

Transformer-Encoders

Aryan Kadiya, Erkan Dadevski

January 13, 2026

Contents

1	Abstract	3
2	Introduction	3
3	Theoretical Background	4
3.1	Data Representation	4
3.2	Evolution of Deep Learning Usage in Sequence Modeling	4
4	Transformers	4
4.1	Architecture	4
4.2	Biological application	5
4.3	DNABERT-2 Architecture	6
4.4	1D-CNN Baseline	6
5	Methodology	6
5.1	Dataset	6
5.2	Preprocessing	7
5.3	Model Architectures and Training	7
5.3.1	DNABERT-2	7
5.3.2	CNN	7
5.4	Overfitting Mitigation Strategies	8
5.5	Evaluation Metrics	8
6	Results	8
6.1	DNABERT-2	8
6.2	CNN	9
6.3	Comparison	9
7	Discussion	9
7.1	Performance Analysis	9
7.2	Model comparison and Computation Efficiency	10
8	AI Tool Acknowledgment	11
9	References	11

List of Figures

1 Abstract

Antimicrobial resistance (AMR) poses a serious threat to global health, necessitating advancements in computational methods for resistance prediction. This study explores the application of encoder-only Transformer models, in this case DNABERT-2, for predicting cefoxitin resistance in *Staphylococcus aureus* from *pbp4* gene sequences. The pre-trained Transformer architecture will be compared to a 1D-CNN baseline, evaluating a dataset of 899 sequences with a strong class imbalance. Both models demonstrated robust classification, with DNABERT-2 achieving an ROC AUC of 0.9887 and the CNN achieving 0.9869 on a test set. Notably the Transformer achieved a decisive 99% Recall for the resistant class, successfully identifying almost all resistant genes. While the CNN proved significantly more computationally efficient, requiring around 71 times fewer parameters, our findings suggest that the Transformer’s strong performance and superior sensitivity provide significant clinical advantages over the CNNs, while they do remain a highly efficient choice for AMR prediction tasks dominated by local sequence features. The study underscores the potential of deep learning in genomics while highlighting practical trade-offs for clinical deployment.

2 Introduction

Antimicrobial Resistance of pathogens (AMR) poses a critical and escalating threat to global public health. Recognized by the World Health Organization (WHO) as a major concern, AMR minimizes the effectiveness of antibiotics, leading to far more severe illnesses, increased mortality rate and higher healthcare needs (Hu et al. 2024). The currently used culture-based antimicrobial susceptibility testing (AST) is time-consuming, requiring 24-72 hours to yield results, making the use of a broader spectrum antibiotic more attractive, favouring the emergence of resistant pathogens (Barbosa et al. 2000, [apparently in](#))[?]. This reinforces the need for faster computational AMR prediction.

Whole-genome sequencing (WGS) offers an option for this, by providing a quick way to assess genetic information, but its speed and accuracy is dependent on advanced computational tools (Su18). While *mecA* is the primary resistance determinant in MRSA, this report focuses on the *pbp4* gene. Mutation in *pbp4* play an accessory role, particularly in borderline-resistant strands where traditional markers may be absent. (Henze 95)

This technical report explores the role of Encoder architectures in the prediction of AMR directly from *pbp4* genomic sequences. As part of the Transformers architectures, Encoders represent a paradigm shift since their introduction in 2017 (Mowbray25). In particular the self-attention characteristic of these Transformer models will be examined in the learning of global biological patterns and long-range mutational dependencies within DNA sequences. This capability is crucial for the identification of resistant mutations and thereby predicting AMR phenotypes. To achieve this, this report will start with an overview of machine learning models for sequence analysis, highlighting the

advancements from traditional models to modern neural networks. Following this, the architectural principles of encoder only Transformer models will be showcased, explaining their key components. Subsequently the practical implementation using a pre-trained encoder model to predict the resistance of cefoxitin in *Staphylococcus aureus* strands, will be described. Finally, the results of various models will be presented with a discussion of encoder models for AMR prediction. [and possible future works](#)

3 Theoretical Background

3.1 Data Representation

To apply machine learning to the *Staphylococcus aureus* gene, biological sequences, i.e. nucleotide sequences (A, C, G, T), must be transformed to numerical representations. Traditional approaches work with the One-Hot Encoding approach, which maps nucleotides to binary vectors or k-mer frequency analysis, which counts the occurrence of short subsequences of length k (Angermüller). However these methods often result in high-dimensional, sparse data which fails to capture the relationships between nucleotides. Modern Deep Learning approaches address this by utilizing Learned Embeddings, where nucleotides are mapped to dense vectors in a continuous space, allowing the model to learn mathematical similarities between biologically similar sequences (Zou et al. 2019).

