Journal of Engineering Faculty

| cumfad.cumhuriyet.edu.tr | Founded: 2023 Available online, ISSN:3023-7203 Publisher: Sivas Cumhuriyet Üniversitesi

Promoter Classification in Human Genome via DNA2Vec and UNK-Aware Deep Neural Networks

Aleyna Mengen[1,a,*], Emre Delibaş[2,b]

[1]*Sivas Cumhuriyet University, Graduate School of Natural and Applied Sciences, Department of Artificial Intelligence and Data Science, Sivas-Türkiye*
[2]*Sivas Cumhuriyet University, Faculty of Engineering, Department of Computer Engineering, Sivas-Türkiye*
**Corresponding author*

Research Article

History

Received: 04/06/2025
Accepted: 16/06/2025

ABSTRACT

This study proposes a new hybrid model combining DNA2Vec-based embedded representations with UNK character support and a deep neural network (DNN) architecture for the classification of promoter and non-promoter DNA sequences belonging to the Homo sapiens genome. The model's objective is twofold: first, to minimize the loss of contextual information, and second, to enhance the generalization performance by representing unknown or low-confidence k-mer sequences with an UNK vector. The model, which was structured with a GELU activation function and an AdamW optimization algorithm, achieved strong and balanced results, including 85.03% accuracy, 0.9376 ROC-AUC, and 0.8444 F1 score, when evaluated using a stratified 5-fold cross-validation method. The findings indicate that the proposed structure provides a more straightforward yet effective approach in comparison to the more complex models documented in the extant literature. Furthermore, this architecture provides pragmatic and comprehensible solutions in bioinformatics applications, particularly since it facilitates motif-independent learning. Future work should address the generalization capacity be increased across species and that the integration with Transformer-based models be evaluated in future studies.

Keywords: Promoter Classification, DNA2Vec, Deep Neural Network, UNK Vector, Bioinformatics

DNA2Vec ve UNK Duyarlı Derin Sinir Ağları ile İnsan Genomunda Promoter Sınıflandırması

ÖZ

Bu çalışma, Homo sapiens genomunda promoter ve non-promoter DNA dizilerinin ayrımını sağlamak amacıyla DNA2Vec tabanlı gömülü temsiller ve UNK karakter duyarlılığı ile güçlendirilmiş derin sinir ağı (DNN) mimarisini bir araya getiren hibrit bir sınıflandırma yaklaşımı önermektedir. Model, bilinmeyen veya düşük güvenilirlikteki k-mer'leri özel olarak başlatılan UNK vektörü ile temsil ederek bağlamsal bilgi kaybını önlemekte ve genelleme kapasitesini artırmaktadır. Veri seti, eşit sayıda promoter ve non-promoter diziden oluşturulmuş, değerlendirmede stratified 5-fold çapraz doğrulama uygulanmıştır. Optimize edilen model; test setinde %85.03 doğruluk, 0.8786 kesinlik, 0.8128 duyarlılık, 0.8444 F1 skoru ve 0.9376 ROC-AUC başarısı elde etmiş ve insan genomu üzerinde yapılan çalışmalarda literatürdeki pek çok karmaşık modele kıyasla daha iyi veya benzer sonuçlar göstermiştir. Sonuçlar, önerilen mimarinin güçlü, yorumlanabilir ve hesaplama açısından verimli bir alternatif sunduğunu ve motif-bağımsız öğrenme yeteneğiyle biyoinformatik uygulamalarda pratik olarak kullanılabileceğini göstermektedir. Gelecek çalışmalarda türler arası genelleme ve Transformer gibi dikkat tabanlı modellerle entegrasyonun araştırılması önerilmektedir.

Anahtar Kelimeler: Promoter Sınıflandırması, DNA2Vec, Derin Sinir Ağı, UNK Vektörü, Biyoinformatik

[a] *aleyna_mengen@hotmail.com* *0009-0008-7310-1394* [b] *edelibas@cumhuriyet.edu.tr* *0000-0001-7564-5020*

How to Cite: Mengen A, Delibaş E (2025) Promoter Classification in Human Genome via DNA2Vec and UNK-Aware Deep Neural Networks, Journal of Engineering Faculty, 3(1): 92-99.

Introduction

Nowadays, systematic classification of DNA sequences plays a fundamental role in a wide range of biomedical diagnostic and modeling processes, from understanding genetic order to discovering disease biomarkers. In this context, accurate identification of gene regulatory regions—especially promoter sequences—is critical for modeling transcriptional control, epigenetic mechanisms, and gene expression profiles. However, the increasing volume and biological complexity of genome data necessitate the development of new methods with strong generalization ability that can go beyond classical analysis approaches [1].

