Technische Universität Braunschweig

Predicting Antimicrobial Resistance (AMR) in Staphylococcus aureus with Transformer Models Using the genomic sequence of the pbp4 gene

Authors: Dagh Zeppenfeld, Phoebe Gwimo

Date: 23.01.2025

Contents

AMR & Staphylococcus aureus

AMR

- Antimicrobial Resistance
- Efficacy of antibiotics as a treatment is reduced ⇒ Global health challenge

Origin of AMR

- Evolutionary behaviour of pathogens
- Very high reproduction rates

Example

- Staphylococcus Aureus and Cefoxitin

AMR & Staphylococcus aureus

Staphylococcus Aureus

- Discovered in the 1880s [3]
- Gram-positive bacterium
- Often harmless, with 20% of individuals persistently colonized with S. Aureus [6]
- Can cause infections [9]
  - Minor skin conditions (impetigo / boils)
  - Life-threatening illnesses (sepsis / endocarditis / pneumonia)

AMR in S. Aureus

- S. Aureus can adapt and resist treatment
- Emergence of methicillin-resistant S. Aureus (MRSA) [16]
- WHO has prioritized development of novel antibiotics against MRSA [18]

AMR & Staphylococcus aureus

Cefoxitin

- β-lactam antibiotic against S. Aureus
- Widely used as surrogate marker for methicillin resistance [5]
- Resistance is mainly driven by variations of the pbp gene
  - encodes penicillin-binding proteins
  - can enhance pathogen survival under β-lactam exposure by strengthening cell walls [2]

Research Issue

- Efficiently classifying methicillin resistance of S. Aureus variations
- Current Method:
  - Confirming presence of specific genes (mecA / mecC) with molecular methods like PCR [8]
- Proposed Method:
  - Predict AMR with a transformer-encoder model, based on the genetic sequence of pbp4

Transformer Model Architecture

Definition

- Neural networks using self-attention mechanisms
- Introduced by Vasvani in 2017 [13]
- Specifically developed for sequence transduction tasks (e.g. translation & sequence tagging)

Architecture [1]

- Encoder-Decoder structure
  - Encoder processes the input sequence and generates contextualized embeddings
  - Decoder uses embeddings to produce the target sequence
- Both consist of multiple layers
  - Each layer containing attention mechanisms and feed-forward neural networks

Transformer Model Architecture

Tokenization

- The input sequence is tokenized into smaller units called tokens
- Tokenizers can be implemented in many different ways

