Seminar Deep Learning in Molecular Biology

Predict AMR for Staphylococcus aureus against Cefoxitin with a Transformer-Encoder Model given the genomic sequence of gene pbp4 as input

Dagh Zeppenfeld
d.zeppenfeld@tu-bs.de
Matriculation Number: 5538469

Phoebe Gwimo
p.gwimo@tu-bs.de
Matriculation Number: 5416712

Project submitted in partial fulfillment of the requirements for the degree of
Master of Science

Technische Universität Braunschweig
January 9, 2025

Contents

1 Introduction

Antimicrobial resistance (AMR) is a pressing global health challenge that undermines the efficacy of antibiotics in treating infectious diseases. Among the pathogens of concern, *Staphylococcus aureus* (*S. aureus*) is a prominent Gram-positive bacterium responsible for infections ranging from minor skin conditions to life-threatening sepsis and pneumonia. The emergence of methicillin-resistant *S. aureus* (MRSA) has intensified the need for rapid and accurate diagnostic tools to predict AMR. Cefoxitin, a β-lactam antibiotic, serves as a widely used surrogate marker for methicillin resistance in *S. aureus*, with resistance primarily driven by genetic variations in penicillin-binding proteins encoded by the *pbp4* gene.

To address this challenge, this report leverages encoder-only transformer models, such as BERT and its variants, to predict AMR in *S. aureus* against cefoxitin using the genomic sequence of the *pbp4* gene. Transformer models, which have revolutionized natural language processing by identifying intricate patterns in sequential data, hold significant promise for genomic data analysis. By treating genomic sequences as analogous to text, these models can capture complex sequence-level features, including motifs, mutations, and interactions, that correlate with antimicrobial resistance.

Building on this potential, we propose a workflow that preprocesses the genomic sequence of the *pbp4* gene, encodes it using transformer-based architectures, and predicts the susceptibility or resistance of *S. aureus* to cefoxitin. This approach aims not only to improve AMR prediction accuracy but also to uncover genomic determinants of resistance. By demonstrating the versatility of transformer models, this report establishes a foundation for their broader application in computational microbiology and genomics.

1.1 Staphylococcus aureus

Discovered in the 1880s [Deurenberg and Stobberingh, 2008], *Staphylococcus aureus* is a pathogenic Gram-positive bacterium. While it often exists as a harmless commensal organism [Foster, 2004], it can act as an opportunistic pathogen, causing infections ranging from minor skin conditions, such as impetigo and boils, to severe, life-threatening illnesses like sepsis, endocarditis, and pneumonia [Minnesota Department of Health, 2010]. The bacterium's ability to adapt and resist treatment, particularly through the emergence of methicillin-resistant strains (MRSA), poses significant challenges to global public health [Wise et al., 1998]. The World Health Organization (WHO) has prioritized the development of novel antibiotics to combat MRSA, emphasizing the need for innovative strategies to preserve future treatment options [World Health Organization, 2024].

S. aureus colonizes the skin, anterior nares (nostrils), and mucous membranes of humans and animals. It is also found in the environment and can persist on surfaces, particularly in healthcare settings where it is a leading cause of hospital-acquired infections. Approximately 20% of individuals are persistently colonized with *S. aureus*, 60% are intermittent carriers, and 20% are non-carriers [Foster, 2004]. Certain populations, such as neonates, the elderly, immunocompromised individuals, and patients with invasive devices, are more vulnerable to *S. aureus* infections. Both hospital-acquired and community-acquired strains present unique epidemiological and clinical challenges [Minnesota Department of Health, 2010].

S. aureus infections are typically treated with antibiotics targeting bacterial functions like cell wall synthesis, protein production, transcription, and DNA replication. However, the rise of antibiotic resistance has led to increased treatment failures with significant human and medical consequences [Vestergaard et al., 2019]. Antimicrobial resistance (AMR) in *S. aureus*, particularly to β-lactam antibiotics such as methicillin and penicillin, is a major concern [Velasco et al., 2005, Da Costa et al., 2018].

The PBP4 gene encodes penicillin-binding protein 4, an enzyme critical for synthesizing and remodeling the bacterial cell wall. β-lactams inhibit bacterial cell wall synthesis by targeting Penicillin-Binding Proteins (PBPs), which facilitate the final steps of cell wall construction. While PBP4 plays a secondary role compared to PBP2a in methicillin resistance [Velasco et al., 2005], it modulates susceptibility to certain β-lactams, including cefoxitin. PBP4 enhances peptidoglycan cross-linking in the cell wall, contributing to bacterial survival under β-lactam exposure, particularly in strains with high-level resistance [Da Costa et al., 2018].