3.2 Evolution of Deep Learning Usage in Sequence Modeling

Early computational approaches to AMR prediction relied on supervised algorithms like Support Vector machines (SVM) and Random Forests. These models usually require extensive feature engineering and prior domain knowledge regarding specific resistance markers (Yang 2020). Deep Learning introduced architectures capable of automatic feature extraction. Convolutional Neural Networks (CNNs) apply sliding filters to sequence data, but struggle to model dependencies between distant parts of sequences, due to limited receptive fields. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) models, on the other hand process data sequentially, making them theoretically suitable for analysis of genomic sequences. RNNs often fail to retain context over long sequences due to the vanishing gradient problem (Vaswani et al 2017, Hochreiter 98). The limitations on these models, especially with long-range dependencies in DNA, necessitated the development of the Transformer architecture.

4 Transformers

4.1 Architecture

The limitations of sequential processing in RNNs were addressed by the introduction of the Transformer architecture. (Vaswani 2017).

While CNNs excel at detecting local sequence motifs and RNNs process nucleotides sequentially, both architectures struggle with capturing long-range dependencies across genomic sequences. These dependencies are biologically significant, as resistance-conferring mutations in one region of a gene may structurally or functionally depend on distant nucleotides. To address the importance of these long-distance dependencies along with short-distance relationships, the Transformer architecture, introduced by Vaswani et al. (2017), which relies on a self-attention mechanism was used replacing models using recurrence or convolution. (Devlin 2018)(Thomas STructural Bases (noaccess)). The encoder component of the Transformer model processes entire sequences in parallel, allowing each nucleotide to be weighed against all others simultaneously. The attention is calculated as a weighted sum of values V based on the compatibility of a query Q with a key K :

$$\mathbf{Attention}(Q, K, V) = \mathbf{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (1)$$

where Q (query), K (key) as vectors and V (value) as a learned vector are derived from the input embeddings. Word embeddings are typically representations of real-valued vectors, which encode meanings of words.

Because the encoder processes the entire sequence in parallel, there is no order to follow (Weiss2021). DNABERT-2 utilizes Attention with Linear Biases (ALiBi) to retain positional context through distance-based penalties, rather than fixed additive encodings. To manage this, positional encodings are added to the input embeddings to retain information about the relative positions of nucleotides.

The model is typically trained using Masked Language Modeling, where a percentage of the input sequence is hidden and the model learns to reconstruct the missing nucleotides based on the context.

4.2 Biological application

Encoders facilitate Transfer Learning. In a biological context, labeled data is scarce. The provided data set contains around 800 training samples. A deep network trained from scratch in a data set of this size, would lead to overfitting of the model.

To overcome this, a pre-trained, and then fine-tuned approach was chosen. Models such as DNABERT (Ji2021) are pretrained on massive, unlabeled bacterial genome databases, consisting of around 50 billion nucleotides, learning the fundamental properties of DNA. This pre-trained model is then fine-tuned to specific tasks, in this case the prediction of genomic sequences resistant to Cefoxitin.

There are models trained on whole amino acid sequences, called Protein Language Models (PLMs), like ProtBERT, the data provided was in form of raw nucleotide inputs, for which genomic models, such as the chosen DNABERT, are specifically optimized.

4.3 DNABERT-2 Architecture

DNABERT-2 is an encoder-only Transformer using byte-pair encoding (BPE) tokenization, Attention with Linear Biases (ALiBi), and pre-training on diverse genomic data. Unlike the original DNABERT which used fixed k-mer tokenization ($k=6$), DNABERT-2’s BPE tokenization learns subword units directly from the data, potentially better capturing biological motifs of variable length [Zhou et al., 2023].

The tokenization is realised by initializing a vocabulary, e.g. of size 4096, with all individual nucleotides A C G T. During the pre-training, the BPE algorithm matches the most frequently adjacent nucleotide pairs to form new tokens. This continues until a predetermined vocabulary size is reached. The resulting vocabulary can represent both single nucleotides, as well as sequences containing a multitude of nucleotides. This allows the model to capture patterns and gauge their biological importance at various scales. For inference, input sequences are tokenized using the learned merge rules, which breaks unknown sequences into known, frequent subword units.

The model consists of 12 Transformer encoder layers with 12 attention heads per layer and a hidden dimension of 768. Positional encoding is added to the input embeddings to provide sequence order information, using sinusoidal functions that allow the model to generalize to sequence lengths not seen during training. The pretraining objective is a masked language model (MLM), where the model predicts randomly masked nucleotides or tokens from their contextual information.

4.4 1D-CNN Baseline

As a computationally efficient baseline, we implemented a convolutional architecture with learned nucleotide embeddings, multiple 1D convolution filters (kernel size 9), global max pooling, and a classification head. CNNs excel at detecting local patterns and motifs, making them suitable for identifying conserved resistance-associated regions within the pbp4 gene. While lacking the long-range dependency modeling of Transformers, CNNs typically require fewer computational resources and can perform well on tasks dominated by local sequence features.