Promoter regions are distinguished as specific DNA segments that function as the initiators of gene expression by facilitating the binding of RNA polymerase and various transcription factors. These regions are typically situated in the vicinity of the transcription start site (TSS), enabling genetic adjustments to be responsive to factors such as timing, tissue-specific expression, and environmental response [2]. Mutations in promoter sequences have been associated with numerous genetic diseases, particularly cancer, thereby increasing the diagnostic and clinical importance of these regions [3].

Classical bioinformatics approaches have frequently relied on representation methods based on *k*-mer frequencies to transform DNA sequences into fixed-size features. However, these methods are inadequate in modeling motif locations and long-range dependencies, as they do not adequately reflect intra-sequence contextual relationships. Convolutional neural networks (CNNs) have been demonstrated to offer a partial solution to this problem; however, classical vectorization strategies predominantly necessitate manual feature engineering and exhibit constrained scalability when applied to large-scale genomic datasets [1], [4].

The development of the DNA2Vec model was motivated by the necessity to overcome the limitations of existing methods. The DNA2Vec model draws inspiration from the Word2Vec approach in natural language processing and produces dense vectors that represent DNA *k*-mers along with their contextual features [5]. Each *k*-mer is situated within a fixed-dimensional vector space, thereby facilitating the modeling of semantic closeness between sequence structures. However, limitations such as fixed window width and context averaging result in an inability to adequately represent flexibility in motif positions and long-range sequence relationships [6].

In recent years, deep learning-based approaches have been developed to model structural motifs and contextual relationships in DNA sequences with greater effectiveness. CNN-based models have demonstrated particular efficacy in the recognition of local motifs [7], [8]. However, CNNs are incapable of adequately capturing distant dependencies due to their limited filter sizes. Consequently, Transformer-based approaches (e.g., DNABERT, CyaPromBERT) have emerged as a solution. These models have been shown to facilitate the creation of more robust contextual representations by leveraging DNA sequences within the framework of natural language structures [9], [10].

As evidenced by the extant literature, a variety of CNN, RNN, and transformer-based architectures have been proposed for the prediction of promoters and the classification of DNA sequences. The PromoterLCNN [11], iPromoter-BnCNN [12], and GSCNN [13] models have demonstrated noteworthy success. Models such as GraphPro have enabled more complex classifications by combining DNA2Vec representations with CNN and GNN structures [14]. Hybrid models that integrate DNA2Vec with deep neural network (DNN) structures have exhibited successful classification performance by combining contextual representation power with learning capacity [15], [16], [17]. However, a notable technical challenge arises in the processing of the 'N' character (unknown nucleotide) in biological data, which poses a distinct problem for embedding models. Therefore, the UNK (UNKnown) vector strategy, adapted from natural language processing techniques, is employed, and unknown *k*-mers are represented either by specially defined fixed vectors or by contextual methods such as FastText and LSHvec [18], [19]. In this context, the integration of DNA2Vec-based contextual embedding models with DNN architectures offers an effective approach in both technical and biological aspects in complex bioinformatics tasks such as the classification of promoter regions. Furthermore, the employment of the UNK vector strategy has the capacity to enhance model performance by preserving the information-carrying capacity of uncertain data samples. This study proposes a hybrid approach that integrates DNA2Vec, DNN, and UNK structures in the classification of promoter and non-promoter DNA sequences of the *Homo sapiens* genome.

Materials and Methods

Dataset Description

The DNA sequences employed in this study were primarily obtained from the EPDnew (Eukaryotic Promoter Database) database, which contains experimentally verified eukaryotic promoter sequences [20]. The EPDnew database includes 29,598 experimentally validated promoter sequences corresponding to 16,455 human genes. In accordance with standard practice in the literature, promoter regions for Homo sapiens were extracted as 600-bp sequences spanning the −499 to +100 nucleotide range relative to the transcription start site (TSS). This window is considered to be enriched for transcription factor binding sites (TFBS) and core promoter motifs, and is frequently preferred for increasing promoter identification success [21].

Since EPDnew provides only positive instances (i.e., promoters), a systematic approach was adopted to generate the negative (non-promoter) class. To achieve this, the entire human reference genome (hg38) was utilized, and non-promoter sequences were generated by randomly sampling segments of equal length (600 bp) from genomic regions that do not overlap with any

annotated promoter loci. All sampled regions were cross-checked to ensure no overlap with EPDnew-annotated promoters, and segments containing ambiguous bases ('N') were excluded. The creation of these negative samples is in line with widely accepted indirect identification strategies in the literature, given the absence of direct annotation for non-promoter sequences. Typical criteria include (i) intergenic or intronic regions distant from TSS, (ii) segments lacking transcriptional activity in transcriptome analyses (such as RNA-seq), and (iii) epigenetically repressed or transcriptionally inactive regions. This approach is widely accepted as a standard method, particularly in studies that utilize experimentally annotated databases such as EPDnew [22].

