scientific reports

Check for updates

OPEN

Integrating genomic and molecular data to predict antimicrobial minimum inhibitory concentration in *Klebsiella pneumoniae*

Byeonggyu Ryu[1], Woosung Jeon[2] & Dongsup Kim[2]✉

Minimum inhibitory concentration (MIC) denotes the in vitro benchmark indicating the quantity of antibiotic required to inhibit proliferation of specific bacterial strains. Determining MIC values corresponding to the infecting bacterial strain is paramount for tailoring appropriate antibiotic therapy. In the interim between specimen collection and laboratory-derived MIC outcomes, clinicians frequently resort to empirical therapy informed by retrospective analyses. Here introduces two deep learning approaches, a Convolutional Neural Network (CNN)-based model and an Enformer-based model, integrating genomic data of *Klebsiella Pneumoniae* and molecular structural data of 20 antibiotics to anticipate the MIC value of the bacterium for each antibiotic under consideration. These models demonstrate enhanced raw accuracy over the existing state-of-the-art model, which rely exclusively on genomic data. The CNN-based model achieves a notable 20% increase in raw accuracy while further mirroring the 1-tier accuracy of the state-of-the-art model. Although the Enformer-based model does not quite reach the performance levels of the CNN-based model, it offers an advantage by eliminating the need for arbitrary data processing steps. This streamlining of the data processing pipeline facilitates fast updates and improves the model interpretability. It is expected that these deep learning paradigms can significantly inform and bolster clinician decision-making during the empirical treatment phase.

Keywords Minimum inhibitory concentration, Convolutional neural networks, Enformer, *Klebsiella Pneumoniae*

Klebsiella pneumoniae (Kpn) is a member of the Enterobacteriaceae family, characterized as a gram-negative, encapsulated, non-motile bacterium[1]. Kpn has long been acknowledged as a significant pathogen in healthcare environments, accounting for 3–8% of all hospital-acquired infections in the United States[1]. Its pathogenicity is heightened by its capability to rapidly acquire resistance mechanisms against a broad spectrum of antibiotics. Over the years, there has been a discernible escalation in resistance rates of Kpn. Notably, regions including Eastern and South-Western Europe, along with Mediterranean nations, have become endemic to multi-drug resistant Kpn due to extended-spectrum beta-lactamase (ESBL) producing strains[2]. More concerning has been the emergence and propagation of Carbapenem-Resistant *Klebsiella pneumoniae* (CRKP). Carbapenems, often considered the last line of defense against multi-drug resistant pathogens, have become increasingly ineffective against these resilient strains[3]. While the majority of regions reported negligible non-susceptibility in 2005, by 2015, there was a pronounced emergence of CRKP in several countries, including Romania, Italy, and Greece, with resistance rates ranging between 40% and 60%[2].

Antimicrobial susceptibility test (AST) serves as the conventional method to ascertain the sensitivity of microbial isolates to specific antimicrobial agents, facilitating tailored therapeutic regimens. Central to AST's methodology is the evaluation of bacterial growth in the presence of graded concentrations of antibiotics, typically employing broth microdilution or disk diffusion assays[4]. While AST remains a cornerstone in clinical microbiology, a pronounced limitation is the protracted turnaround time. The need to culture bacterial isolates until they reach a discernible growth phase can extend the diagnostic window by 24 to 48 h or even longer for slow-growing pathogens[5]. This inherent delay potentially compromises patient outcomes, as physicians may

[1]Department of Biological Sciences, Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea. [2]Department of Bio and Brain Engineering, Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea. ✉email: kds@kaist.ac.kr

need to administer empirical, broad-spectrum antibiotic therapy in the interim, which might inadvertently exacerbate the burgeoning antibiotic resistance crisis[6].

Recently, sharing of whole genome sequence (WGS) data in conjunction with clinical antimicrobial resistance (AMR) metadata through public platforms has facilitated the application of machine learning (ML) and deep learning (DL) techniques to predict AMR phenotypes[7]. Davis et al. and Drouin et al. proposed reference-free methods using *k*-mer features and AMR phenotype labels to generate ML models predicting resistance traits in multiple bacterial species[7–9]. Other studies including Coelho et al., Stoesser et al., Niehaus et al., Bradley et al., Pesesky et al., and Jeukens et al. have employed AMR genes, SNPs, WGS data, or a mix of these to construct ML classifiers that demonstrate high accuracy[7,10–15]. Specifically, for Kpn, Nguyen et al.[7] developed an XGBoost Regressor model grounded on 10-mer attributes and laboratory-sourced minimum inhibitory concentrations (MIC) value labels.

In a subsequent study, Nguyen et al.[16] developed a deep learning (DL) model on the same Kpn dataset, albeit with 8-mer features. Because they used 8-mer features, rather than 10-mer features as in previous studies[7], prediction accuracy of their CNN model was worse than that of XGBoost model. Reason for using shorter 8-mer features was that computational cost for building a CNN model was prohibitively large if 10-mer sequence features were used.

Notwithstanding the contributions of the aforementioned studies, there has been a conspicuous absence of efforts to incorporate the properties of the antimicrobials in the prediction of AMR phenotypes. Nguyen et al.[7] employed a one-hot encoding approach for the 20 antibiotics and appended them to the 10-mer feature vector. Nguyen et al.[16] devised distinct models for each of the 20 antibiotics.

