Exploring NLP Approaches for Classifying Promoter Regions in *E. coli* DNA Sequences

Rebeka Maneva, Konstantin Lozhankoski
Faculty of Computer Science and Engineering,
“Ss. Cyril and Methodius” University in Skopje, North Macedonia

***Abstract*—Promoter recognition in bacterial genomes is difficult for computational approaches, which limits our understanding of gene regulation. Here we treat promoter detection as a binary sequence classification problem in Escherichia coli, comparing traditional machine learning (k-mer features with XGBoost) against neural architectures (CNN, BiLSTM and hybrid CNN+BiLSTM models) and a transformer-based approach (DNABERT). Using fixed sequence windows of 100 bp and 200 bp, we assess performance through ROC-AUC and PR-AUC alongside standard metrics. CNNs show the strongest performance on the RegulonDB dataset in our evaluation, while BiLSTM and DNABERT appear more robust when tested on the UCI dataset. Genome-wide scanning of E. coli K-12 MG1655 highlights challenges in peak-based promoter detection. Code: https://github.com/RebekaManeva/NLP-Ecoli-Promoter-Classification.**

***Index Terms*—promoter prediction, DNA sequence classification, NLP for genomics, CNN, BiLSTM, DNABERT, XGBoost, genome scanning**

I. Introduction

Promoter prediction in bacteria relies on detecting sequence motifs, but accurate interpretation requires both local signals and positional context. This makes the problem similar to language modeling, where meaning depends on tokens and their arrangement.

Although NLP inspired architectures, convolutional, recurrent and transformer based, have been adapted to genomic sequences, we do not know if increasing representational complexity improves performance for compact, motif driven bacterial promoters such as those in E. coli.

We conduct a controlled comparison under uniform preprocessing and evaluation protocols on curated E. coli datasets, followed by genome-wide scanning, to determine when additional contextual modeling meaningfully improves bacterial promoter recognition.

II. Related Work

Early promoter studies identified conserved motifs and positional patterns [2]. With machine learning, researchers began treating DNA as structured sequence data with contextual dependencies; NLP inspired representations have proven useful for biological sequence interactions [1]. Deep learning approaches have advanced promoter prediction through direct learning of sequence motifs and dependencies. CNNs and LSTMs, often in combination, capture local patterns alongside longer range regulatory effects [6]. Transformer models pretrained on DNA sequences via masked language modeling (notably DNABERT [9]) have established competitive benchmarks by treating nucleotide sequences analogously to natural language. Subsequent promoter specific variants like BERT-Promoter and multi scale adaptations have refined these architectures through targeted pretraining and interpretability methods [3], [4].

Direct comparison across studies is difficult due to varying datasets, sequence window sizes and evaluation protocols. We address this by benchmarking multiple architectures under uniform conditions on *E. coli* data. Beyond standard classification metrics, we perform genome-wide scanning to assess false-positive rates and evaluate practical promoter detection through peak identification.

III. Data

We use three complementary sources focused on *E. coli*: RegulonDB promoter annotations for training and in-domain testing, the UCI dataset (53 positives, 53 negatives) for cross-dataset evaluation and the *E. coli* K-12 MG1655 reference genome (U00096.3) for genome-wide scanning.

IV. Preprocessing

All sequences are uppercased and ambiguous symbols are removed during preprocessing. We evaluate two fixed input lengths, $L \in \{100, 200\}$ bp. Positive training windows are centered on annotated TSS where available, while negatives are sampled from non-promoter genomic regions. Neural models (CNN, BiLSTM, hybrids) use one-hot encoding to represent sequences as $(L, 4)$ tensors. The transformer model (DNABERT) instead requires tokenization into overlapping 6-mers following the original protocol [9].

V. Models

A. XGBoost baseline

We train XGBoost on flattened one-hot encoded windows as a lightweight baseline. This approach tests whether shallow statistical features—nucleotide composition and local patterns—suffice for promoter detection without explicit sequential modeling.

B. CNN, BiLSTM and CNN+BiLSTM

The 1D CNN detects motif-like patterns through local receptive fields, appropriate for bacterial promoters with their short conserved signals [2]. The BiLSTM captures sequential dependencies bidirectionally. The hybrid CNN+BiLSTM architecture first extracts motif activations via convolution, then integrates them with recurrent layers to model longer-range context [6].

C. DNABERT

DNABERT is a transformer pretrained on DNA sequences through masked language modeling with k-mer tokenization [9]. We fine-tune it for binary promoter classification on the full RegulonDB training set.