Cefoxitin, a β-lactam antibiotic, is commonly used as a surrogate marker for detecting methicillin resistance in *S. aureus* [Fernandes et al., 2005]. It strongly induces the expression of PBP2a, making methicillin resistance easier to detect [Vestergaard et al., 2019]. Resistance to cefoxitin is assessed through both phenotypic and molecular methods. In disk diffusion assays, a cefoxitin disk is placed on an agar plate inoculated with the bacterium, and the inhibition zone diameter is measured. Minimum inhibitory concentration (MIC) testing provides a quantitative measure of the cefoxitin concentration required to inhibit bacterial growth [Fernandes et al., 2005]. Molecular methods, such as PCR, confirm the presence of the *mecA* or *mecC* genes, which are definitive markers of methicillin resistance [Memmi et al., 2008].

1.2 Transformer Model Architecture

The Transformer model is a neural network architecture introduced by Vaswani [2017] to address sequence transduction tasks. It eliminates the need for recurrent and convolutional layers by utilizing self-attention mechanisms, allowing for parallelized computations.

The Transformer follows a standard encoder-decoder structure, where the encoder processes the input sequence and generates contextualized embeddings, which the decoder uses to produce the target sequence. Both the encoder and decoder consist of multiple layers incorporating self-attention and feed-forward sub-layers, as shown in Figure 1.

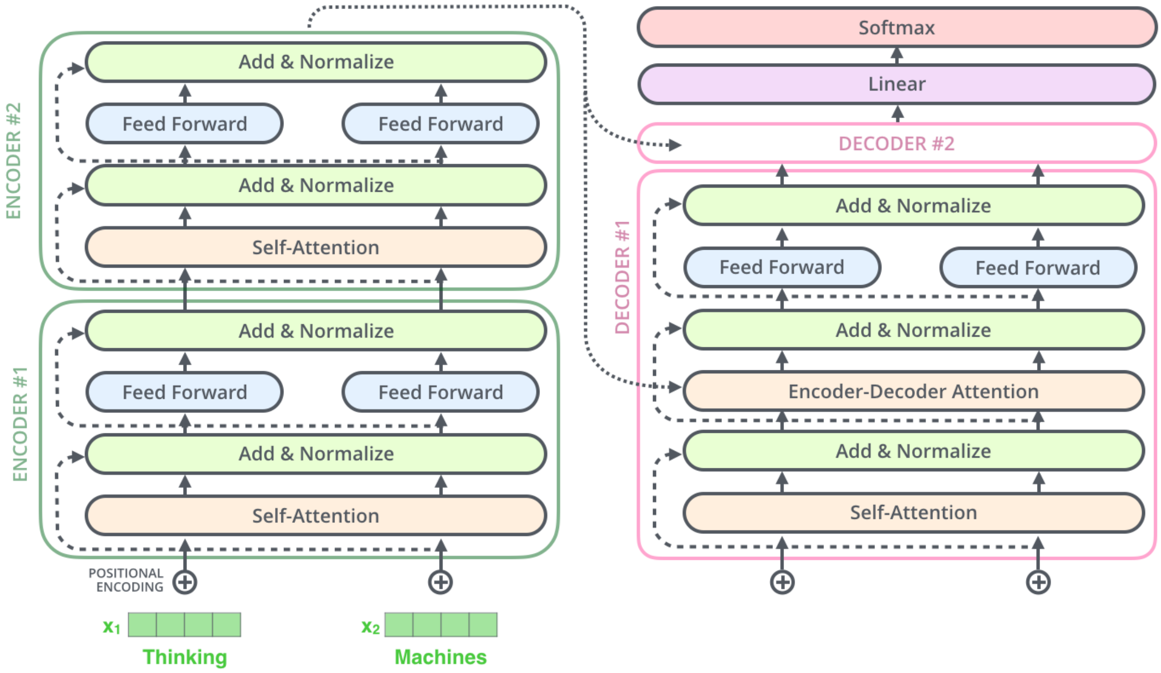

Figure 1: Illustration of the Transformer architecture as described by Alammar [2018]. The model consists of an encoder and a decoder, each composed of multiple layers. This example shows two encoder and two decoder layers.

The encoder tokenizes the input sequence into smaller units called tokens, which are then passed through an embedding layer to produce continuous vector representations. To incorporate sequence order—lacking in the model's architecture—positional encodings are added to the embeddings. These encodings use sine and cosine functions of varying frequencies to capture positional information.

The embeddings are then passed to the multi-head attention layer. Attention, defined in Equation (1), maps a query and a set of key-value pairs to an output:

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

For a sequence of n tokens, the input $X \in \mathbb{R}^{n \times d_{model}}$, where d_{model} is the dimension of the embedding vectors. Three matrices are derived:

- Query (Q): Represents the token being processed.
- Key (K): Encodes information about all tokens in the sequence.
- Value (V): Contains the information to be extracted.