5 Methodology

5.1 Dataset

The dataset provided contains 899 pbp4 sequences from *S. aureus* clinical isolates. 800 of these were used as the training set, with 80%, 640, of these used for the training of the model, and the remaining 20%, 160, for validation of the model. This splitting of the sequences ensured that both sets maintained a similar distribution of resistant and susceptible strands, allowing for reliable evaluation during fine-tuning. These samples are labeled as either 0, for *susceptible*, or 1, for *resistant* to the antibiotics. The left-over 99 samples are used to test the models. Furthermore, the datasets have a strong class

imbalance, consisting of around 65% resistant strands in the training/validation set and a distribution of 78 resistant to 21 susceptible strands in the test set, corresponding to an imbalance of 79% resistant strands.

The mean sequence length of the provided pbp4 genome sequences is 1290 nucleotides. This exceeds the standard 512-token limit.

5.2 Preprocessing

To prepare data sets for use in machine learning settings, the data has to be processed first. In this particular case for use of nucleotide sequences in transformer models, the sequences were first cleaned up to only include uppercase A C G T symbols, to remove not clearly definable nucleotides.

Since the genomic sequences contain around 1290 nucleotides, which exceed the 512-token limit, the sequences are segmented into overlapping windows of size 510 nucleotides, with a stride of 250 nucleotides. This approach ensured that no critical local motifs were truncated at the sliding window boundaries and that contextual information was preserved across the whole gene strand. It preserves local context while handling sequences longer than the model’s context window, by ensuring.

5.3 Model Architectures and Training

5.3.1 DNABERT-2

The utilized base model is the zhihan1996/DNABERT-2-117M, a model pre-trained on unlabeled bacterial genomes using MLM. The fine-tuning of this model consists of a dense linear layer in the final embedding of [CLS](Classification) token. With a dropout value of 0.1 the model was regularized and the generality of the model was supported. To handle the imbalance of the input a weighted cross-entropy was introduced, with the weights being 0.64 for resistant and 2.24 for susceptible.

The model was fine-tuned for 8 epochs using an effective batch size of 16 and a learning rate of 2×10^{-5} . To ensure training stability and memory efficiency, the AdamW optimizer was used with a weight decay of 0.01, while employing Mixed Precision (FP16), gradient checkpointing and gradient clipping with a maximum norm of 1.0.

5.3.2 CNN

For comparison purposes a simple CNN was also deployed. Consisting of an Embedding layer, a convolution (Conv1D) layer with 256 filters and a kernel of size 9, a rectifier (ReLU), a global max pool layer, a dropout of 0.1 and a linear classifier layer. The same weighted loss function as used in DNABERT-2 was used here as well.

This model was trained for a maximum 10 epochs with a learning rate of 1×10^{-3} ,

a AdamW optimizer with a 0.001 weight decay and an early stopping with a 3-epoch patience, based on validation loss and a batch size of 32.

5.4 Overfitting Mitigation Strategies

Since the dataset only contained 800 samples, multiple techniques were implemented to prevent overfitting. Both models apply a 0.1 dropout rate before the classification layer. The models apply weight decay with 0.01 L2 regularization for DNABERT-2 and 0.001 for CNN. The CNN also employs an early stopping mechanism, which monitors loss with a 3-epoch patience.

To address the imbalance of the training samples, the classes are assigned weights inversely proportional to their frequency in the data. While advanced regularization techniques such as data augmentation or gradient norm clipping could further improve robustness, these strategies are not implemented in the current training pipeline.

The bottom 6 transformer layer are frozen during DNABERT-2 fine-tuning. This prevents its weight from being further updated. Finally the maximum gradient norm is set to 1.0 to prevent exploding gradients.

These strategies collectively maintain a generality of the models, despite the relatively small dataset size and strong class imbalance.

5.5 Evaluation Metrics

Various metrics were computed on validation and test sets to evaluate the trained models. These include Precision, Recall and F1-score, per class and weighted average, accuracy, area under the receiver operating characteristic curve (ROC AUC) and confusion matrices for error analysis.

Recall is critical in a clinical setting, since a false negative could lead to ineffective treatment, while false positives are not as bad, while Precision ensures that susceptible patients are not unnecessarily treated with resisted antibiotics.

The F1-Score, the harmonic mean of precision and recall, to provide a balanced view of model performance.

ROC AUC was utilized as a primary metric since it's insensitive to class imbalance and provides threshold-independent performance assessment.

6 Results

6.1 DNABERT-2

In Table 1 the results for the metrics discussed earlier for the validation and test set are presented. With an achieved accuracy of 0.87 and an ROC AUC of 0.9456 for the validation set and an accuracy of 0.95 and a ROC AUC of 0.9887 for the test set.