D. Implementation Details

Neural models were implemented in TensorFlow/Keras using Adam optimizer, batch size 32 and early stopping (patience 4, max 15 epochs). The CNN uses two Conv1D layers (64, 128 filters, kernel size 5), the BiLSTM uses two bidirectional LSTM layers (64, 32 units) and the hybrid combines both. XGBoost used 300 estimators, max depth 6, learning rate 0.05. DNABERT fine-tuning used the pretrained `zhihan1996/DNA_bert_6` model with HuggingFace transformers.

VI. Evaluation Protocol

We use stratified 80/20 train-test splits and report Accuracy, Precision, Recall, F1, ROC-AUC and PR-AUC. We prioritize PR-AUC given the class imbalance and the high cost of false positives in promoter prediction, so precision-recall curves reveal how model confidence thresholds affect the precision-recall tradeoff.

VII. Results

A. In-domain evaluation on RegulonDB

Table I shows performance on RegulonDB for both window sizes. In our experiments, the CNN shows the strongest overall performance, with high ROC-AUC and PR-AUC at 100 bp and the highest scores at 200 bp. BiLSTM performance is comparable at 100 bp but drops considerably at 200 bp. DNABERT shows competitive performance but does not substantially outperform the simpler CNN architecture.

B. ROC and Precision–Recall curves

Fig. 1 compares ROC and PR curves at both window lengths. The CNN achieves the highest ROC-AUC and PR-AUC on RegulonDB. However, BiLSTM and DNABERT maintain better precision at higher recall thresholds, making them preferable when minimizing false positives is more critical.

C. Cross-dataset evaluation on UCI

Table II reports results on the UCI dataset, which represents a distribution shift from RegulonDB. The performance ranking differs notably: BiLSTM (100 bp) and CNN+BiLSTM (200 bp) show more stable performance in our experiments, which may indicate that recurrent context modeling transfers better than purely convolutional motif detection, though further investigation with larger sample sizes would be needed to confirm this trend. CNN+BiLSTM achieves perfect precision but very low recall on UCI, suggesting overly conservative predictions.

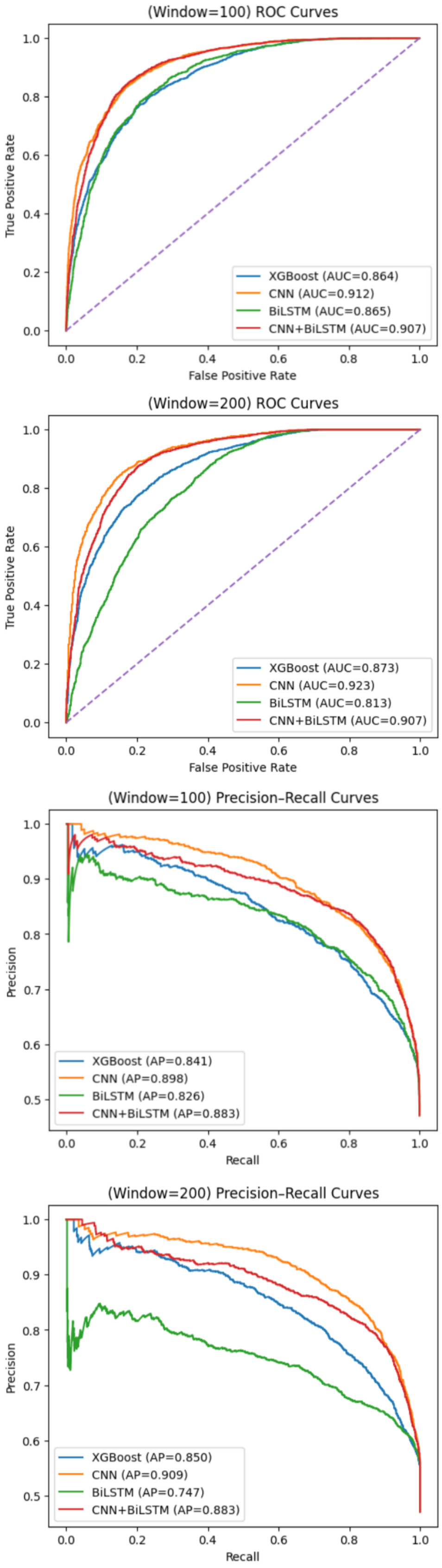

Fig. 1. ROC and precision-recall curves for RegulonDB test data at both window lengths. Top: ROC curves with area under curve (AUC). Bottom: Precision-recall curves with average precision (AP).

TABLE I
Model performance on the RegulonDB test set. (binary promoter classification).