In this study, we propose two DL approaches to integrate genomic and molecular structural data into the task of predicting laboratory derived MIC values of Kpn. The first model uses selected 10-mer features of Kpn genomes together with simplified molecular input line entry system (SMILES) data of 20 antibiotics. The integrated data matrix is processed through CNN layers to output the MIC value. The second model uses only AMR genes. The collection of AMR genes is integrated with SMILES data and processed through an Enformer architecture proposed by Avsec et al.[17]. Enformer is a DL architecture combining convolutional layers and transformer layers to allow memory efficient processing of long DNA sequences. The Enformer model's ability to take holistic sequences as input enhances its interpretability by preserving the sequential context of the data. Two metrics were primarily assessed: raw accuracy and accuracy within a ± 1 two-fold dilution factor (termed 1-tier), in line with prior studies by Nguyen et al.[7] and Nguyen et al.[16]. Both CNN and Enformer-based models outperformed the state-of-the-art model in raw accuracy, recording 0.84 and 0.77 respectively, compared to the state-of-the-art's 0.70. The CNN-based model demonstrated a 1-tier accuracy equivalent to the state-of-the-art model at 0.92, while the Enformer-based model attained a 0.85 1-tier accuracy. The CNN-based model's accuracies for individual antibiotics and error rates for predicting susceptibility profiles were also evaluated to enable a more comprehensive comparison with models from prior studies[7,16]. The interpretability of the Enformer model was assessed and illustrated by analyzing the attention matrices, providing insights into the significant positions within the input sequences.

Methods

Dataset preparation

The dataset used in this study is identical to the one used in Nguyen et al.[7]. The original dataset was obtained from Long et al.[18,19]. Over the course of 6 years (2011 ~ 2017) Kpn isolates from patients has been cultured by the Houston Methodist Hospital System[18]. Genomic assemblies were conducted using SPAdes[20] via the PATRIC assembly platform[7,21,22]. MIC values have been measured by the BD-Phoenix test for 20 antibiotics for each of the strain. The resulting database had total 1667 genomic data and 32,312 genome-antibiotic pairs with a corresponding MIC value. WGS contigs for the 1667 Kpn strains were retrieved in the FASTA '.fna' file format directly from the PATRIC FTP repository, using the specific PATRIC IDs enumerated in Nguyen et al.[7].

For the purpose of classification, MIC labels underwent a data cleansing process to convert them into integer values. Values delineated as '> x' were recalibrated to 2x, while those marked as '< x' were adjusted to x/2. Labels such as '>= x', '<= x', and x remained unaltered and were represented as x. For compound antibiotics like 'Ampicillin/Sulbactam', only the MIC of the primary antibiotic was considered, given that the value of the latter is either a constant or contingent upon the former[7].

An overview of the dataset revealed that, from a total of 32,312 genome-antibiotic pairs, three specific data points (573.12878 - Meropenem, 573.12981 - Meropenem, and 573.12924 - Meropenem) did not have a MIC label corresponding to a power of two. To ensure the coherence of integer values after Log2 scaling, these three data points were excluded. The refined dataset, comprising 32,309 data points, was then partitioned into 11 subsets of approximately equivalent size. A 10-fold cross-validation approach utilized ten of these sets to develop ten distinct models, reserving the eleventh set exclusively for testing across all models.

CNN-based model feature selection and encoding

The overall workflow of the CNN-based model is illustrated in Fig. 1c. In the process of feature engineering for the CNN-based model, we employed the *k*-mer strategy leveraging WGS data. The dataset, comprising 1667 genomic samples, yielded 524,800 unique 10-mers. However, forming a matrix with these rows presents significant computational memory challenges.

Guided by the methodology described in Nguyen et al.[7], we first constructed ten XGBoost Regressor models via 10-fold cross-validation on the WGS dataset. The 10-mer sequence counts were quantified utilizing KMC2[23] software. Each 10-mer was allocated an index, spanning from 0 to 524,799, ordered alphabetically. Subsequently, 20 antibiotics were attributed indices from 524,800 to 524,819, and the antibiotic corresponding to each data point was assigned a count value of 1. These files were integrated into the XGBoost model as a DMatrix, an