Table 1: DNABERT-2 Performance on Validation and Test Sets

Dataset	Class	Precision	Recall	F1-Score	Support
Validation	Susceptible	0.69	0.75	0.72	36
	Resistant	0.93	0.90	0.91	124
	Accuracy			0.87	160
	Macro Avg	0.81	0.83	0.82	160
	Weighted Avg	0.87	0.87	0.87	160
Test	Susceptible	0.94	0.81	0.87	21
	Resistant	0.95	0.99	0.97	78
	Accuracy			0.89	99
	Macro Avg	0.95	0.90	0.92	99
	Weighted Avg	0.95	0.95	0.95	99

6.2 CNN

In Table 2 the results for the test set using the CNN model are presented. With an achieved accuracy of 0.9192 and an ROC AUC of 0.9869.

Table 2: **CNN Baseline Performance on Test Set.** Evaluation on 99 samples.
ROC AUC: 0.9869.

Class	Precision	Recall	F1-Score	Support
Susceptible	0.7826	0.8571	0.8182	21
Resistant	0.9605	0.9359	0.9481	78
Accuracy			0.9192	99
Macro Avg	0.8716	0.8965	0.8831	99
Weighted Avg	0.9228	0.9192	0.9205	99

6.3 Comparison

In Table 3 the results of the important metrics for all the different models are compared.

7 Discussion

7.1 Performance Analysis

Both models achieved strong performance on the test set despite the small dataset size and class imbalance. DNABERT-2 showed excellent discrimination (ROC AUC 0.9887) and very high precision/recall for the resistant class (0.99/0.97), which is clinically important to avoid missing resistant strands. The CNN baseline didn't quite match the

Table 3: **Model Comparison on Test Set (n=99)**. Comparison of the proposed DNABERT-2 model against the CNN baseline. Best values are highlighted in bold.

Metric	DNABERT-2	CNN Baseline
Accuracy	0.8900	0.8716
ROC AUC	0.9771	0.9869
F1-Score (Resistant)	0.96	0.95
F1-Score (Susceptible)	0.87	0.82
Weighted F1	0.94	0.92

Transformer-based model on ROC AUC (0.9887 vs. 0.9869), suggesting while local sequence motifs are important, the Transformer’s self-attention mechanism more effectively captures the full resistant phenotype. Notably, the Transformer achieved a clinically decisive 99% Recall for the resistant class, successfully identifying almost all actually resistant genes whereas the CNN struggled to achieve this with a Recall of 0.9359.

The better performance on the test set compared to validation may be due to the even stronger imbalance in the test set (79% resistant), making the task easier for models that learn to predict the majority class well. This highlights the importance of balanced evaluation sets that reflect real-world class distributions.

7.2 Model comparison and Computation Efficiency

While the CNN baseline remains computationally efficient, DNABERT-2 demonstrated superior performance. This result suggests that the resistance signals within the *pbp4* gene are not solely driven by local sequence motifs—features that Convolutional Neural Networks are inherently optimized to detect, but benefit from modeling long-range genomic dependencies, which the Transformer’s self-attention is facilitating.

From a computational perspective, the architectural complexity of the Transformer model incurs a significant cost. As summarized in Table 4, the CNN baseline demonstrates superior efficiency across all resource metrics. Notably, the CNN requires approximately $71\times$ fewer parameters and significantly less GPU memory than DNABERT-2. These efficiency disparities have critical implication sfor clinical deployment. In resource-constrained environments, the CNN’s low memory footprint might be preferred. Given the Transformer’s superior clinical sensitivity, however, DNABERT-2 offers a significant advantage for precision medicine, where diagnostic accuracy is paramount.

These efficiency disparities have critical implications for clinical deployment. In resource-consstrained environments, such as hospital laboratories lacking high-end GPU clusters, the CNN’s low memory footprint and rapid inference capability provide a tangible advantage for real-time diagnostic decision-making. However, DNABERT-2 may still offer superior potential for generalization if applied to multi-gene tasks where long-range interactions become more prevalent.

Table 4: **Computational Efficiency Comparison.** Metrics were recorded on a single NVIDIA A100 GPU. The CNN baseline offers significant reductions in memory usage and inference latency.

Metric	CNN Baseline	DNABERT-2
Parameter Count	1.8 M	117 M
Memory Requirement	1.2 GB	5.8 GB
Training Time	2.1 hours	3.2 hours
Inference Speed	15 ms/seq	42 ms/seq

8 AI Tool Acknowledgment

Literature research was completed with the assistance of Scholar Labs, the formatting of the results was taken from a combination of Gemini 3 and ChatGPT. [anything more like text assistance, drafting, maybe structure of report? also code? should we write ai helped code?](#)

9 References

impact00 Teresa M. Barbosa, Stuart B. Levy Addison Wesley, Massachusetts, 2nd Edition, 1994.