InputLen	Model	Acc	Prec	Rec	F1	ROC-AUC	PR-AUC
100	XGBoost	0.777	0.741	0.808	0.773	0.864	0.841
100	CNN	0.828	0.797	0.852	0.823	0.912	0.898
100	BiLSTM	0.781	0.725	0.860	0.787	0.865	0.826
100	CNN+BiLSTM	0.823	0.846	0.764	0.803	0.907	0.883
100	DNABERT	0.784	0.775	0.764	0.769	0.864	0.837
200	XGBoost	0.783	0.745	0.822	0.781	0.873	0.850
200	CNN	0.835	0.871	0.762	0.813	0.923	0.909
200	BiLSTM	0.726	0.665	0.845	0.744	0.813	0.747
200	CNN+BiLSTM	0.829	0.815	0.824	0.819	0.907	0.883
200	DNABERT	0.758	0.698	0.858	0.770	0.846	0.814

TABLE II
External evaluation on UCI (balanced labels 53/53).

InputLen	Model	Acc	Prec	Rec	F1	ROC-AUC	PR-AUC
100	XGBoost	0.632	0.646	0.585	0.614	0.701	0.683
100	CNN	0.547	0.593	0.302	0.400	0.643	0.622
100	BiLSTM	0.745	0.760	0.717	0.738	0.791	0.745
100	CNN+BiLSTM	0.585	1.000	0.170	0.290	0.823	0.821
100	DNABERT	0.688	0.727	0.603	0.659	0.741	0.759
200	XGBoost	0.604	0.617	0.547	0.580	0.631	0.663
200	CNN	0.632	0.706	0.453	0.552	0.653	0.660
200	BiLSTM	0.613	0.573	0.887	0.696	0.716	0.756
200	CNN+BiLSTM	0.726	0.773	0.642	0.701	0.796	0.788
200	DNABERT	0.716	0.661	0.886	0.758	0.775	0.792

D. Genome-wide scanning on E. coli using BiLSTM

We perform genome-wide scanning by sliding windows across the reference genome and merging high-scoring positions into peaks. Table III shows BiLSTM results under varying thresholds. At 100 bp, higher thresholds reduce false positives while maintaining reasonable recall. At 200 bp, the model achieves near-zero recall, suggesting poor generalization to whole-genome peak detection.

E. Genome-wide scanning on E. coli using DNABERT

We apply the same genome-scanning protocol to DNABERT. Windows of 100 bp slide across the genome at 10 bp intervals, tokenized into overlapping 6-mers as required by the DNABERT architecture. Each window yields a promoter probability score. Because true promoter regions generate high scores across multiple overlapping windows, we merge detections within 200 bp into single peaks. This merging strategy applies identically to both base and fine-tuned DNABERT models.

1) Base vs. fine-tuned DNABERT: The pretrained `zhihan1996/DNA_bert_6` model lacks a classifier head trained for promoter detection, resulting in poorly calibrated probabilities. We fine-tune DNABERT on *E. coli* promoter data, using windows centered on known TSS as positives and windows sampled from non-TSS regions as negatives. This approach follows established methods for adapting pretrained DNA language models to promoter-specific tasks [3], [4]. Fine-tuning yields more stable probability scores genome-wide; predictions no longer collapse when applying stricter thresholds.

2) Peak-based evaluation with tolerance: We evaluate predicted peaks against known TSS locations using a tolerance window. A TSS counts as detected (true positive) if any predicted peak falls within ±tol base pairs, unmatched TSS are false negatives and peaks outside all tolerance windows are false positives. We test tolerances of 200 bp and 300 bp to account for positional uncertainty.

3) Results and tolerance choice: Table III presents genome-scanning results for DNABERT at ±300 bp tolerance. We tested ±200 bp as well and observed the same trends: base DNABERT fails at thresholds of 0.8 and above, while fine-tuned DNABERT maintains stable recall and improves precision as the threshold increases. We report the 300 bp tolerance because the step size and peak-merging strategy introduce positional uncertainty that makes narrow tolerances unrealistic for practical use.

TABLE III
GENOME SCANNING RESULTS AT TOLERANCE ±300 BP. BILSTM USES 100 BP WINDOWS WITH 10 BP STEP. DNABERT USES 100 BP WINDOWS WITH 10 BP STEP; BASE MODEL USES PRETRAINED WEIGHTS, FINE-TUNED MODEL IS TRAINED ON *E. coli* PROMOTER DATA.