The attention mechanism computes:

- QK^T: Measures the similarity between queries and keys.

- $\frac{1}{\sqrt{d_k}}$: A scaling factor to prevent excessively large dot-product values.
- softmax: Normalizes similarities into probabilities, which serve as attention weights.[1]
- The output: A weighted sum of the values V based on attention weights.

To enhance learning, the Transformer uses multi-head attention, projecting Q, K, and V into h subspaces with dimension d_k. The attention function is computed in parallel for each head, and their outputs are concatenated and linearly transformed:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \tag{2}$$

$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{3}$$

Here, W_i^Q, W_i^K, W_i^V, and W^O are learnable weight matrices.

Residual connections add the original input X to the sub-layer output, followed by normalization. This output is passed to the Feed-Forward Network (FFN), a two-layer neural network with a ReLU activation function. The FFN enriches the token representations with non-linear transformations while retaining the original sequence structure through residual connections.

The encoder stack consists of N identical layers, each applying multi-head attention, FFN, and residual normalization sequentially. The final encoder layer produces contextualized embeddings for the entire sequence, which are passed to the decoder.

1.2.1 Encoder-Only Models

Encoder-only models consist solely of the encoder stack from the Transformer architecture, tailored for tasks requiring input sequence understanding rather than sequence generation. Examples include classification, named entity recognition, and masked language modeling (MLM).

Bidirectional Encoder Representations from Transformers (BERT) is a notable encoder-only model that transformed Natural Language Processing (NLP) [Devlin, 2018]. BERT uses bidirectional self-attention within the encoder stack to generate contextualized embeddings for each token. These embeddings capture rich semantic and contextual information, enabling high performance across a range of NLP tasks.

2 Existing Work

Some work has already been done on the issue of predicting antibiotic resistance (AMR) with transformers. The goal is to make it easier to find working antibiotics for specific pathogens.

To work towards this future, Tharmakulasingam et al. [2023] has developed a new model that can predict antibiotic resistance (AMR) based on publicly available data from government agencies. This data includes over 700 features, such as patient demographic information, antibiotic prescriptions, medical procedures, and previous AMR test results. This approach is different from the current work, which only considers the genetic sequence of PBP4 for prediction. The most proficient model from Tharmakulasingam et al. [2023] achieved a F1-score of 0,47.

In his master's thesis, Olsson [2024] developed a transformers-based AMR prediction for *Escherichia coli* (E. coli) pathogens.Two distinct datasets were utilized, each comprising a unique set of features. The first contained approximately 14 million data points for E. coli, while the second comprised over 370 thousand. The utilization of diverse features and the substantial volume of data distinguish this study from the current one. The research yielded a wide range of results in terms of sensitivity and specificity, with models that were trained on a larger set of data and then fine-tuned for a specific task performing significantly better in making predictions than those without pre-training.

3 Methodology

In this report, a transformer encoder model that has been pre-trained on the domain of genomics is fine-tuned for the purpose of AMR prediction for S. aureus. This method utilizes the generality of larger

[1] Vaswani uses the softmax activation function in the article, however this can be replaced with the Sigmoid activation function for binary classification tasks.

models while serving a very specific purpose in a domain related to the pre-training. DNABERT is selected for this purpose due to its extensive documentation and substantial user base. The models are available in a range of possible configurations, which differ in the tokenizer used (ji2021dnabert).

All randomized actions are executed with a pre-defined seed to ensure the reproducibility of the results reported in this work.

3.1 Introduction to DNABERT model

DNABERT is a specialized adaptation of the BERT model, originally developed for natural language processing (NLP), designed for the genomics domain [Ji et al., 2021, Zhou et al., 2023]. It models DNA sequences as a "language" that captures their contextual semantics and syntactic relationships, enabling it to understand the complexities and patterns inherent in DNA sequences to predict biological functions effectively.

3.1.1 Tokenization

In this model sequences are tokenized into k-mers (overlapping substrings of length k), and a vocabulary of all possible k-mers is constructed. This tokenization approach helps the model capture the local sequence patterns and dependencies that are vital for genomic understanding. The overlapping nature of the k-mers ensures that the model retains contextual continuity, which is particularly important in genomic sequences where the meaning often depends on the surrounding nucleotides. Its robust pre-training and fine-tuning pipeline allows it to excel in diverse sequence analysis tasks, offering unprecedented performance, interpretability, and cross-species applicability. Examples of k-mers of several sizes on genetic sequences is shown in table 1. The DNABERT pre-trained models are available for k-mers of size three, four, five, and six.