A balanced binary classification dataset was thus created by ensuring an equal number of positive (promoter) and negative (non-promoter) samples. All DNA sequences were converted to uppercase and filtered to remove ambiguous nucleotides prior to further processing.

Furthermore, the 'N' character, which is frequently observed in DNA sequences, signifies nucleotides with low sequencing reliability or those of unknown origin. Given the inability of the model to directly process k-mers containing these characters, there is a possibility that the representativeness of the data may be compromised. In the extant literature, this situation is addressed in two ways: (i) the application of filters to sequences containing 'N' [23], or (ii) representing k-mers containing 'N' with special vector representations (e.g., UNK vector). In this study, the second method was selected to prevent information loss. All k-mers that are not present in the DNA2Vec dictionary or contain the character 'N' are represented by a fixed-defined UNK (unknown) vector instead of the zero vector. This strategy aligns with approaches proposed in advanced models such as DNABERT, which aim to preserve contextual representations of out-of-vocabulary (OOV) tokens [10],[24].

Preprocessing and k-mer Embedding

Prior to the implementation of deep learning models, raw DNA sequences underwent a series of preprocessing steps to ensure their integrity and quality. Initially, all sequences obtained in FASTA format were converted to uppercase and cleared of space characters. This process was implemented to ensure the integrity of the sequence structure and to prevent character-based separations.

Subsequently, each DNA sequence was segmented into *k*-mer subsequences of a fixed length with overlapping segments. In this study, length *k*-mers of the form k=5 were found to be optimal in accordance with the DNA2Vec model. To illustrate, the sequence "ACGTGTC" was segmented into consecutive *k*-mers, such as ACGTG, CGTGT, and GTGTC. This process was executed from the sequence's inception to its conclusion by employing the sliding window method.

Subsequent to *k*-mer decomposition, each *k*-mer sequence was represented by vectors that had been pre-trained by DNA2Vec. The development of the DNA2Vec model was motivated by the Word2Vec algorithm, a foundational approach in the domain of natural language processing. The DNA2Vec model generates contextual dense vector representations for DNA sequences, thereby facilitating the analysis and interpretation of genetic data [8]. In this context, each *k*-mer is mapped to a fixed-size vector, and a DNA sequence is transformed into a fixed-size input matrix by sequential concatenation of the vectors.

In instances where *k*-mers are not incorporated within the DNA2Vec lexicon or contain the character 'N', the zero vector is not employed in lieu of the pertinent *k*-mer. Instead, a fixed-defined UNK (UNKnown) vector is employed to minimize information loss. This strategy draws from out-of-vocabulary (OOV) token representation methods, which have found extensive application in representation learning based on natural language processing. A similar approach has been employed in contextual models such as DNABERT [10, 24]. Consequently, the incorporation of all *k*-mer sequences results in each DNA sequence being presented to the model as a fixed-size matrix. This structural transformation ensures that the sequence context information is preserved and that the model can learn this information.

Vector Construction and Input Representation

Following the *k*-mer-based embedding process, each DNA sequence was represented by a fixed-size vector array. Consequently, each DNA sequence was generated by integrating the DNA2Vec vector equivalents of fixed-length *k*-mers, which were arranged in a sequential manner. Consequently, each sample was transformed into a two-dimensional tensor (matrix) and converted to a format compatible with the model's requirements.

In this study, each DNA sequence belonging to the *Homo sapiens* genome was structured to cover nucleotides between −499 and +300, with the TSS as the center. Consequently, the length of each sequence was fixed at 800 base pairs (bp). These fixed-length sequences were separated into *k*-mers by 5-mer segmentation, and each *k*-mer was represented by vectors pre-trained by DNA2Vec [8]. Consequently, the representation of each sequence was expressed by a fixed-size matrix consisting of the product of the number of *k*-mers (n) and the embedding size (d).

The model inputs are structured as `X.shape = (number of samples, number of k-mers, vector size)`. These fixed structures enable the model to process all samples of the same size and to learn sequence dependencies with stability. Given the fixed sequence lengths, the application of additional padding is not necessary. Given that all sequences contain an equal number of *k*-mers, there is no variation in the input sizes.