Model	**Thr.**	**Peaks**	**TP**	**FP**	**FN**	**Prec.**	**Rec.**	**F1**
BiLSTM	0.40	14342	3378	10964	586	0.236	0.852	0.369
	0.60	11047	3145	7902	819	0.285	0.793	0.419
	0.80	7911	2895	5016	1069	0.366	0.730	0.488
	0.90	4830	2429	2401	1535	0.503	0.613	0.552
Base DNABERT	0.50	21282	3964	17318	0	0.186	1.000	0.314
	0.70	93	49	44	3915	0.527	0.012	0.024
	0.80	0	0	0	3964	0.000	0.000	0.000
	0.90	0	0	0	3964	0.000	0.000	0.000
Fine-tuned DNABERT	0.50	11933	3926	8007	38	0.329	0.990	0.494
	0.70	10086	3893	6193	71	0.386	0.982	0.554
	0.80	8863	3851	5012	113	0.435	0.971	0.600
	0.90	6678	3710	2968	254	0.556	0.936	0.697

F. Summary

CNN models achieve the highest F1 and PR-AUC on RegulonDB at both window sizes. This likely reflects CNN's ability to directly capture short bacterial promoter motifs (-10 and -35 boxes) through convolutional filters, which aligns well with the conserved structure of RegulonDB promoters. BiLSTM and DNABERT show greater robustness on UCI, suggesting that bidirectional context modeling transfers better than pure motif detection when sequence variations increase.

DNABERT's limited advantage despite pretraining is noteworthy. The 6-mer tokenization may fragment biologically meaningful motifs and bacterial promoters may lack the hierarchical structure that benefits transformer models in NLP. For genome scanning, BiLSTM at 100 bp achieves reasonable precision-recall tradeoffs. Base DNABERT requires task-specific fine-tuning. Once fine-tuned, thresholds of 0.8 and above substantially improve precision while maintaining strong recall.

VIII. LIMITATIONS

Several limitations should be considered. First, we use single 80/20 splits without cross-validation or significance testing; performance differences represent observed trends rather than validated conclusions. Second, transformer fine-tuning requirements may have limited hyperparameter exploration for DNABERT compared to lighter architectures.

Our evaluation focuses exclusively on *E. coli*; results may not transfer to other species. Window sizes (100 bp, 200 bp) were chosen from prior work. The scanning tolerance (±300 bp) may be considered arbitrary. We did not perform ablation studies on architectural choices or explore hard-negative sampling strategies. Despite these limitations, our work provides systematic comparison under consistent protocols.

IX. CONCLUSION

Under consistent evaluation settings, CNNs achieve the strongest in domain performance, indicating that local motif detection is highly effective for compact bacterial promoters. Recurrent and transformer models show improved robustness under cross dataset evaluation, but DNABERT does not substantially outperform simpler architectures and requires organism specific fine tuning for reliable genome wide peak detection. Overall, increased representational complexity does not automatically lead to superior practical performance for bacterial promoter prediction.

Future work should extend this comparison across species, incorporate more rigorous cross validation protocols and explore calibration and hardnegative strategies to improve genome wide promoter detection.

REFERENCES

[1] W. Zeng, M. Wu and R. Jiang, "Prediction of enhancer-promoter interactions via natural language processing," *BMC Genomics*, 2018. [Online]. Available: https://link.springer.com/article/10.1186/s12864-018-4459-6

[2] M. E. Mulligan and W. R. McClure, "Analysis of the occurrence of promoter-sites in DNA," *Nucleic Acids Research*, 1986. [Online]. Available: https://academic.oup.com/nar/article/14/1/109/2385379

[3] N. Q. K. Le, Q.-T. Ho, V.-N. Nguyen and J.-S. Chang, "BERT-Promoter: Improved promoter prediction using BERT + SHAP," 2022. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/35863177/

[4] Y. Li *et al.*, "Multi-scale DNABERT model for promoter prediction," 2024. [Online]. Available: https://link.springer.com/article/10.1186/s12915-024-01923-z

[5] L. Moraes *et al.*, "Cap-sProm: Capsule network model for promoter identification," 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S001048252200419X

[6] M. Oubounyt, Z. Louadi, H. Tayara and K. T. Chong, "DeePromoter — CNN+LSTM-based promoter classifier," 2019. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/31024615/

[7] H. Tayara, M. Tahir and K. T. Chong, "iPSW (PseDNC-DL) — Hybrid PseDNC + Deep Learning promoter/strength predictor," 2019. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/31437540/

[8] M. Carlucci *et al.*, "iPromoter-BnCNN — Branched CNN for bacterial promoter subtype prediction," 2020. [Online]. Available: https://academic.oup.com/bioinformatics/article/36/6/1952/5614815

[9] Y. Ji, Z. Zhou, H. Liu and R. V. Davuluri, "DNABERT — Pre-trained bidirectional encoder representations from transformers model for DNA-language in genome," 2021. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/33538820/

[10] H. Dalla-Torre *et al.*, "Nucleotide Transformer: building and evaluating robust foundation models for human genomics," 2024. [Online]. Available: https://www.nature.com/articles/s41592-024-02523-z