Original Sequence	**k-mer Size**	**Tokenized k-mers**
ACGTACGTA	3	ACG, CGT, GTA, TAC, ACG, CGT, GTA
ACGTACGTA	4	ACGT, CGTA, GTAC, TACG, ACGT, CGTA
ACGTACGTA	5	ACGTA, CGTAC, GTACG, TACGT, ACGTA
ACGTACGTA	6	ACGTAC, CGTACG, GTACGT, TACGTA

Table 1: The examples illustrate the process of k-mer tokenization for a DNA sequence. Each k-mer constitutes a substring of length k extracted from the DNA sequence. The provided examples align with the availability of DNABERT pre-trained models.

3.1.2 Architecture

The general architecture of all DNABERT models is consistent with that of BERT (devlin2018bert, ji2021dnabert). However, it should be emphasized that DNABERT employs an encoder-only transformer, specifically utilizing the encoder part of the transformer architecture. The model incorporates 12 encoder layers, each comprising 768 hidden units and 12 attention heads. It is imperative to note that DNABERT models are capable of processing input sequences up to 512 tokens in length. To ensure compatibility with the model's context window, longer sequences must undergo truncation or reduction to this threshold.

3.1.3 Pre-Training

Pretraining in the context of transformer models refers to the initial phase of training a model on a large, general dataset using unsupervised or self-supervised learning objectives. The pretraining of DNABERT relies on masked language modeling (MLM), a technique widely used in transformer models like BERT. During this process, a portion of the k-mers within each DNA sequence is randomly masked. The model is then trained to predict the masked k-mers using the remaining sequence context. This strategy allows DNABERT to learn bidirectional representations of DNA sequences, capturing information from both upstream and downstream nucleotides.

DNABERT was pretrained using the human reference genome (GRCh38), which encompasses the entirety of the human genomic sequence. This comprehensive dataset provides a rich and varied foundation for the model, enabling it to learn the intricate patterns and variations found in human DNA. The use of reference genomes aid in the generalizability of the model to various downstream bioinformatics tasks [National Center for Biotechnology Information, 2025].

3.2 Fine-tuning the DNABERT model

Fine-tuning is the process of taking a pretrained model and adapting it to perform a specific task. This involves training the model on a smaller, task-specific dataset while leveraging the knowledge it learned during the pretraining phase. It builds upon the representations learned during pretraining, enabling the model to perform various analyses such as promoter prediction, splice site detection, and mutation impact evaluation.

3.2.1 Environment Setup

In the DNABERT setup process, the initial steps entail enabling GPU acceleration in Google Colab and installing the requisite libraries, such as GPU-compatible versions of PyTorch and TorchVision, utilizing the pip module.The DNABERT GitHub repository[2] is then cloned to obtain the necessary functions for training, with all dependencies from its ”requirements.txt” file being installed. Subsequently, the Seminar DLMB 2024 Winter GitHub repository[3] is cloned to acquire training and test examples. Additionally, the pre-trained DNABERT .zip-file with all the weights of the model is loaded from a google drive[4].

3.2.2 Data Pre-processing

The first step in fine-tuning is preparing task-specific datasets. The dataset used in this study focuses on S. aureus and its resistance to cefoxitin, an antimicrobial agent. The raw data, comprising 150 sequences and their corresponding labels, was sourced from publicly available datasets. Among these, 40 sequences represent non-resistant strains, while 110 sequences are resistant, highlighting a significant imbalance in the dataset as also visualized by figure 2.

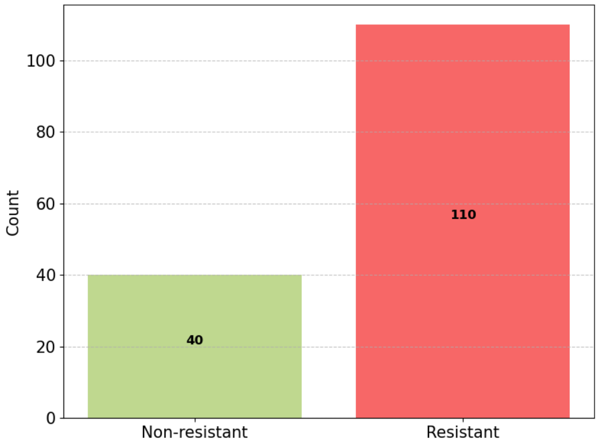

Figure 2: The bar chart illustrates the class imbalance present in the available dataset concerning *S. aureus* resistance to cefoxitin.The first class, which consists of non-resistant pathogens, is represented by the color green. The second class, which consists of resistant pathogens, is represented by the color red.

Functions from the Seminar DLMB 2024 Winter Public repository were utilized to structure and pre-process the data. The dataset was processed using the create_gene_datasets function, which splits the sequences into training and testing sets based on genes. This function organizes sequences by their gene associations, filters for genes with sufficient data coverage, and outputs gene-specific files along with their corresponding label files. For downstream analysis, the load_gene_data function was employed to load the processed data into a structured format, providing tuples of gene names, sequences, and labels for both training and testing sets.