Furthermore, *k*-mers that do not have a counterpart in the DNA2Vec dictionary or contain the character 'N' are represented not by a zero vector, but by a fixed-defined UNK vector. This strategy is compatible with methods that

are based on the purpose of preserving the contextual representations of out-of-vocabulary (OOV) tokens. Such methods have been proposed in models such as DNABERT.

In our implementation, the UNK vector is initialized randomly and treated as a trainable embedding, meaning it is updated during backpropagation along with other model parameters. This allows the model to learn an optimal contextual representation for ambiguous or rare k-mers, rather than relying on a static or zero embedding.

Model Architecture and Mathematical Description

The classification model developed in this study is schematically presented in Figure 1. The model is predicated on a multilayer DNN architecture that accepts as input fixed-size (100-dimensional) DNA2Vec embedding vectors representing DNA sequences belonging to *Homo sapiens*. Each DNA sequence is represented in a fixed format of 600 base pairs (bp), and these sequences are represented with DNA2Vec vectors after being divided into *k*-mer segments and then reduced to the average representation. Consequently, each example is presented to the model as a fixed-size vector, without preserving the sequential context information.

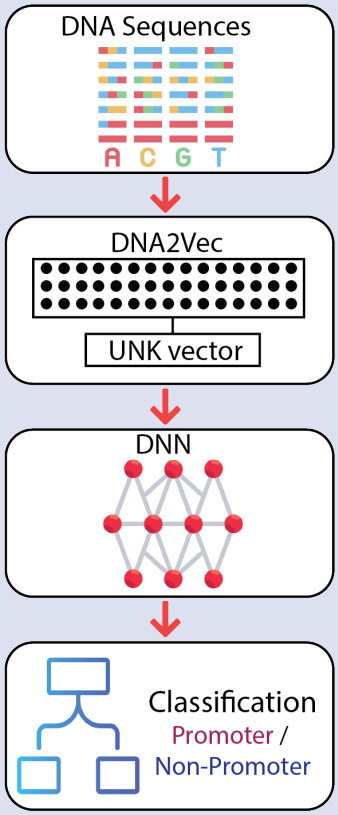

Figure 1. Schematic representation of the proposed DNA2Vec based DNN model architecture.

The model's architecture comprises three hidden fully connected layers, with 256, 128, and 64 neurons, respectively. Each layer utilizes the GELU (Gaussian Error Linear Unit) activation function, followed by Layer Normalization and Dropout (ratios ranging from 0.1 to 0.5). In comparison to the conventional ReLU function, the GELU function facilitates a more seamless and differentiable transition, thereby enhancing the stability of the learning process in settings with limited data samples. The employment of Layer Normalization and Dropout in conjunction serves to mitigate the risk of overfitting, thereby enhancing the model's generalizability.

The output layer of the model contains a single neuron, and it performs the binary classification task with a sigmoid activation function. The loss function is implemented by binary focal loss, a metric sensitive to the imbalance between classes in the sample. This function, with the parameters γ=2.0 and α=0.25, serves to reduce the impact of easily classified examples and optimize the decision surface of the model by focusing on challenging examples [25].

In the training process, the AdamW optimization algorithm was used, with weight decay set to 1e-4. Unlike the classical Adam optimizer, AdamW decouples weight decay from the learning rate update, enabling more effective generalization [26]. Hyperparameter optimization was performed using random search over the learning rate, dropout rate, and number of epochs. The model yielding the highest validation accuracy was selected, and the final training was performed with these optimal parameters.

Three callback functions are included in the training process to monitor the overfitting and learning performance of the model:

- EarlyStopping (validation loss, patience=5),
- ReduceLROnPlateau (val_loss, patience=3, factor=0.5),
- ModelCheckpoint (saving the best model).

Accuracy, Precision, Recall, F1-Score and *ROC-AUC* metrics are calculated as a result of each experiment and the averages of these metrics are reported comparatively. The developed architecture offers a balanced structure in terms of both computational efficiency and classification performance; despite the elimination of sequential context information, it provides high accuracy classification thanks to the power of representations.

The operation in each hidden layer of the model consists of linear transformation, GELU activation, layer normalization and dropout operations. This structure increases the generalizability of the model with both nonlinear transformations and regularization mechanisms in the learning process. The general mathematical expression of this operation can be defined as follows:

$$h^{(l)} = Dropout(LayerNorm(GELU(W^{(l)}h^{(l-1)} + b^{(l)})))$$

Here $h^{(l)}$ represents the activation vector in the *l*-th layer; $W^{(l)}$ and $b^{(l)}$ represent the weight and bias parameters, respectively. The GELU (Gaussian Error Linear Unit) activation function increases the learning capacity of the model by providing a smoother activation instead of linear ReLU. Binary classification was performed in the