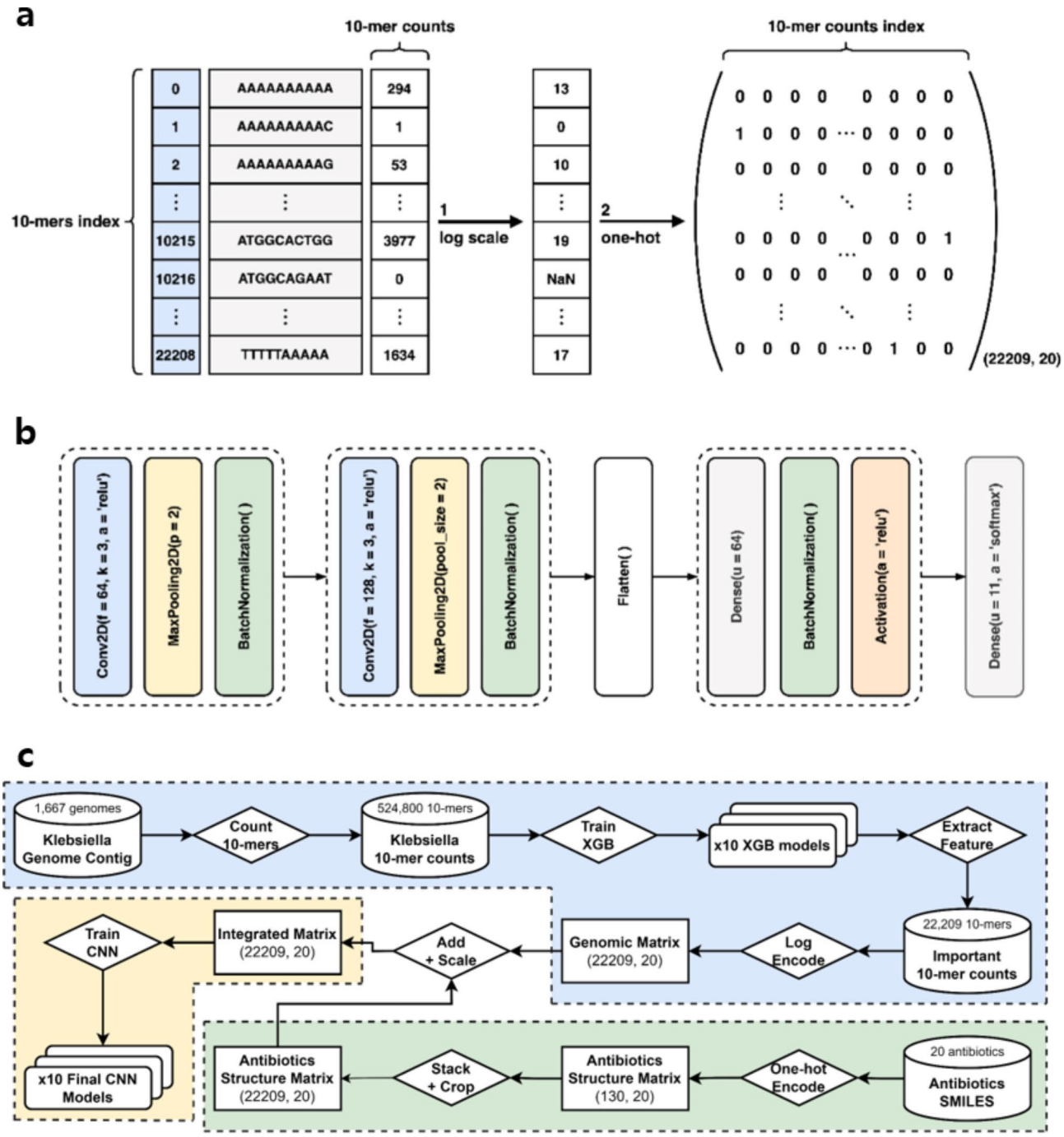

Fig. 1. Data encoding schematics in CNN based MIC prediction. (a) Representation of the encoding process for the 22,209 10-mer sequences derived from the XGBoost models. These sequences undergo log scaling with a base of 1.5529 before one-hot encoding, yielding a matrix of dimensions (22209, 20). (b) Configuration of the CNN model, which is designed in alignment with the architecture delineated by Nguyen et al.[16]. Layers within dotted areas represent modular components of the overall model. The first two dotted areas denote convolutional operations, each consisting of a convolutional layer followed by pooling and normalization. The last dotted area represents a dense layer operation followed by normalization and activation. (c) Overview of the data processing flow within the CNN-based model framework. The blue dotted area represents the data processing pipeline for *Klebsiella pneumoniae* WGS data, while the green dotted area represents the pipeline for antibiotic structural data. The yellow dotted area represents the training of the model with the integrated matrix resulting from the processing pipelines of WGS and antibiotic structural data.

internal data structure of XGBoost optimized for both memory efficiency and training speed. Extensive grid search was performed to identify optimal hyperparameters by Nguyen et al.[7]. After conducting several grid search trials in our study, we confirmed that the hyperparameters reported by Nguyen et al.[7] were optimal and adopted them for our work. XGBoost models were trained wherein the 'objective' was configured as 'reg: squarederror'. The parameters 'colsample_bylevel' and 'colsample_bytree' were set to 1, 'learning_rate' to 0.0625, and 'max_depth' to 4, conducting the training over 695 epochs. Employing a 10-fold cross-validation technique yielded a set of ten distinct models (Supplementary Table 1). Next, the 'weight' feature importance of all 10-mer features was examined. 'Weight' represents the number of nodes where a feature was utilized to determine a split. This metric provides an integer value of importance for each input feature. A weight of 0 indicates that the feature was never used to determine a split in the ensemble tree, allowing to confidently discard such features with minimal loss of information. Only those 10-mer features with non-zero importance in at least one of the ten models were extracted and re-indexed from 0 to 22,208. Subsequent data preparations for the CNN-based model exclusively utilized these 22,209 10-mer sequences. This selective utilization of significant features made it possible to construct a CNN model.

Prior to the development of a CNN model, an initial trial was conducted to train the XGBoost model solely utilizing the isolated 22,209 features. Employing identical hyperparameters, a compact version of the XGBoost model was trained. The compact XGBoost model closely mirrored the performance of the original XGBoost model (Table 1 and Supplementary Table 2). These results affirmed the legitimacy of our attempts to construct a model that exclusively employs the selected features to further enhance accuracies.

Emulating the encoding procedure described in Nguyen et al.[16], the log-scaled count values of each 10-mer were one-hot encoded, forming a matrix representation of the WGS data. With 10-mer count values ranging from 0 to 4,281, a logarithmic transformation of base 1.5529 effectively rescaled these values to lie between 0 and 19.

$$\mathrm{N} = \begin{cases} NaN, & F = 0 \\ \lceil \log_{1.5529} F \rceil, & F \geq 1 \end{cases}$$

Count values for each 10-mer were subjected to a log scale transformation and subsequently rounded up. This ensured all count values to range between 0 and 19, excluding those with an original count of 0. Instances with a count value of 0 were maintained as a zero-vector of (20,), whereas all others were one-hot encoded at the index corresponding to the log-scaled count value. This transformation led to the encoding of each WGS sample as a (22209, 20) matrix, with each row corresponding to a specific 10-mer sequence and each column reflecting the rescaled frequency of the associated 10-mer (Fig. 1a).

Antibiotic isomeric SMILES strings were retrieved from the PubChem[24] database, utilizing the 'pubchempy' library, based on their chemical identifier (CID). Employing the SmilesVectorizer of the molvecgen[25] library, conversion of the isomeric SMILES string into the RDKit[26] mol object followed by one-hot character encoding yielded a (130, 20) matrix. The SMILES matrix was stacked repeatedly over itself to match the dimensions of the WGS matrix. Stacking 256 identical SMILES matrix vertically resulted in a (130×2^8, 20) matrix, which, upon cropping from index [11,071:], produced a (130×2^8 – 11071, 20) matrix, matching the dimensions of the WGS matrix (22209, 20) for subsequent addition and scaling.