The pre-processing scripts leveraged the BioPython library for parsing and manipulating DNA sequences and ensured consistency in the dataset format for subsequent analysis. All scripts and methods adhere to reproducibility standards, enabling others to replicate the dataset creation process.

[2] DNABERT GitHub link: https://github.com/jerryji1993/DNABERT

[3] Seminar GitHub link: https://github.com/hzi-bifo/seminar-dlmb-2024-winter-public

[4] DNABERT-6 Google Drive link: https://drive.google.com/file/d/1BJjqb5Dl2lNMg2warsFQ0-Xvn1xxfFXC/view

3.2.3 Training, Validation and Testing Dataset Preparation

The data imbalance poses challenges for machine learning models, which may become biased towards the majority class. To ensure fair representation of both classes in the train, validation, and test sets, a process of shuffling and splitting the data in a balanced manner is necessary. First, the combined dataset of sequences and labels is separated into two groups based on class labels: resistant (class 1) and non-resistant (class 0). Each class is then shuffled randomly to eliminate any bias introduced by the order of the original data. This randomization ensures diverse and unbiased subsets for training and evaluation.

The splitting process creates balanced subsets. The test set is formed by selecting an equal number of samples (seven) from each class, ensuring fair representation of both resistant and non-resistant sequences. Similarly, the validation set is constructed by selecting another seven samples from each class. The remaining data is used to form the training set, which reflects the overall class distribution but still includes sufficient examples of both classes for effective learning. The resulting dataset split can be seen in figure 3.

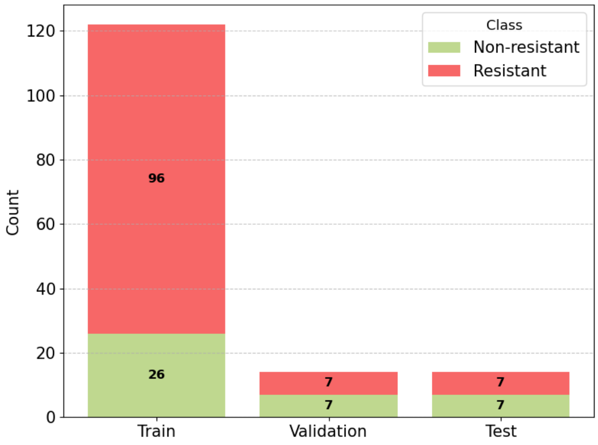

Figure 3: The dataset is divided into three sets: a training set, a validation set, and a test set. The class balance is highlighted by colors, with the number of non-resistant examples per set highlighted in green and the number of resistance examples highlighted in red.

This balancing process addresses the inherent class imbalance in the dataset (40 non-resistant and 110 resistant sequences). A balanced test set ensures unbiased model evaluation, preventing performance metrics from being dominated by the majority class. Likewise, a balanced validation set provides reliable feedback for tuning model hyperparameters. Random shuffling reduces the risk of biased splits, and using a fixed random seed ensures the essential reproducibility of the splits. This careful preparation is particularly important when working with imbalanced datasets to achieve robust and fair model performance.

3.2.4 Tokenization of Sequences

Prior to the utilization of the datasets for training, validation, and testing, it is imperative to tokenize the pbp4 sequences in a manner that is compatible with DNABERT. Among the various k-mers sizes of DNABERT, the 6-mers has been identified as the most proficient in the tests outlined in the original paper [Ji et al., 2021]; therefore, it should be employed in this work. To achieve 6-mer tokenization, a customized tokenizer is developed.

PBP4 sequences are 1,296 symbols in length. Following classical 6-mer tokenization, 1,291 tokens remain, with each token representing the identifier of one of the possible 4,096 tokens. This presents a challenge, as the context window of DNABERT is only 512 tokens long, meaning that less than half of the available genomic data fits in. Truncating the sequence leads to a substantial loss of information. Alternatively, the authors of DNABERT propose an XL version that divides the input sequences into smaller units and combines the output embeddings for classification [Ji et al., 2021]. However, this approach was subsequently rejected due to concerns regarding efficiency and effectiveness [Zhou et al., 2023].

For this work, the issue is circumvented by utilizing solely every third token obtained through classical 6-mers tokenization. Consequently, the length of the tokenized sequences is diminished to 430 tokens, thereby aligning with the context window of DNABERT. Given the implementation of 6-mers, no information is discarded, as there remains a three symbols overlap between neighboring tokens. This principle is illustrated in figure 4. The identifier for each token's structure remains unaltered, ensuring that the pre-trained model continues to recognize tokens based on its internal knowledge.