Table 1. Classification performance comparison of different optimization and activation combinations

Model Configuration	Accuracy	Precision	Recall	F1 Score	ROC-AUC
Adam + GELU	0.7842	0.7935	0.7684	0.7807	0.8706
Adam + LeakyReLU	0.7842	0.7647	0.8211	0.7919	0.8509
Adam + LeakyReLU + Dropout(0.3)	0.7684	0.7476	0.8105	0.7778	0.8592
Adam + LeakyReLU (α = 0.01)	0.7737	0.7500	0.8211	0.7839	0.8684
AdamW + GELU + Dropout(0.2)	0.8000	0.8000	0.8000	0.8000	0.8635
AdamW + GELU (without dropout layer)	0.8000	0.8065	0.7895	0.7979	0.8577
AdamW + GELU (stratified sampling)	0.8456	0.8731	0.8088	0.8397	0.9293
AdamW + GELU (Random search)	0.8503	0.8786	0.8128	0.8444	0.9376

output layer with a sigmoid activation function. As a loss function, Binary Focal Loss, which is sensitive to class imbalance, was preferred. This function aims to focus on more difficult examples by reducing the effect of easily classified examples. The Focal Loss formulation is as follows:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t)$$

Here, α_t represents the class weight and γ represents the focus parameter. This structure is suggested as an effective strategy to improve the classification performance, especially in imbalanced datasets.

Results

Model Optimization Experiments

Hyperparameter experiments were performed on various DNN architectures with different activation functions, optimization algorithms and dropout rates to improve the overall performance of the model. These comparisons were made under fixed input vector (DNA2Vec 5-mer, 100-dimensional) and fixed number of layers, by only changing optimization strategies and activation types.

Table 1 summarizes the accuracy, precision, recall, F1 score and ROC-AUC values for each alternative configuration.

The LeakyReLU and GELU activation functions implemented under the Adam algorithm produced analogous results in terms of overall accuracy and ROC-AUC. LeakyReLU demonstrated efficacy in achieving high scores, particularly in the recall metric. However, this approach resulted in a relative decrease in precision value. In contrast, the GELU function demonstrated a more balanced precision-recall distribution.

The employment of dropout layers proved beneficial in mitigating the over-learning of the model; however, it resulted in minor reductions in the F1 score in certain configurations. The Dropout (0.3) configuration implemented with LeakyReLU exhibited inferior performance in terms of accuracy when compared to the other models, despite the high recall value.

The models that demonstrated the highest accuracy (85.03%), F1 score (0.8444), and ROC-AUC value (0.9376) employed the AdamW optimization algorithm and GELU activation with random search optimization. The observation that comparable outcomes were attained under both constant and variable dropout rates lends further credence to the hypothesis that this architectural design possesses a notable capacity for generalization.

Confusion Matrix and Class-Based Metrics

As a result of the hyperparameter experiments, the model that provided the highest overall success achieved the following performance metrics on the test set:

Table 2. Evaluation metrics on the test set (for the best model)

Meric	Value
Accuracy	0.8503
Precision	0.8786
Recall	0.8128
F1 Score	0.8444
ROC-AUC	0.9376

In order to analyze the class-based performance of the model in more detail, a confusion matrix evaluation was performed. The matrix presented in Figure 2 below shows the prediction accuracy of the model on positive (promoter) and negative (non-promoter) classes.

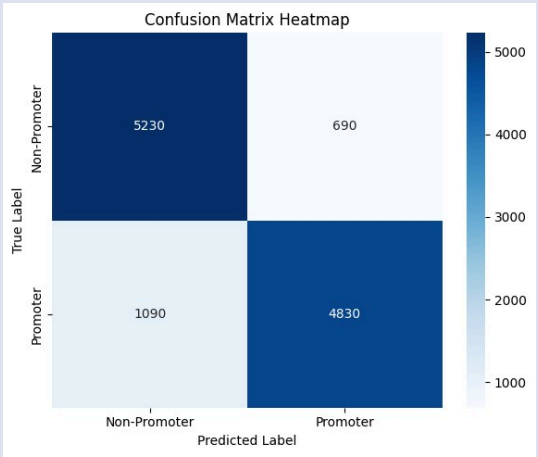

Figure 2. Confusion matrix output of the best model on the test set. Positive class: promoter, Negative class: non-promoter.

When the matrix is examined, it is seen that the model can successfully identify the positive class (promoter sequences) and provide high sensitivity (recall). However, it is observed that it works with a low error rate in the negative class. This structure shows that the model can make decisions without experiencing the problem of unbalanced class learning.