Each genome-antibiotic pairing from the 32,309 samples was encoded through an additive combination of the WGS and SMILES matrices, followed by a linear scaling operation (division by 2). The rationale for adding the two matrices was to integrate genomic sequence and antibiotic structure information without disrupting their individual spatial organization. Previous studies, such as Hirohara et al.[27] demonstrated that CNNs can effectively learn chemical motifs from one-hot encoded SMILES strings without prior knowledge of important substructures. Similarly, CNNs applied to 8-mer representations of WGS data[16] confirmed the feasibility of using CNNs on k-mer representations. Other combination methods, including multiplication, concatenation, and separate CNN applications before combining at the Dense layer have been explored, but addition yielded the best performance. This is likely due to preserving the spatial organization of each matrix for effective feature extraction of sequences. To scale data points between 0 and 1, a linear scaling operation was applied. Adding two one-hot encoded matrices results in values ranging from 0 to 2, thus dividing by 2 ensured proper min-max scaling. Each of these integrated matrices was subsequently labeled with the integer equivalent to the Log2 of the laboratory-derived MIC value. This labeling framed our research as a multi-class classification task employing a

Original XGBoost a		Compact XGBoost b		8-mer CNN Model c		CNN-based Model d		Enformer-based Model e	
Raw	1-tier	Raw	1-tier	Raw	1-tier	Raw	1-tier	Raw	1-tier
0.696 ± 0.002	0.920 ± 0.001	0.697 ± 0.003	0.923 ± 0.001	0.810	0.815	0.838 ± 0.002	0.922 ± 0.001	0.770 ± 0.005	0.854 ± 0.006

Table 1. Test accuracies obtained from five distinct models. [a] Original XGBoost model built based on the hyperparameters reported by Nguyen et al.[7]. [b] Compact XGBoost model trained with identical parameters as original XGBoost but trained only on significant 10-mer features. [c] 8-mer CNN model reported by Nguyen et al.[16]. The 95% confidence interval has not been reported in the original study. [d] CNN-based model trained on integrated matrix of important 10-mers data and antibiotics SMILES data. [e] Enformer-based model trained on integrated matrix of AMR gene sequences and antibiotics SMILES data.

simple CNN framework. The MIC labels were one-hot encoded into an (11,) vector, with each cell representing integer values from −3 to +7.

CNN-based model architecture and training

The architecture of the CNN was principally modeled on the framework proposed by Nguyen et al.[16]. The input matrix of dimensions (22209, 20) was channeled through a sequence of layers, beginning with a 2D convolutional layer configured with 64 features, a kernel size of 3, and the incorporation of a Rectified Linear Unit (ReLU) activation function. Subsequently, the processed data navigated through a 2D Max-pooling layer featuring a kernel size of 2 and underwent batch normalization. A second convolutional layer, outfitted with 128 features, was followed by an additional pooling and normalization procedure. The resultant matrix was then flattened and transitioned through a dense layer, engineered with an output size of 64, and underwent further batch normalization and ReLU activation. The ensuing (64,) vector was directed through a final dense layer, culminating in an output vector of dimensions (11,) (Fig. 1b). Softmax activation coupled with categorical cross-entropy loss was employed, framing the problem as a multi-class classification task.

The training initiated with a learning rate set at 1e-5. If no enhancement in the validation loss was observed over a span of 2 epochs, the learning rate was scaled down by a factor of 0.25. Training was terminated should the validation loss remain stagnant over 4 consecutive epochs.

Enformer-based model feature encoding

Overview of the workflow of Enformer-based model is illustrated in Fig. 2b. The Enformer architecture was conceptualized by Avsec et al.[17] to predict long-range enhancer-promoter interactions spanning up to 100 kb. This deep learning design incorporates three key modules: seven convolutional blocks, eleven transformer blocks, and a culmination of cropping with a final convolutional layer[17]. In our attempt to estimate MIC values from genomic and antimicrobial datasets, we adapted the Enformer model, tailoring it to our multi-class classification objectives (Fig. 2a). While transformers excel at discerning sequence dependencies, they encounter difficulties with extensive DNA sequences. Their predominant challenge is the quadratic space complexity, $O(n^2)$, stemming from the self-attention mechanism. Enformer integrates convolutional layers, introducing pooling prior to the transformer layers, which facilitates feature extraction from condensed sequence lengths, rendering them compatible with transformers. Such a strategy markedly diminishes the memory requisites for DNA sequence analysis.

The genomic size of Kpn typically spans between 5.34 and 5.58 Mb[28]. Despite utilizing the Enformer architecture, sequences exceeding 100 kb presents substantial computational demands. Consequently, to achieve computational efficiency, our Enformer based methodology was designed to primarily focus on AMR genes. AMR genes were derived from the primary '.fna' file contigs utilizing the Resistance Gene Identifier (RGI) tool associated with the CARD[29] database. Only genes that registered as 'Perfect' or 'Strict' hits within the CARD database were incorporated into the Enformer-based model input. The identified AMR genes for each isolate, along with their detailed information, are available in the GitHub repository. From this data we discerned that a single Kpn strain's aggregated AMR genes averaged a length of 49.2 kb, with a range from 17.5 kb to a maximum of 69.9 kb. To standardize the AMR genes of each strain to a consistent length of 98,304 bp, the genes were padded with 'N' sequences. Specifically, if a given strain contained n AMR genes, n + 1 paddings of equal length were interspersed between the genes and at both ends of the sequence to achieve the designated total length. The lengths of 'N' sequences were determined by the formula:

$$\left\lfloor \frac{98{,}304 - \sum_i len\,(gene_i)}{\text{total number of genes} + 1} \right\rfloor$$

For the one-hot encoding process nucleotide bases 'A', 'C', 'G', 'T', and 'N' were respectively encoded as '[1, 0, 0, 0]', '[0, 1, 0, 0]', '[0, 0, 1, 0]', '[0, 0, 0, 1]', and '[0, 0, 0, 0]'. This procedure yields a 98,304 × 4 matrix for each of the 1,667 genomic datasets. The molecular structure of each antibiotic was initially represented as a 130 × 20 matrix, analogous to the procedure employed in the CNN-based model. This matrix underwent reshaping into a (650, 4) matrix, which was repeatedly stacked upon itself until its size exceeded 98,304 rows. Subsequent cropping delivered a matrix of dimensions (98304, 4). The genomic data matrix and antibiotic data matrix were added and linearly scaled by 2. Consistent with the CNN-based model, MIC values were encoded as an (11,) vector.