Example

- Input:
  - ACGTACGTA
- Tokens:
  - ACG, CGT, GTA, TAC, ACG, CGT, GTA

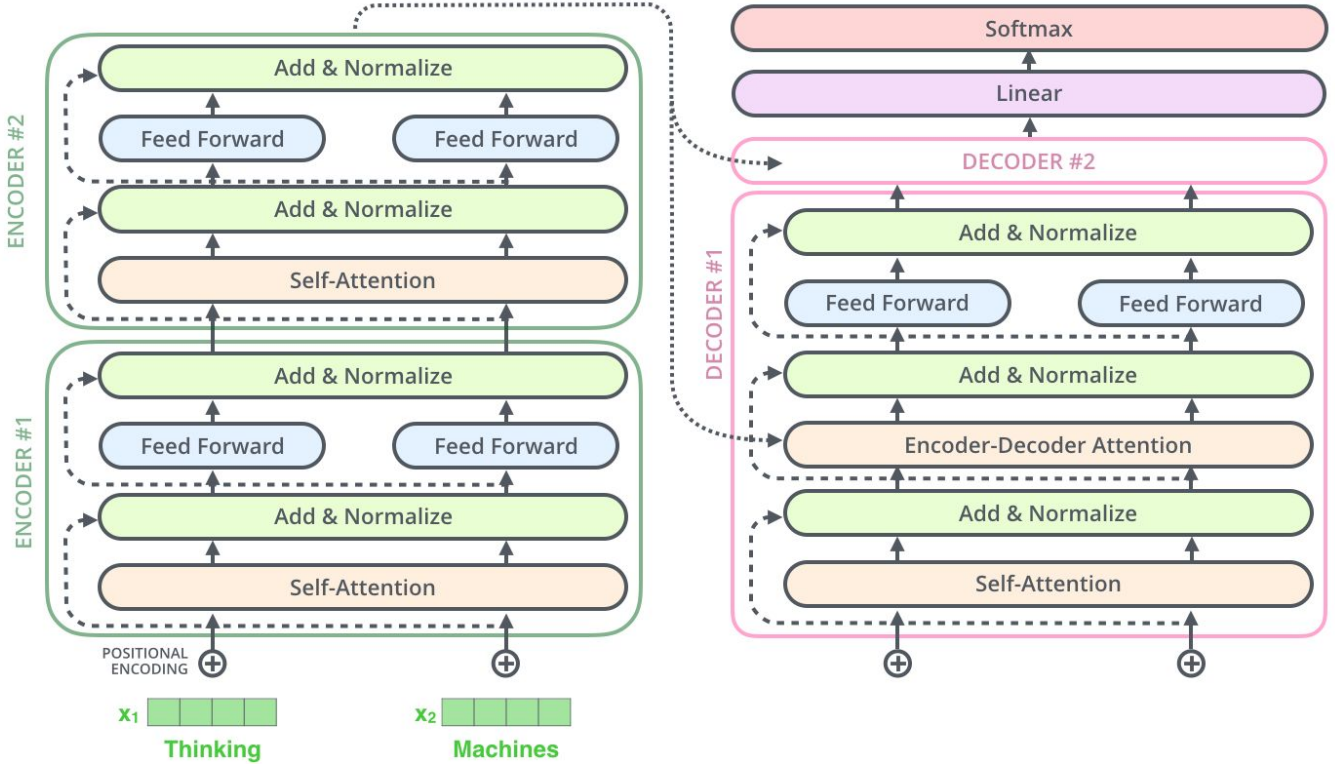
Embedding Layer

- produces continuous vector representations from tokens

Positional Encoding

- capture positional information which is not inherent to the model's architecture

Transformer Model Architecture



[1]

Transformer Model Architecture

Attention

- Makes context / relationships between tokens visible
- Defined by an equation:

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

Where

- 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.
- A scaling factor to prevent excessively large dot-product values.
- softmax: Normalizes similarities into probabilities, which serve as attention weights.

Multi-head attention: The attention process is done multiple times to capture various types of relationships.

Transformer Model Architecture

Residual Connections and Layer Normalization

- Residual connections add the original input to the sub-layer output, followed by normalization

Feedforward Layer

- This output is passed to the Feed-Forward Network (FFN), a two-layer neural network with a ReLU activation function.

Stacking Transformer Layers

- The encoder stack consists of N identical layers, each applying multi-head attention, FFN, and residual normalization sequentially.

Transformer Model Architecture

Encoder-Only Models

- not all applications require both encoder and decoder

  ⇒ can be used individually!
- tailored for tasks requiring input sequence understanding rather than sequence generation

Examples

- Bidirectional Encoder Representations from Transformers (BERT) [4]
- DNABERT [7,19]

Existing Work

Tharmakulasingam et al (2023) [12]

- *TransAMR: An Interpretable Transformer Model for Accurate Prediction of Antimicrobial Resistance Using Antibiotic Administration Data*
- AMR prediction with Machine Learning
- Based on a large dataset
  - Over 700 Features (demographic / procedural / …)
  - High number of training examples

Olsson (2024) [11]

- *Predicting antibiotic resistance using fusion transformers: A framework for training a multimodal transformer using data fusion of genotype, phenotype, and metadata to improve predictions of antibiotic resistance in Escherichia coli*
- Transformer-based AMR prediction for Escherichia coli
- Based on large dataset
  - Millions of data points
  - Diverse features available

Existing Work

Comparison:

- Both studies used very large and diverse datasets
- Tharmakulasingam et al. reached a peak performance of F1 = 0,47
- Olsson encountered a wide range of results

Research Gap:

- Focus on S. Aureus
- Focus on a singular feature
  - like pbp4 genomic sequence

DNABERT

Model: DNABERT [7,19]

- Specialized adaptation of BERT for genomic sequences
- Models DNA sequences as a “language”

Tokenization

- Uses k-mers tokenizer
  - Overlapping substrings of length k
- Longer k-mers help to capture local sequence pattern and dependencies
- Overlapping ensures contextual continuity