ROC Curve and AUC Analysis

The overall classification performance of the model was evaluated visually with the ROC (Receiver Operating Characteristic) curve, which is a threshold-independent

metric. The ROC curve seen in the graph in Figure 3 reflects the balance between the sensitivity (Recall) and specificity (1 − False Positive Rate) of the model at different decision thresholds.

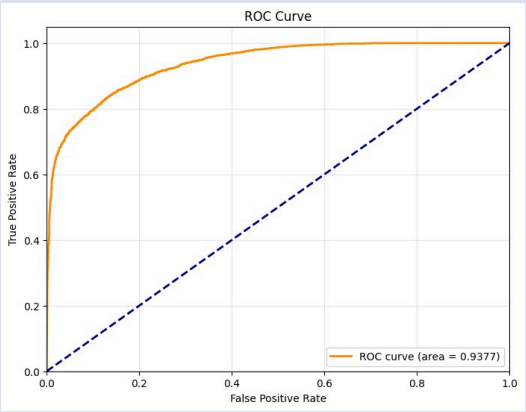

Figure 3. ROC curve obtained on the test set

The area under the curve, i.e. the ROC-AUC score, was calculated as 0.9376. This value shows that the model has a strong ability to distinguish between positive (promoter) and negative (non-promoter) classes. The fact that the ROC curve is above the ideal performance line, y = x, and close to the upper left corner, reveals that the model provides high sensitivity while keeping the false positive rate low. This analysis, when evaluated together with the confusion matrix and accuracy-based metrics, supports the general classification ability of the developed model both quantitatively and visually.

Although the source code is not publicly available, it has been documented in detail within the manuscript. The implementation can be shared with interested researchers upon reasonable request for academic and non-commercial use.

Discussion

This study introduces a hybrid classification strategy for distinguishing promoter and non-promoter DNA sequences within the *Homo sapiens* genome by integrating DNA2Vec-based embedding representations with a DNN classifier. The inclusion of a predefined UNK vector for handling unknown nucleotides contributed significantly to preserving contextual semantics during embedding and improved the model's generalization capabilities. As shown in Table 2, the proposed model achieved a test accuracy of 0.8503, F1-score of 0.8444, and ROC-AUC of 0.9376—indicating a well-balanced and acceptable performance level.

During the model tuning phase, multiple variants were tested by changing the activation function (GELU vs. LeakyReLU), optimizer (Adam vs. AdamW), and dropout ratios. These results, presented in Table 1, demonstrate that the best-performing architecture combined GELU activation with the AdamW optimizer without dropout regularization. This combination led to stable convergence and superior generalization compared to other configurations and yielded the highest metrics: 85.03% accuracy, 0.8444 F1-score, and 0.9376 ROC-AUC on the test set.

Importantly, the proposed model offers a simplified yet effective architecture that contrasts with the increasing trend of complex models such as CNNs, BiLSTMs, GNNs, and Transformers. While our model does not outperform cutting-edge architectures in absolute accuracy (e.g., GraphPro [98.1%], ACNN-BLSTM [97.5%]), its lightweight design, interpretability, and training efficiency make it a compelling candidate for practical deployment in genomic annotation pipelines.

To provide context for the performance of the proposed system, Table 3 summarizes results from relevant literature and benchmarking studies. While deep architectures trained on bacterial or species-specific datasets often report high scores, models trained on human genome data (such as ours and [27]) typically yield more conservative metrics due to sequence diversity and biological complexity. Within this framework, our model stands out by offering a favorable trade-off between architectural simplicity and classification performance. It is important to note that the models listed in Table 3 were trained and evaluated on various datasets, including bacterial (e.g., E. coli) and human (Homo sapiens) genomes. While the proposed model specifically focuses on the human genome, some referenced models were optimized for bacterial sigma factor classification. Therefore, although the presented accuracy values provide a broad performance perspective, differences in species and dataset characteristics should be taken into account when interpreting comparative results.

Moreover, the confusion matrix and ROC curve (Figures 2 and 3) highlight the model's sensitivity and its ability to maintain performance even in the presence of ambiguous sequence data. This further supports the viability of embedding-based, non-sequential modeling for DNA classification tasks.

Conclusion

This work presents a DNN-based classification approach enhanced by DNA2Vec embeddings and a predefined UNK vector to handle ambiguous bases. The model achieves strong and balanced performance in promoter prediction tasks on human DNA sequences, showing robustness and interpretability without relying on complex sequential architectures.

Compared to more elaborate deep learning models, our method emphasizes clarity, efficiency, and ease of training, which makes it suitable for applications in scalable genomics and real-world deployment. Nevertheless, limitations remain—most notably the lack of positional encoding and dependence on average-pooled embeddings, which may hinder the detection of long-range dependencies.