Enformer-based model architecture and training

The Enformer-based model is instantiated with a default configuration of 192 channels, two transformer layers, eight attention heads, and 'max' as the pooling type. The 'stem' section of the model comprises a sequence of convolutional, pooling, and residual layers. It commences with a 1D convolutional layer with a kernel size of 7, followed by a series of max-pooling with window size 2 and convolutional blocks. Each convolutional block comprises of a cross-replica batch normalization layer, Gaussian Error Linear Unit (GELU) activation, and a 1D convolutional layer with specified kernel size. After three convolutional blocks with kernel size 7, 3, and 3 respectively, each followed by a max-pooling layer, one last residual convolutional block with kernel size 1 and a max-pooling layer is applied. Following the stem, the model transitions into the transformer section, which contains 2 transformer blocks. Each block comprises of a residual layer of layer normalization, multihead attention module of Avsec et al.[17], and subsequent layers of layer normalization, ReLU activation, and Dense layer of output size 192. Subsequent to the transformer, the architecture integrates a final pointwise layer. This segment comprises another max-pooling module, a GELU activation function, and a flattening operation, rendering the output suitable for the subsequent dense layers. The architecture culminates by applying a final dense module. This module consists of a dense layer with 256 units, followed by a batch normalization layer, a

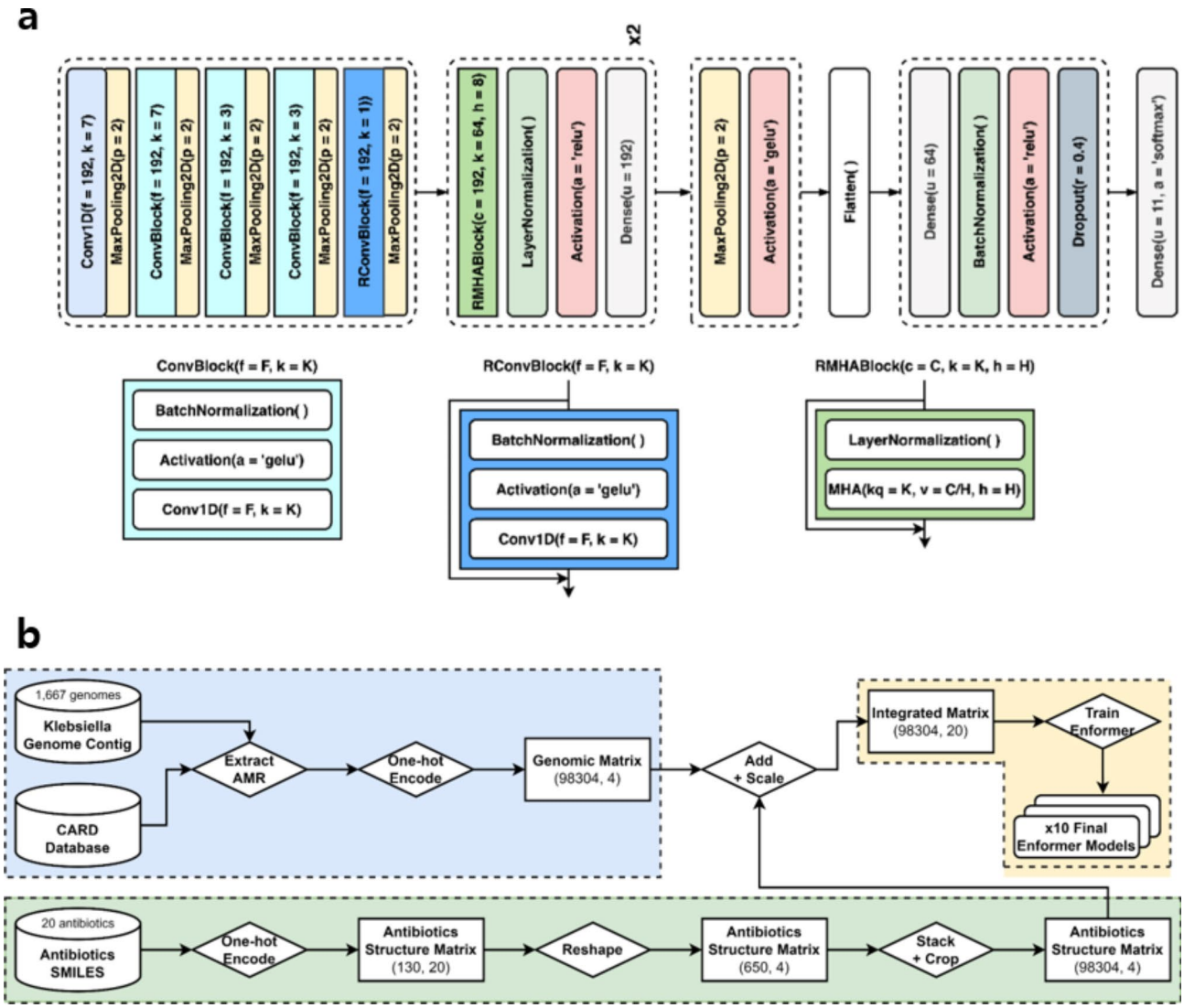

Fig. 2. Data encoding schematics in Enformer based MIC prediction. (**a**) Enformer-based model architecture: A tailored version of the Enformer architecture, originally outlined by Avsec et al.[17], adjusted for the specific needs of the MIC prediction task. Layers within dotted areas represent modular components of the overall model. The first dotted area represents the 'stem' section, a series of convolutional operations. The model then transitions into the transformer section, depicted by the second dotted area. The third dotted area prepares the output of the transformer section (pooling and activation) before it is flattened and passed into the final dense layer section to output the predicted MIC. (**b**) Overview of the data processing flow within the Enformer-based model framework. The blue dotted area represents the data processing pipeline for *Klebsiella pneumoniae* WGS data, which is notably shorter compared to Fig. 1c. The green dotted area represents the data processing pipeline for antibiotic structural data. The yellow dotted area represents the training of the model with the integrated matrix resulting from the processing pipelines of WGS and antibiotic structural data.

