g., Lisser_BZ_transition, Flexibility_slide) which give fair performances with all kinds of neural network-based models. These two properties can be said as the most informative physicochemical properties of dimers. Again, from Figure 5, we can see that physicochemical properties literally hold static properties of genome sequences rather than syntactical properties dependent on the DNA sequence. So, MLP and CNN models are performing better than the LSTM models.

4.4 Public Code

We have made our source codes and dataset public in the google drive so that one can utilize our model for the identification and classification of sigma factor dependent promoters in *E. coli* bacteria. One can find our open source code from here and the benchmark dataset from here.



(a) Lisser_BZ_transition

(b) Flexibility_slide

Fig. 5. Two significant physicochemical properties with all type of neural network-based models

4.5 Discussions

From our findings, we can say that polymers holds syntactical information in a particular DNA sequence. Again, there belongs to positional similarity among polymers. So, influenced by NLP, we implement word2vec (skip-gram) model to find the vector representations of different polymers. We also find that this vector representation is an informative feature rather than structural characteristics. We determine the static and dynamic behavior of different physicochemical properties and identify most significant properties for different neural network-based architectures too. We also identify potential motif using Bi-LSTM based attention mechanism integrated with our model, iPromoter-RNN. We find that the motif sequence “ATAAA” belongs to 37.87% of the promoter DNA sequence in the benchmark dataset.

5 CONCLUSIONS AND FUTURE WORKS

In this work, we have applied sequence model architecture and developed iPromoter-RNN for sigma promoter identification and classification in *E. coli* bacteria. Our architecture combines four different kinds of features from each sample through the use of four one dimensional convolution branches along with coordinator dense layers at the end. We also integrate NLP-based approach word2vec (SG) to implement an informative vector representation of polymers inside DNA sequence. Our proposed tool recognizes the specific promoter types in a stage by stage manner with the goal of handling the class imbalance problem. We expect iPromoter-RNN to act as a useful automation tool in the world of computational biology. In the future, we have a plan to augment the data using GANs architecture and train semi-supervised model for better performance. Furthermore, we have a scheme of developing a species-independent promoter identification and classification tool.

REFERENCES

[1] Ruhul Amin, Chowdhury Rafeed Rahman, Sajid Ahmed, Md Sifat, Habibur Rahman, Md Nazmul Khan Liton, Md Rahman, Md Khan, Zahid Hossain, Swakkhar Shatabda, et al. 2019. iPromoter-BnCNN: a novel branched CNN based predictor for identifying and classifying sigma promoters. *Bioinformatics* (2019).

[2] Steve Busby and Richard H Ebright. 1994. Promoter structure, promoter recognition, and transcription activation in prokaryotes. *Cell* 79, 5 (1994), 743–746.

[3] Wei Chen, Tian-Yu Lei, Dian-Chuan Jin, Hao Lin, and Kuo-Chen Chou. 2014. PseKNC: a flexible web server for generating pseudo K-tuple nucleotide composition. *Analytical biochemistry* 456 (2014), 53–60.

[4] Rafael Vieira Coelho, Scheila de Avila e Silva, Sergio Echeverrigaray, and Ana Paula Longaray Delamare. 2018. Bacillus subtilis promoter sequences data set for promoter prediction in Gram-positive bacteria. *Data in brief* 19 (2018), 264–270.

[5] Scheila de Avila e Silva, Franciele Forte, Ivaine TS Sartor, Tahila Andrighetti, Günther JL Gerhardt, Ana Paula Longaray Delamare, and Sergio Echeverrigaray. 2014. DNA duplex stability as discriminative characteristic for Escherichia coli σ^{54}-and σ^{28}-dependent promoter sequences. *Biologicals* 42, 1 (2014), 22–28.

[6] Pengmian Feng, Hui Ding, Hui Yang, Wei Chen, Hao Lin, and Kuo-Chen Chou. 2017. iRNA-PseColl: identifying the occurrence sites of different RNA modifications by incorporating collective effects of nucleotides into PseKNC. *Molecular Therapy-Nucleic Acids* 7 (2017), 155–163.

[7] Socorro Gama-Castro, Heladia Salgado, Alberto Santos-Zavaleta, Daniela Ledezma-Tejeida, Luis Muñiz-Rascado, Jair Santiago García-Sotelo, Kevin Alquicira-Hernández, Irma Martínez-Flores, Lucia Pannier, Jaime Abraham Castro-Mondragón, et al. 2016. RegulonDB version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond. *Nucleic acids research* 44, D1 (2016), D133–D143.