Future directions should explore hybrid architectures incorporating attention mechanisms or Transformer-based embeddings (e.g., DNABERT), and cross-species validation on broader datasets to assess generalizability. Furthermore, the differentiation of functional promoter subtypes could enhance biological insight and refine classification capabilities.

Table 3. Comparative benchmarking of the proposed model against existing literature.

Study	Accuracy	F1-Score	AUC	Notes
PromoterLCNN [11]	94.10%	0.94	–	Bacterial dataset
iPromoter-BnCNN [12]	91.5–95.8%	0.91	0.94	*E. coli* dataset
GSCNN [13]	97.43%	0.97	0.99	Sigma54-specific
ML + *k*-mer + PCA [27]	93.30%	0.93	0.95	*H. sapiens*
DNA2Vec + CNN [15]	94.70%	0.94	0.96	Simple hybrid
DNA2Vec + CNN + BiLSTM [16]	96.20%	0.96	0.97	Protein–DNA binding
GraphPro [14]	98.10%	0.98	0.99	Multiple promoter types
Proposed Model	**85.03%**	**0.84**	**0.94**	*DNA2Vec + UNK + DNN*

References

[1] M. W. Libbrecht and W. S. Noble, "Machine learning applications in genetics and genomics," Nat Rev Genet, vol. 16, no. 6, pp. 321–332, May 2015, doi: 10.1038/NRG3920;SUBJMETA=114,1305,208,212,2415,631;KWRD=GENOMICS,MACHINE+LEARNING,STATISTICAL+METHODS.

[2] S. T. Smale and J. T. Kadonaga, "The RNA polymerase II core promoter," Annu Rev Biochem, vol. 72, no. Volume 72, 2003, pp. 449–479, Jul. 2003, doi: 10.1146/ANNUREV.BIOCHEM.72.121801.161520/CITE/REFWORKS.

[3] P. Carninci et al., "Genome-wide analysis of mammalian promoter architecture and evolution," Nat Genet, vol. 38, no. 6, pp. 626–635, Jun. 2006, doi: 10.1038/NG1789,.

[4] J. W. Fickett and C. shung Tung, "Assessment of protein coding measures," Nucleic Acids Res, vol. 20, no. 24, pp. 6441–6450, Dec. 1992, doi: 10.1093/NAR/20.24.6441.

[5] P. Ng, "Dna2vec: Consistent vector representations of variable-length *k*-mers," arXiv preprint arXiv:1701.06279, Jan. 2017, Accessed: Jun. 03, 2025. [Online]. Available: https://arxiv.org/pdf/1701.06279

[6] L. Chen, C. Cai, V. Chen, and X. Lu, "Learning a hierarchical representation of the yeast transcriptomic machinery using an autoencoder model," BMC Bioinformatics, vol. 17, no. 1, pp. 97–107, Jan. 2016, doi: 10.1186/S12859-015-0852-1/FIGURES/6.

[7] D. R. Kelley, J. Snoek, and J. L. Rinn, "Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks," Genome Res, vol. 26, no. 7, pp. 990–999, Jul. 2016, doi: 10.1101/GR.200535.115.

[8] H. Zeng, M. D. Edwards, G. Liu, and D. K. Gifford, "Convolutional neural network architectures for predicting DNA–protein binding," Bioinformatics, vol. 32, no. 12, pp. i121–i127, Jun. 2016, doi: 10.1093/BIOINFORMATICS/BTW255.

[9] D. H. A. Mai, L. T. Nguyen, and E. Y. Lee, "TSSNote-CyaPromBERT: Development of an integrated platform for highly accurate promoter prediction and visualization of Synechococcus sp. and Synechocystis sp. through a state-of-the-art natural language processing model BERT," Front Genet, vol. 13, p. 1067562, Nov. 2022, doi: 10.3389/FGENE.2022.1067562/BIBTEX.

[10] Y. Ji, Z. Zhou, H. Liu, and R. V. Davuluri, "DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome," Bioinformatics, vol. 37, no. 15, pp. 2112–2120, Aug. 2021, doi: 10.1093/BIOINFORMATICS/BTAB083.

[11] D. Hernández, N. Jara, M. Araya, R. E. Durán, and C. Buil-Aranda, "PromoterLCNN: A Light CNN-Based Promoter Prediction and Classification Model," Genes (Basel), vol. 13, no. 7, Jul. 2022, doi: 10.3390/GENES13071126,.