ReLU activation function, a dropout layer with a rate of 0.4, and concludes with an 11-unit dense layer equipped with a softmax activation function, generating the final predictions (Fig. 2a).

The model training was instantiated with an initial learning rate of 1e-14. Throughout a warm-up phase comprising two epochs and extending over 3,304 steps, the learning rate was linearly escalated to achieve a value of 0.002. Upon encountering a stagnation in the enhancement of validation 1-tier accuracy over a duration of 4 epochs, the learning rate was subsequently diminished by a factor of 0.5. Training was terminated after 100 epochs or in the event that the validation 1-tier accuracy failed to exhibit improvement for a successive span of 16 epochs.

Results

Learning curves

Utilizing a hardware setup equipped with a 256GB RAM CPU and eight 11GB GeForce RTX 2080 Ti GPUs, the CNN-based model training required an average of 21.4 epochs (Supplementary Fig. 1). The duration for these processes was approximately 4.19 h.

Under the same hardware setup, Enformer-based model training required an average of 54.8 epochs, consuming 13.23 h (Supplementary Fig. 2).

Overall model accuracies

Two accuracy metrics were employed: raw accuracy which evaluates the prediction of the precise MIC label, and 1-tier accuracy capturing predictions within a ± 1 two-fold dilution factor of the actual MIC label. Measuring accuracy within this two-fold dilution factor is consistent with the current FDA standards for diagnostic tools and established practices in clinical microbiology[7,30,31]. Utilizing 10-fold cross-validation for CNN-based models, the mean validation raw accuracy was found to be 0.846 (±0.005, 95% CI), the validation 1-tier accuracy 0.920 (±0.003, 95% CI), the test raw accuracy 0.838 (±0.002, 95% CI), and the test 1-tier accuracy 0.922 (±0.001, 95% CI).

The achieved raw accuracy of 0.838 represents a significant improvement, marking a 20% increase from the accuracy (0.696) set by the state-of-the-art XGBoost model[7]. In addition, the 1-tier accuracy closely paralleled the XGBoost model's performance at 0.922. Compared to the 8-mer CNN model, the raw accuracy of the CNN-based model increased by 3.4%, rising from 0.810. Additionally, the 1-tier accuracy showed a substantial improvement of 13%, increasing from 0.815 (Table 1 and Supplementary Table 3).

In accordance with the methodology of the CNN-based model, both raw accuracy and 1-tier accuracy were assessed in the Enformer-based model. A 10-fold cross-validation yielded a mean validation raw accuracy of 0.783 (±0.004 95% CI), a validation 1-tier accuracy of 0.866 (±0.003 95% CI), a test raw accuracy of 0.770 (±0.003 95% CI), and a test 1-tier accuracy of 0.854 (±0.004 95% CI). Despite the inability to emulate the 1-tier accuracy of the state-of-the-art XGBoost model[7], the Enformer model exhibited an 11% increase in test raw accuracy (Table 1). Compared to the 8-mer CNN model, although the Enformer-based model did not surpass the raw accuracy of the 8-mer CNN model, it demonstrated a 4.7% improvement in 1-tier accuracy (Table 1 and Supplementary Table 4).

Individual antibiotics accuracies

The CNN-based model exhibited the highest overall performance in both metrics of accuracies. Further examination was conducted to evaluate the raw and 1-tier accuracies of the CNN-based model individually across the spectrum of 20 antibiotics. The model achieved raw accuracies exceeding 0.900 for seven antibiotics, while another seven exhibited accuracies between 0.800 and 0.900. A smaller fraction, comprising two antibiotics, showed raw accuracies within the range of 0.700–0.800, followed by three antibiotics with accuracies between 0.600 and 0.700, and one with an accuracy ranging from 0.500 to 0.600. In terms of 1-tier accuracy, a significant majority, sixteen antibiotics, recorded accuracies over 0.900. Two antibiotics presented 1-tier accuracies between 0.800 and 0.900, and another two fell into the range of 0.700–0.800. Ampicillin and Cefuroxime sodium were notable standouts, each achieving 1-tier accuracies surpassing 0.990, and raw accuracies of 0.972 and 0.981, respectively. In contrast, Cefepime displayed the lowest model performance, with a raw accuracy of 0.589 and a 1-tier accuracy of 0.764 (Supplementary Tables 5 and 6).

Ablation study and comparative analysis

As compared to the model proposed by Nguyen et al.[16], our CNN-based model exhibits two primary distinctions. First, it employs 10-mer sequences as features, deviating from the 8-mers employed by Nguyen et al.[16]. Second, our model incorporates molecular structure data of antibiotics into the data matrix. To understand the factors contributing to the enhanced performance, we conducted an ablation study.

We first compared the performance of our 10-mer CNN-based model against the 8-mer model proposed by Nguyen et al.[16]. Our model demonstrated a marked improvement in performance compared to Nguyen et al.'s model, as evidenced in the pairwise comparison plot (Fig. 3a). Among the 20 antibiotics analyzed, 19 showed an increased test 1-tier accuracy relative to the previous model, with an average improvement of 11.1%p.

To quantify the impact of integrating antibiotic structural information into MIC value prediction, we created and 10-fold cross-validated 20 separate models without this information. The performance of these ablated models was then compared with the CNN-based model (Fig. 3b). Out of 20 antibiotics, 12 showed improved test 1-tier accuracies in our CNN-based model in comparison to the models without antibiotic structural data, achieving an average enhancement of 6.2%p. Notably, the most significant improvement was observed in the prediction of MIC for Nitrofurantoin, where the 1-tier accuracy surged from 51.2 to 95.7% upon inclusion of antibiotics structural data.