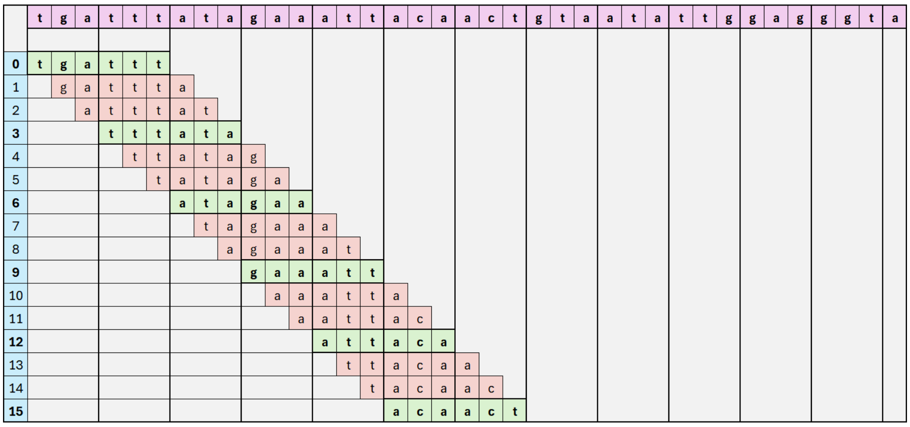

Figure 4: The figure illustrates the functionality of the specialized tokenizer employed in this work. The tokenizer utilized operates on a 6-mers tokenizer, yet it employs only every third token. The green tokens represent the tokens that are employed, while the red ones are discarded. This approach results in a substantial reduction of the total number of tokens per sequence. As demonstrated in the provided figure, the original overlap of five symbols is reduced to three, indicating that no information is lost and the fundamental principles of k-mers remain applicable.

3.2.5 Custom Weighted Sequence Classification Model Using DNABERT

A custom sequence classification model using BERT from Hugging Face's [Wolf et al., 2020] Transformers library is defined. The model incorporates a weighted loss function to address class imbalance[5]. It is built by extending PyTorch's torch.nn.Module, allowing for seamless integration with PyTorch's training and evaluation workflows. A pretrained BERT model is loaded using BertForSequenceClassification, and a class-weighted cross-entropy loss function (torch.nn.CrossEntropyLoss) is defined to ensure that errors on minority classes are penalized more heavily during training.

The forward method processes input sequences and attention masks, returning either raw predictions or logits along with a computed loss if labels are provided. This setup makes the model versatile for both training, where loss computation is needed, and inference, where only predictions are required. By using class weights, the model adjusts its learning to account for the underrepresentation of the minority class in the dataset, mitigating the risk of bias towards the majority class.

3.2.6 Definition of hyper-parameters for fine-tuning

The fine-tuning of DNABERT is achieved through the implementation of a specific set of hyper-parameters, with the training process structured into epochs comprising 16 examples per batch.The training is then executed with a learning rate of 1e-5 for a total of 50 epochs. As a preventive measure against overfitting the model to the training data, the weight decay is set to 1e-2.

4 Results and Discussion

In the subsequent section, the results of this report are presented and critically discussed.

[5] Tensor to counter class imbalance is defined as: [2.3462, 0.6354]

4.1 Training and Validation Loss

Firstly, the progression of the model error during the training process can be observed in Figure 5. The blue graph describes the loss on training data, which is known to the model from training. This loss is continuously decreasing as the training progresses to later epochs. The orange graph shows the loss on validation data. The orange graph fluctuates significantly due to its smaller sample size, and no substantial improvement trend is apparent in this graph.

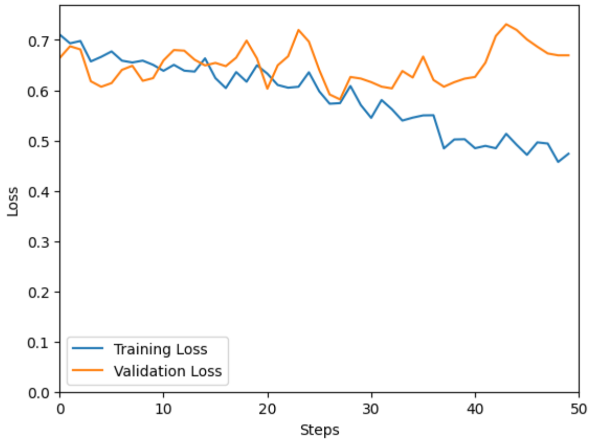

Figure 5: Training and validation loss (error) during fine-tuning of the DNABERT-6 model.