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

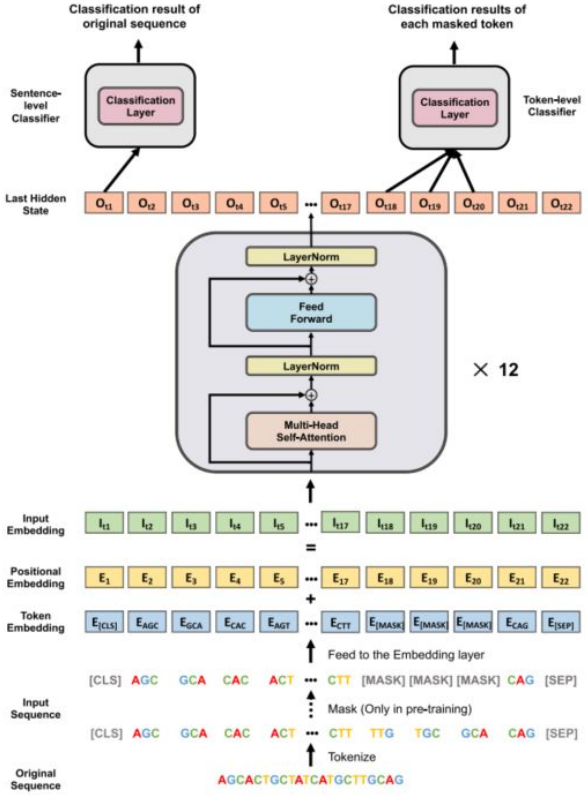
DNABERT

Architecture

- Encoder-only model
- Incorporates 12 encoder layers
- Neural nets comprise of 768 hidden units
- Capable of sequences up to 512 tokens in length
- Adds a classification layer on top

Pre-Training

- Pre-trained on a large and general dataset
- DNABERT was trained on human reference genome
  - GRCh38
  - Known to aid the generalizability of models [10]
- Masked Language Modeling (MLM) was used
  - Portion of tokens are masked and need to be predicted by the model



[7]

Fine-tuning of DNABERT

Definition of Fine-tuning

- Adapting a pre-trained model to perform a specific task
- Dataset can be much smaller
- Takes much less compared compared to pre-training

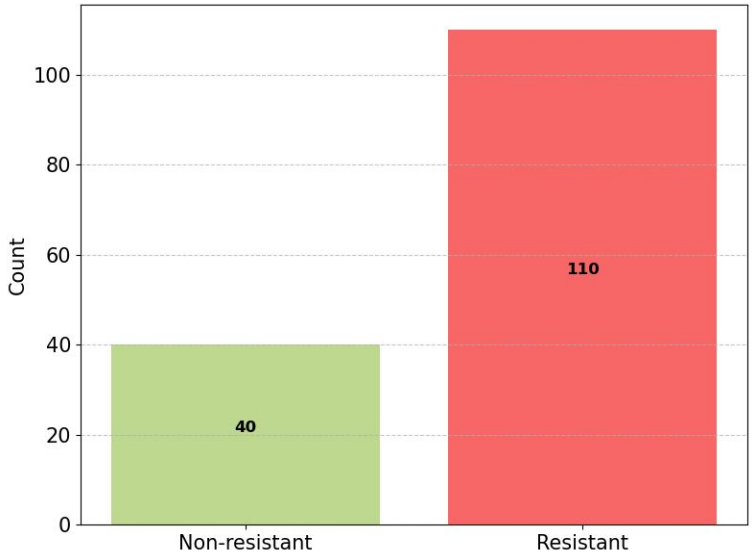
Steps needed:

1. Data collection and preprocessing
2. Tokenization using k-mers
3. Fine-tuning DNABERT for AMR prediction

Fine-tuning of DNABERT: Data collection and preprocessing

Dataset:

- 150 genomic sequences
- Imbalance in number of samples
  - 40 non-resistant vs. 110 resistant



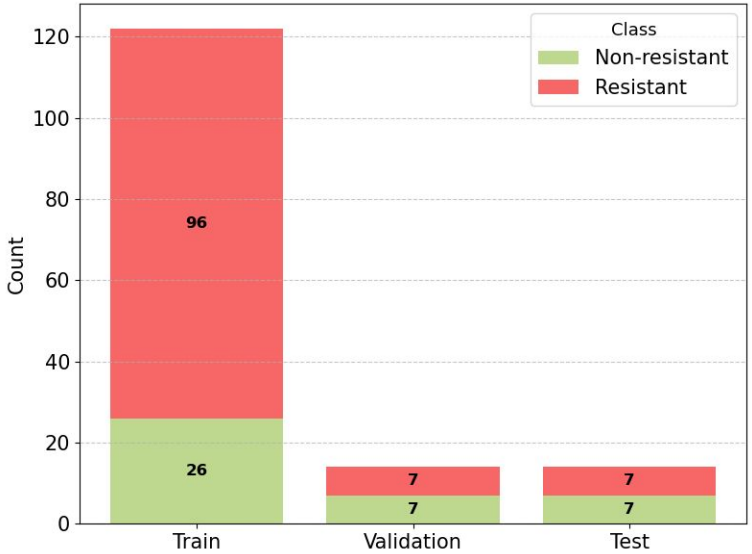
Fine-tuning of DNABERT: Data collection and preprocessing

Data Splitting:

- Training, validation, and testing
- Validation and Testing are balanced

Class Weights

- To combat imbalance in training data



Fine-tuning of DNABERT: Tokenization using k-mers

Tokenizer

- 6-mer tokenization
  - Best performance in original paper [7]

Challenge

- pbp4 sequences are 1296 symbols in length

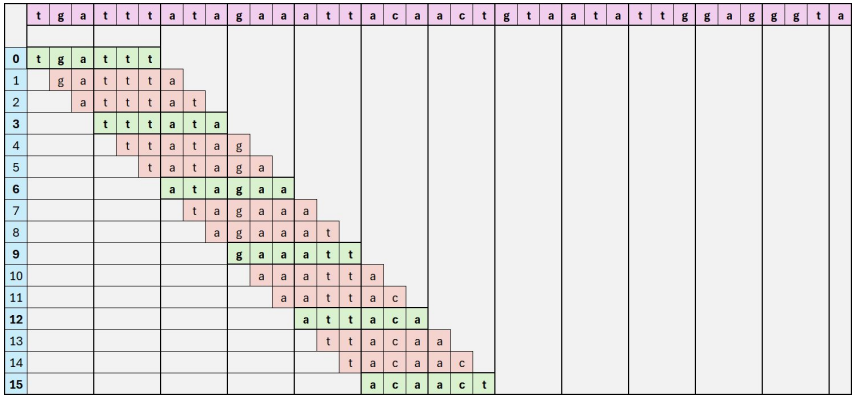
  ⇒ exceeds DNABERT 512-token limit

Solution

- Select every third token

  ⇒ No information is lost

  ⇒ Maintains overlap



Fine-tuning of DNABERT

Environment Setup:

- GPU-accelerated processing in Google Colab
- Libraries: PyTorch, Hugging Face Transformers

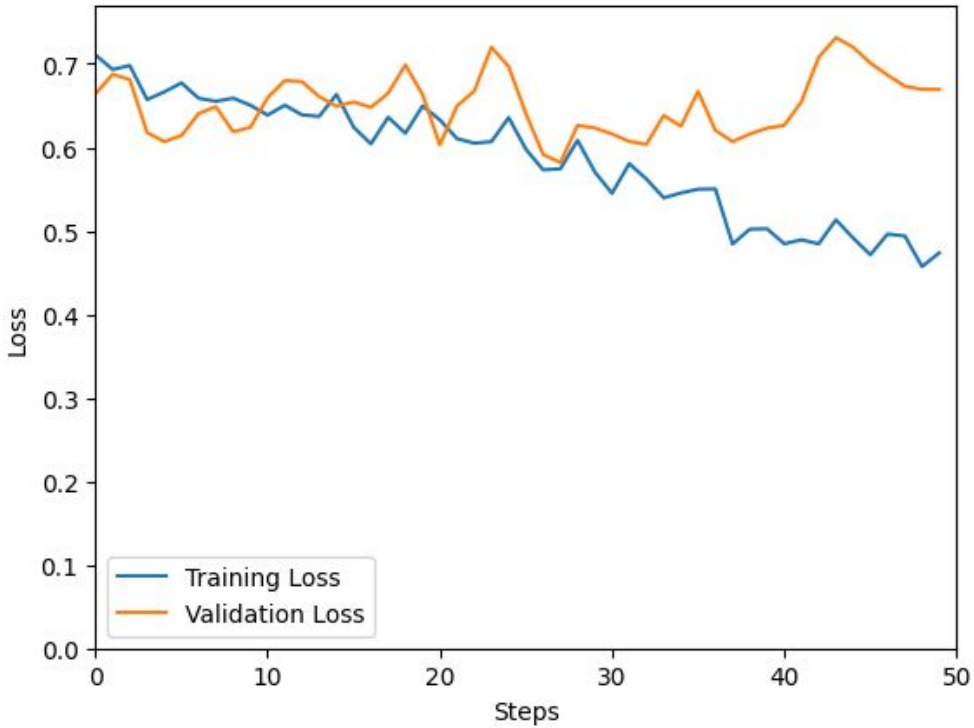
Hyperparameters:

- Batch size: 16
- Learning rate: 1e-5
- Epochs: 50
- Weight decay: 1e-2

Results and Discussion

Training Performance:

- Training loss decreases consistently
- Validation loss shows fluctuations



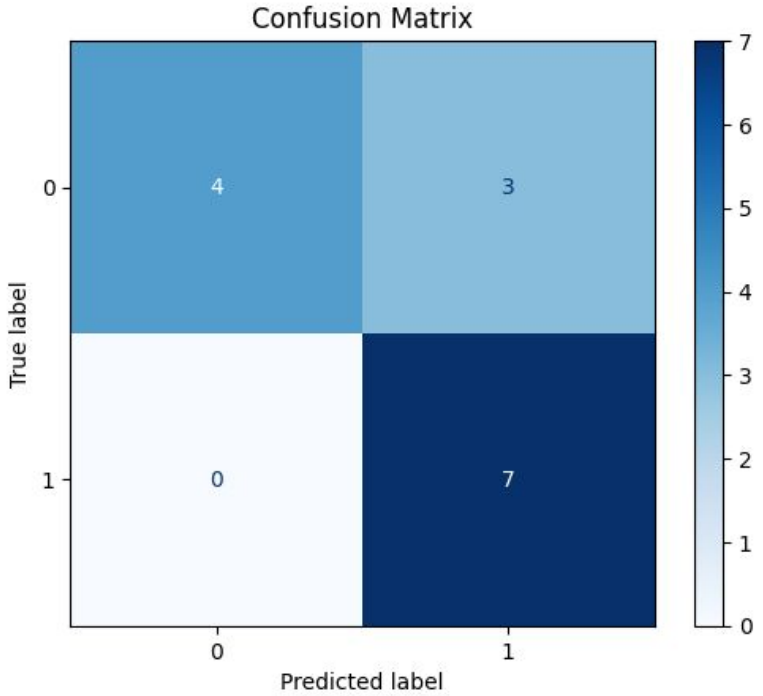
Results and Discussion

Test Set Performance:

- Precision: 0.85
- Recall: 0.79
- F1-score: 0.78 (improves on existing work)

Confusion Matrix Analysis

- Non-resistant predictions are highly reliable
- Some inaccuracies in resistant predictions
- Balanced performance but scope for improvement



Limitations and Challenges

Issues Identified:

- Small dataset size
  - statistical significance negatively influenced
- Class imbalance
  - Difficult to fine-tune
- Context window constraints in DNABERT-6

Solutions

- Pre-trained model chosen to combat low data availability
- Adjusting the k-mers tokenizer by skipping 2/3rds of tokens
  - Impact is not examined in detail

Conclusion

Result

- The method was implemented prototypically
- A solid prediction of AMR against Cefoxitin can be made
- F1 performance is greater than in similar classification efforts

Future Work

- Applying the proposed method on larger datasets
- Further boosting prediction accuracy
- Direct comparison of proposed and current classification methods

Acknowledgments and References

[1] - Jay Alammar (2018): *The Illustrated Transformer*

[2] - Thaina M. Da Costa et al. (2018): *Pbp4: A new perspective on Staphylococcus aureus β-lactam resistance*

[3] - Ruud H. Deurenberg and Ellen E. Stobberingh (2008): *The evolution of Staphylococcus aureus*

[4] - Jacob Devlin (2018): *BERT: Pre-training of deep bidirectional transformers for language understanding*

[5] - Clarence J. Fernandes et al. (2005): *Cefoxitin resistance as a surrogate marker for the detection of methicillin-resistant Staphylococcus aureus*

[6] - Timothy J. Foster (2004): *The Staphylococcus aureus "superbug"*

[7] - Yanrong Ji et al. (2021): *DNABERT: Pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome*

[8] - Guido Memmi et al. (2008): *Staphylococcus aureus Pbp4 is essential for β-lactam resistance in community-acquired methicillin-resistant strains*

[9] - Minnesota Department of Health (2010): *Staphylococcus aureus Fact Sheet*

Acknowledgments and References

[10] - National Center for Biotechnology Information (2025): *Genome Data Viewer - Homo sapiens (Human)*

[11] - Jesper Olsson (2024): *Predicting Antibiotic Resistance Using Fusion Transformers*

[12] - Mukunthan Tharmakulasingam et al. (2023): *TransAMR: An interpretable transformer model for accurate prediction of antimicrobial resistance*

[13] - A. Vaswani (2017): *Attention Is All You Need*

[14] - David Velasco et al. (2005): *Evaluation of different methods for detecting methicillin resistance in Staphylococcus aureus*

[15] - Martin Vestergaard et al. (2019): *Antibiotic resistance and the MRSA problem*

[16] - R. Wise et al. (1998): *Antimicrobial resistance is a major threat to public health*

[17] - Thomas Wolf et al. (2020): *Transformers: State-of-the-Art Natural Language Processing*

[18] - World Health Organization (2024): *WHO Bacterial Priority Pathogens List*

[19] - Zhihan Zhou et al. (2023): *DNABERT-2: Efficient Foundation Model and Benchmark for Multi-Species Genome*

Thank you for your attention

Any questions?