Susceptibility profile error rates

Given that a primary application of these deep learning tools is to predict which *Klebsiella pneumoniae* isolates are resistant, assessing sensitivity and specificity is necessary. For comprehensive comparison with previous studies[7,16], we evaluated the very major error (VME) rate and major error (ME) rate of our models. The VME rate, defined as 1 − sensitivity, represents the percentage of resistant isolates incorrectly predicted as susceptible. The ME rate, defined as 1 − specificity, represents the percentage of susceptible isolates incorrectly predicted as resistant. MIC thresholds for resistance and susceptibility were determined based on the CLSI breakpoints reported in 2023[32].

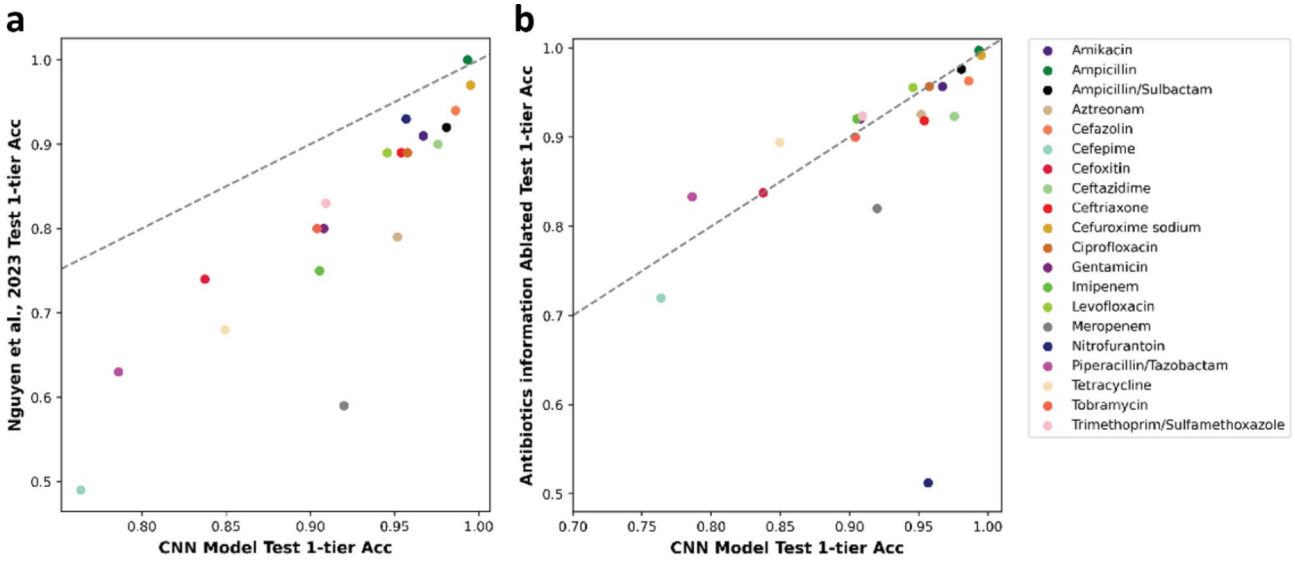

Fig. 3. Comparative analysis of model performance (**a**) Pairwise comparison plot demonstrating the enhanced performance of our 10-mer CNN-based model against the 8-mer model proposed by Nguyen et al.[16]. (**b**) Performance comparison between the CNN-based model and the antibiotics-ablated 10-mers model, indicating the effect of incorporating antibiotic structural information.

Direct comparison with previous studies is challenging due to the multiple updates to the CLSI breakpoints since 2017, which Nguyen et al.[7] used for their susceptibility profiles. Moreover, Nguyen et al.[16] did not specify which clinical breakpoint standards they utilized and did not report the overall VME/ME rates for the entire dataset, focusing instead on individual antibiotics. Additionally, Nguyen et al.[16] excluded Ampicillin from their VME/ME rate evaluation due to the limited number of susceptible samples (4 out of 1639).

Despite these limitations, a side-by-side comparison of VME and ME values for each model is provided in Table 2. The CNN-based model achieved a VME rate of 0.034, comparable to the state-of-the-art XGBoost model[7], which reported a VME rate of 0.031 (±0.003, 95% CI). The CNN-based model outperformed the XGBoost model in 6 antibiotics including Amikacin, Cefazolin, Ciprofloxacin, Levofloxacin, Nitrofurantoin, and Trimethoprim/Sulfamethoxazole (Table 2). When compared to the 8-mer CNN model, our CNN-based model showed substantial improvements across all 19 antibiotics available for comparison (Table 2).

The ME rate for the CNN-based model was 0.131, higher than the XGBoost model's ME rate of 0.037 (±0.004, 95% CI). This indicates a bias in the CNN-based model towards predicting higher MIC values, potentially due to the imbalanced dataset, where resistant data points outnumber susceptible data points by over a factor of three (Table 2). Antibiotics with an ME rate exceeding 0.5 included Ampicillin and Nitrofurantoin, with ME rates of 1.0 and 0.8, respectively. Ampicillin had only 4 susceptible datapoints out of 1639, and Nitrofurantoin had 55 out of 774, representing the lowest counts of susceptible data points among the 20 antibiotics. The bias may be attributed to framing the task as a multi-class classification rather than a regression task, as the former penalizes solely based on the correctness of predictions rather than the margin of error. Future research should focus on expanding the dataset and achieving a more balanced distribution to mitigate these biases.

Attention analysis

The Enformer architecture compresses input sequences into "bins" using convolutional layers while preserving their sequential order[17]. As a result, the output matrix from the convolutional operation (the input for the transformer) maintains the original sequence order. In the transformer layer, each row of the input matrix is treated as an embedded token. By examining the attention matrix, the level of attention allocated to each position in the original input sequence can be evaluated (Fig. 4a).