4.2 Model Performance on the Test Set

Subsequent to the completion of the training phase, the refined DNABERT-6 model was assessed on a designated test set comprising seven examples for each class. These instances were previously unseen by the model, as they had not been incorporated into any training or validation procedures. This approach was adopted to ensure the test's independence from potential errors induced by overfitting. A concise overview of key metrics is provided in Table 2. Notably, the model exhibits perfect precision on non-resistant pathogens and perfect recall for resistant pathogens. The model's overall performance is well balanced between precision and recall, with both values being similar to each other. The model's quality is best characterized by the F1-score of 0.78, indicating that a significant portion of the variance can be explained by the model. However, it should be noted that further improvements are possible and that the model is not yet optimal. Nevertheless, the achieved performance is significantly better than that of Tharmakulasingam et al. [2023] (F1 = 0.47).

Category	**Precision**	**Recall**	**F1-score**
`Non-resistant`	1.0	0.57	0.73
`Resistant`	0.70	1.00	0.82
`Combined`	0.85	0.79	0.78

Table 2: Performance of fine-tuned DNABERT-6 model on the test set.

4.3 Confusion Matrix Analysis

A thorough evaluation of the model's predictions enables a more profound analysis of its strengths and limitations. A review of the confusion matrix depicted in Figure 6 reveals a notable observation: predictions of a pathogen being non-resistant are very trustworthy, and with four out of seven cases, the majority is identified. However, the classification of resistant pathogens does exhibit some inaccuracies, though the overall accuracy remains satisfactory. Further testing is necessary to ascertain the definitive status of antimicrobial resistance (AMR) in these unclear cases, contingent upon the specific application under consideration.

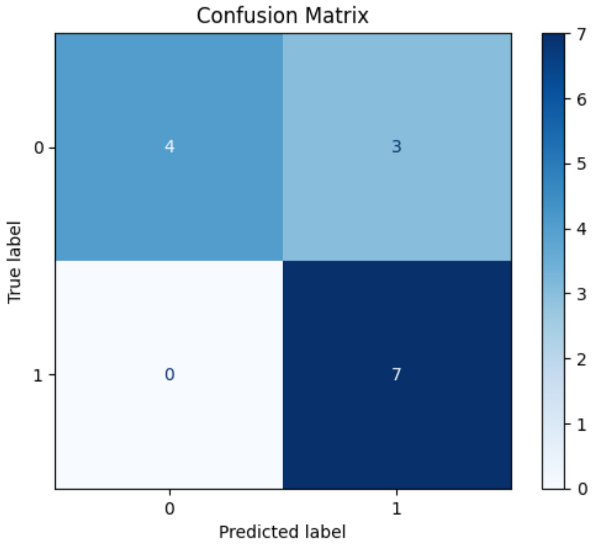

Figure 6: Confusion matrix of performance of fine-tuned transformer encoder model on the test set.

4.4 Limitations and Challenges

The validity of the results reported in this work is evidently diminished by the limited number of classification examples available for training, validation, and testing. With a test set comprising merely 14 examples, the statistical significance is found to be unsatisfactory. Additionally, the dataset exhibits a considerable imbalance between the two classes, which further exacerbates the issue. Notwithstanding these limitations, this report aims to provide a preliminary assessment of the potential performance of encoder-only transformers in predicting AMR in *S. aureus*.

The utilization of a pre-trained DNABERT model serves to address the challenge posed by the scarcity of training examples by offering a model that is universally applicable and already adept at handling genomic sequences. K-mers, a prevalent choice, will likely continue to be utilized even if it is not considered state-of-the-art in 2025. More advanced tokenizers employed in subsequent DNABERT models, as referenced in Zhou et al. [2023], are more intricate to implement and less transparent in their application.

A problem encountered with the DNABERT-6 model employed is the short context window of 512 tokens. As the XL variant proposed by Ji et al. [2021] is unclear in quality and complex in implementation, a different solution was chosen. The large overlap of the 6-mers tokenization was used to skip two-thirds of tokens and reduce the number of tokens this way. This approach ensures that no information is lost between tokens, enabling the processing of the entire sequence while maintaining the model's capacity to recognize individual tokens. The potential negative implications of excluding the other tokens remain uncertain. The current findings indicate that this technique enhances the model's performance compared to simply truncating the sequences and omitting half the information, as evidenced by the test performance of both approaches. However, it is important to acknowledge that the results of both approaches are relatively similar, and further research is necessary to definitively evaluate this method.

A comparison of the present AMR detection procedures with those involving an encoder-only transformer reveals several notable advantages. The former approach is characterized by its enhanced efficiency, reduced cost, and simplified operation. This is primarily due to the fact that the genomic sequence of the PBP4 gene is the sole requisite, thereby eliminating the need for additional manual or physical labor.

5 Conclusion

This report investigated the potential of utilizing a transformer encoder model, specifically DNABERT-6, to predict antimicrobial resistance (AMR) in *S. aureus* against cefoxitin. By leveraging the genomic sequence of the PBP4 gene, the research aimed to enhance AMR prediction accuracy and demonstrate the applicability of transformer models in the domain of computational microbiology.