[12] R. Amin et al., "iPromoter-BnCNN: a novel branched CNN-based predictor for identifying and classifying sigma promoters," Bioinformatics, vol. 36, no. 19, pp. 4869–4875, Dec. 2020, doi: 10.1093/BIOINFORMATICS/BTAA609.

[13] S. Sasikala and T. Ratha Jeyalakshmi, "GSCNN: a composition of CNN and Gibb Sampling computational strategy for predicting promoter in bacterial genomes," International Journal of Information Technology (Singapore), vol. 13, no. 2, pp. 493–499, Apr. 2021, doi: 10.1007/S41870-020-00565-Y.

[14] Q. Zhang, Y. Wei, and L. Liu, "GraphPro: An interpretable graph neural network-based model for identifying promoters in multiple species," Comput Biol Med, vol. 180, p. 108974, Sep. 2024, doi: 10.1016/J.COMPBIOMED.2024.108974.

[15] U. M. Akkaya and H. Kalkan, "Classification of DNA Sequences with *k*-mers Based Vector Representations," Proceedings - 2021 Innovations in Intelligent Systems and Applications Conference, ASYU 2021, 2021, doi: 10.1109/ASYU52992.2021.9599084.

[16] L. Deng, H. Wu, and H. Liu, "D2VCB: A Hybrid Deep Neural Network for the Prediction of in-vivo Protein-DNA Binding from Combined DNA Sequence," Proceedings - 2019 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2019, pp. 74–77, Nov. 2019, doi: 10.1109/BIBM47256.2019.8983051.

[17] V. Rajendran, H. Anandaram, S. Sachin Kumar, K. P. Soman, and S. Dhivya, "A Comparative Analysis of Machine Learning and Deep Learning Approaches for Circular RNA Classification," Proceedings of International Conference on Contemporary Computing and Informatics, IC3I 2023, pp. 1026–1034, 2023, doi: 10.1109/IC3I59117.2023.10397741.

[18] S. Ganesan, S. Sachin Kumar, and K. P. Soman, "Biological Sequence Embedding Based Classification for MERS and SARS," Communications in Computer and Information Science, vol. 1440 CCIS, pp. 475–487, 2021, doi: 10.1007/978-3-030-81462-5_43/FIGURES/4.

[19] L. Shi and B. Chen, "LSHvec: A vector representation of DNA sequences using locality sensitive hashing and FastText word embeddings," Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics, BCB 2021, Jan. 2021, doi: 10.1145/3459930.3469521.

[20] R. Dreos, G. Ambrosini, R. C. Périer, and P. Bucher, "EPD and EPDnew, high-quality promoter resources in the next-generation sequencing era," Nucleic Acids Res, vol. 41, no. Database issue, p. D157, Jan. 2012, doi: 10.1093/NAR/GKS1233.

[21] P. Bucher, "Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences," J Mol Biol, vol. 212, no. 4,

pp. 563–578, Apr. 1990, doi: 10.1016/0022-2836(90)90223-9.

[22] P. J. Wei, Z. Z. Pang, L. J. Jiang, D. Y. Tan, Y. Sen Su, and C. H. Zheng, "Promoter prediction in nannochloropsis based on densely connected convolutional neural networks," Methods, vol. 204, pp. 38–46, Aug. 2022, doi: 10.1016/J.YMETH.2022.03.017.

[23] B. Sahu et al., "Sequence determinants of human gene regulatory elements," Nature Genetics 2022 54:3, vol. 54, no. 3, pp. 283–294, Feb. 2022, doi: 10.1038/s41588-021-01009-4.

[24] E. C. Alley, G. Khimulya, S. Biswas, M. AlQuraishi, and G. M. Church, "Unified rational protein engineering with sequence-based deep representation learning," Nat Methods, vol. 16, no. 12, pp. 1315–1322, Dec. 2019, doi: 10.1038/S41592-019-0598-1;SUBJMETA=114,1305,338,469,552,61,631;KWRD=MACHINE+LEARNING,PROTEIN+DESIGN,SYNTHETIC+BIOLOGY.

[25] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," IEEE Trans Pattern Anal Mach Intell, vol. 42, no. 2, pp. 318–327, Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.

[26] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," 7th International Conference on Learning Representations, ICLR 2019, Nov. 2017, Accessed: Jun. 04, 2025. [Online]. Available: https://arxiv.org/pdf/1711.05101

[27] M. M. Uddin, J. Shiddike, A. Ahmed, and T. Ahsan, "Promoter Prediction in DNA Classification Using Machine Learning Algorithms," Proceedings - 2024 3rd International Conference on Sentiment Analysis and Deep Learning, ICSADL 2024, pp. 254–260, 2024, doi: 10.1109/ICSADL61749.2024.00047.