[8] Tanja M Gruber and Carol A Gross. 2003. Multiple sigma subunits and the partitioning of bacterial transcription space. *Annual Reviews in Microbiology* 57, 1 (2003), 441–466.

[9] Wenying He, Cangzhi Jia, Yucong Duan, and Quan Zou. 2018. 70ProPred: a predictor for discovering sigma70 promoters based on combining multiple features. *BMC systems biology* 12, 4 (2018), 44.

[10] Sarath Chandra Janga and Julio Collado-Vides. 2007. Structure and evolution of gene regulatory networks in microbial genomes. *Research in microbiology* 158, 10 (2007), 787–794.

[11] Miki Jishage, Akira Iwata, Susumu Ueda, and Akira Ishihama. 1996. Regulation of RNA polymerase sigma subunit synthesis in Escherichia coli: intracellular levels of four species of sigma subunit under various growth conditions. *Journal of bacteriology* 178, 18 (1996), 5447–5451.

[12] Weizhong Li and Adam Godzik. 2006. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* 22, 13 (2006), 1658–1659.

[13] Yifeng Li, Chih-Yu Chen, and Wyeth W Wasserman. 2015. Deep feature selection: Theory and application to identify enhancers and promoters. In *International conference on research in computational molecular biology*. Springer, 205–217.

[14] Hao Lin, En-Ze Deng, Hui Ding, Wei Chen, and Kuo-Chen Chou. 2014. iPro54-PseKNC: a sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition. *Nucleic acids research* 42, 21 (2014), 12961–12972.

[15] Hao Lin, Zhi-Yong Liang, Hua Tang, and Wei Chen. 2017. Identifying sigma70 promoters with novel pseudo nucleotide composition. *IEEE/ACM transactions on computational biology and bioinformatics* (2017).

[16] Bin Liu, Fan Yang, De-Shuang Huang, and Kuo-Chen Chou. 2018. iPromoter-2L: a two-layer predictor for identifying promoters and their types by multi-window-based PseKNC. *Bioinformatics* 34, 1 (2018), 33–40.

[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[18] Md Siddiqur Rahman, Usma Aktar, Md Rafsan Jani, and Swakkhar Shatabda. 2019. ipromoter-fsen: Identification of bacterial σ70 promoter sequences using feature subspace based ensemble classifier. *Genomics* 111, 5 (2019), 1160–1166.

[19] Satish Raina, Dominique Missiakas, and Costa Georgopoulos. 1995. The rpoE gene encoding the sigma E (sigma 24) heat shock sigma factor of Escherichia coli. *The EMBO journal* 14, 5 (1995), 1043–1055.

[20] Alberto Santos-Zavaleta, Heladia Salgado, Socorro Gama-Castro, Mishael Sánchez-Pérez, Laura Gómez-Romero, Daniela Ledezma-Tejeida, Jair Santiago García-Sotelo, Kevin Alquicira-Hernández, Luis José Muñiz-Rascado, Pablo Peña-Loredo, et al. 2019. RegulonDB v 10.5: tackling challenges to unify classic and high throughput knowledge of gene regulation in E. coli K-12. *Nucleic acids research* 47, D1 (2019), D212–D220.

[21] Ilham Ayub Shahmuradov, Rozaimi Mohamad Razali, Salim Bougouffa, Aleksandar Radovanovic, and Vladimir B Bajic. 2017. bTSSfinder: a novel tool for the prediction of promoters in cyanobacteria and Escherichia coli. *Bioinformatics* 33, 3 (2017), 334–340.

[22] Michael Towsey, Peter Timms, James Hogan, and Sarah A Mathews. 2008. The cross-species prediction of bacterial promoters using a support vector machine. *Computational biology and chemistry* 32, 5 (2008), 359–366.

[23] Ramzan Kh Umarov and Victor V Solovyev. 2017. Recognition of prokaryotic and eukaryotic promoters using convolutional deep learning neural networks. *PloS one* 12, 2 (2017), e0171410.

[24] Meng Zhang, Fuyi Li, Tatiana T Marquez-Lago, André Leier, Cunshuo Fan, Chee Keong Kwoh, Kuo-Chen Chou, Jiangning Song, and Cangzhi Jia. 2019. MULTiPly: a novel multi-layer predictor for discovering general and specific types of promoters. *Bioinformatics* 35, 17 (2019), 2957–2965.