The model, which was subjected to a fine-tuning process, attained an F1-score of 0.78 on a balanced test set. This result indicates that the DNABERT-6 model significantly outperforms existing approaches, including the work by Tharmakulasingam et al. [2023]. The findings of this report underscore the potential of transformer models in the analysis of genomic data and highlight their capacity to detect complex sequence-level patterns associated with antibiotic resistance. The report further validates the efficacy of k-mer tokenization when working with genomic sequences.

Despite the noteworthy accomplishments, several challenges have been identified. The limited dataset size, in conjunction with class imbalance, has constrained the statistical robustness of the results. Moreover, the context window constraint imposed by DNABERT-6 has necessitated the development of a custom token reduction strategy, which, while demonstrating efficacy, introduces potential ambiguities regarding performance loss. These limitations underscore the necessity for larger datasets and more sophisticated pre-processing methodologies to fully leverage the capabilities of transformer models for AMR prediction.

Future studies should therefore center on confronting these challenges through the incorporation of more extensive and varied datasets to enhance the model's generalizability. Further research into sophisticated tokenization techniques and hyperparameter optimization holds the potential to boost prediction accuracy. Moreover, a direct comparison of this transformer-based approach with conventional AMR detection methods in terms of efficiency, cost, and scalability would offer significant insights for practical applications.

References

Jay Alammar. The illustrated transformer. https://jalammar.github.io/illustrated-transformer/, 2018. Accessed: 2025-01-06.

Thaina M Da Costa, Carolina R De Oliveira, Henry F Chambers, and Som S Chatterjee. Pbp4: a new perspective on staphylococcus aureus β-lactam resistance. *Microorganisms*, 6(3):57, 2018.

Ruud H Deurenberg and Ellen E Stobberingh. The evolution of staphylococcus aureus. *Infection, genetics and evolution*, 8(6):747–763, 2008.

Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Clarence J Fernandes, Lorna A Fernandes, and Peter Collignon. Cefoxitin resistance as a surrogate marker for the detection of methicillin-resistant staphylococcus aureus. *Journal of Antimicrobial Chemotherapy*, 55(4):506–510, 2005.

Timothy J. Foster. The staphylococcus aureus "superbug". *The Journal of Clinical Investigation*, 114(12):1693–1696, 2004. doi: 10.1172/JCI200423825. URL http://www.jci.org/articles/view/23825.

Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, 02 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btab083. URL https://doi.org/10.1093/bioinformatics/btab083.

Guido Memmi, Sergio R Filipe, Mariana G Pinho, Zhibiao Fu, and Ambrose Cheung. Staphylococcus aureus pbp4 is essential for β-lactam resistance in community-acquired methicillin-resistant strains. *Antimicrobial agents and chemotherapy*, 52(11):3955–3966, 2008.

Minnesota Department of Health. Staphylococcus aureus fact sheet. Minnesota Department of Health website, 2010. Available at: https://www.health.state.mn.us/diseases/staph/basics.html [Accessed January 5, 2025].

National Center for Biotechnology Information. Genome data viewer - homo sapiens (human), 2025.

Jesper Olsson. Predicting antibiotic resistance using fusion transformers. Technical report, Chalmers University of Technology, 2024. URL http://hdl.handle.net/20.500.12380/308525.

Mukunthan Tharmakulasingam, Wenwu Wang, Michael Kerby, Roberto La Ragione, and Anil Fernando. Transamr: An interpretable transformer model for accurate prediction of antimicrobial resistance using antibiotic administration data. *IEEE Access*, 11, 2023. doi: 10.1109/ACCESS.2023.3296221.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

David Velasco, Maria del Mar Tomas, Monica Cartelle, Alejandro Beceiro, Astrid Perez, Francisca Molina, Rita Moure, Rosa Villanueva, and German Bou. Evaluation of different methods for detecting methicillin (oxacillin) resistance in staphylococcus aureus. *Journal of Antimicrobial Chemotherapy*, 55 (3):379–382, 2005.

Martin Vestergaard, Dorte Frees, and Hanne Ingmer. Antibiotic resistance and the mrsa problem. *Microbiology spectrum*, 7(2):10–1128, 2019.

R Wise, T Hart, O Streulens M Cars, R Helmuth, and P Huovinen. Antimicrobial resistance is a major threat to public health british medical journal, 5 sept. 1998.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. https://github.com/huggingface/transformers, 2020. Accessed: 2025-01-05.

World Health Organization. Who bacterial priority pathogens list, 2024. ISBN: 9789240093461.

Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. Dnabert-2: Efficient foundation model and benchmark for multi-species genome, 2023.