To illustrate this, the attention matrices for the genomic sequence of PATRIC ID: 573.12902 combined with the SMILES data of 20 antibiotics were analyzed. Isolate 573.12902 was randomly selected from those with data points for all 20 antibiotics. Each row and column of the attention matrix directly extracted from the model represents a 32-bp bin of the corresponding position in the input sequence (Fig. 4b and c). We calculated the mean value of each column to measure the overall attention allocated to each bin. The attention distribution across the sequence is shown in a line graph (Fig. 4d and Supplementary Fig. 3).

The resulting graphs indicate that for all 20 antibiotics, the Enformer model focuses extensively on the AMR gene regions (Supplementary Fig. 3). The input sequence comprises alternating ['N' padding + SMILES] and [AMR gene + SMILES] regions. Given that the SMILES data is uniformly distributed across the input sequence, the model's selective attention to regions containing both AMR gene and SMILES data suggests it is identifying significant positions with abundant information relevant to MIC prediction.

Positions with the highest attention peaks were further analyzed. Out of 3,072 positions, the top 15 positions (less than 0.5% of the entire sequence) with the highest attention were extracted and their corresponding

Antibiotic	Resistant	Susceptible	XGBoost[7] a		8-mer CNN[16] b		10-mer CNN c	
			VME	ME	VME	ME	VME	ME
All	22,393	7044	0.031	0.037	N/A	N/A	0.034	0.131
Amikacin	454	155	0.298	0.000	0.300	0.023	0.013	0.006
Ampicillin	1635	4	0.000	0.000	N/A	N/A	0.000	1.000
Ampicillin/ Sulbactam	1455	90	0.003	0.032	0.014	0.056	0.008	0.111
Aztreonam	1407	216	0.001	0.398	0.099	0.318	0.013	0.444
Cefazolin	1570	97	0.060	0.018	0.032	0.200	0.004	0.289
Cefepime	963	418	0.007	0.137	0.175	0.171	0.109	0.392
Cefoxitin	828	667	0.077	0.009	0.157	0.075	0.122	0.133
Ceftazidime	1488	136	0.005	0.123	0.074	0.143	0.006	0.272
Ceftriaxone	1528	80	0.000	0.188	0.0013	0.000	0.001	0.013
Cefuroxime sodium	1469	91	0.002	0.010	0.061	0.111	0.004	0.154
Ciprofloxacin	1506	0	0.005	0.025	0.028	0.100	0.000	NaN
Gentamicin	741	752	0.072	0.009	0.147	0.054	0.082	0.061
Imipenem	478	1160	0.040	0.032	0.0142	0.052	0.103	0.028
Levofloxacin	1335	0	0.016	0.020	0.039	0.000	0.000	NaN
Meropenem	481	1130	0.048	0.027	0.083	0.061	0.112	0.020
Nitrofurantoin	719	55	0.018	0.227	0.069	0.333	0.000	0.800
Piperacillin/ Tazobactam	1230	272	0.025	0.012	0.143	0.116	0.035	0.342
Tetracycline	778	739	0.114	0.008	0.195	0.095	0.175	0.153
Tobramycin	1077	566	0.040	0.012	0.151	0.051	0.095	0.081
Trimethoprim/ Sulfamethoxazole	1251	416	0.119	0.108	0.095	0.122	0.039	0.197

Table 2. VME and ME rates of three distinct models. [a] VME and ME rate of the XGBoost model reported by Nguyen et al.[7]. [b] VME and ME rate of the 8-mer CNN model reported by Nguyen et al.[16]. The values for overall antibiotics and Ampicillin were omitted. [c] VME and ME rate of the CNN-based model.

positions in the original input sequence were examined. For 18 out of the 20 antibiotics, at least one of these top 15 positions corresponded to an AMR gene related to resistance against the respective antibiotic. For example, for 573.12902 - Piperacillin/Tazobactam, 6 out of the 15 highest attention peaks corresponded to AMR genes responsible for penam resistance (*acrB*, *marA*, *KpnH*, and *pbp3*). Similarly, for 573.12902 - Ciprofloxacin, 5 out of the 15 highest attention peaks corresponded to AMR genes responsible for fluoroquinolone antibiotic resistance (*adeF*, *oqxA*, *emrR*, and *KpnG*). This demonstrates that the Enformer-based model allocates high attention to AMR genes related to the specific drug class in question.

Discussion

Approaches utilizing ML and DL can significantly enhance clinician decision-making during the empirical treatment phase, bridging the gap between specimen collection and laboratory-derived MIC outcomes from AST. Traditional AST requires culturing bacterial isolates until a discernible growth phase is reached, extending the diagnostic window by 24 to 48 h or longer for slow-growing pathogens[5]. In contrast, with a well-established sequencing pipeline, WGS data can be obtained within 8–15 h from the initiation of sample preparation, including DNA extraction[33,34]. Once the WGS data is ready and aligned, a pre-trained model can predict the MIC value almost instantaneously.

Among the methodologies examined in this study, the CNN-based model exhibited the highest performance. Notably, a raw accuracy of 0.838 represents a 20% improvement from Nguyen et al.'s[7] state-of-the-art XGBoost model. This enhancement underscores the benefit of applying CNN to 10-mer genomic features, in contrast to XGBoost model and 8-mer based CNN model. By comparing with 8-mer CNN model, we showed that high performance of our CNN-based model primarily stems from effective genomic feature selection procedure. This process enabled the construction of the CNN model using longer 10-mer genomic sequence features, which was hindered by the prohibitively high computational cost in a previous study. We also demonstrated that integrating antibiotic structural data in the model was beneficial. To our understanding, this marks the first effort to integrate antibiotic structural data into antibiotic resistance prediction. The ablation of antibiotic structural information data indicates that the adoption of 10-mers along with the incorporation of antibiotic structural data, collectively contributes to the enhancement of the model's accuracy.

While predicting MIC values within a ± 1 two-fold dilution adheres to FDA guidelines and conventional standards, this margin of variability could pose challenges for clinical application. The model's predictions are intended to support clinicians in determining the appropriate antibiotics and their concentrations for

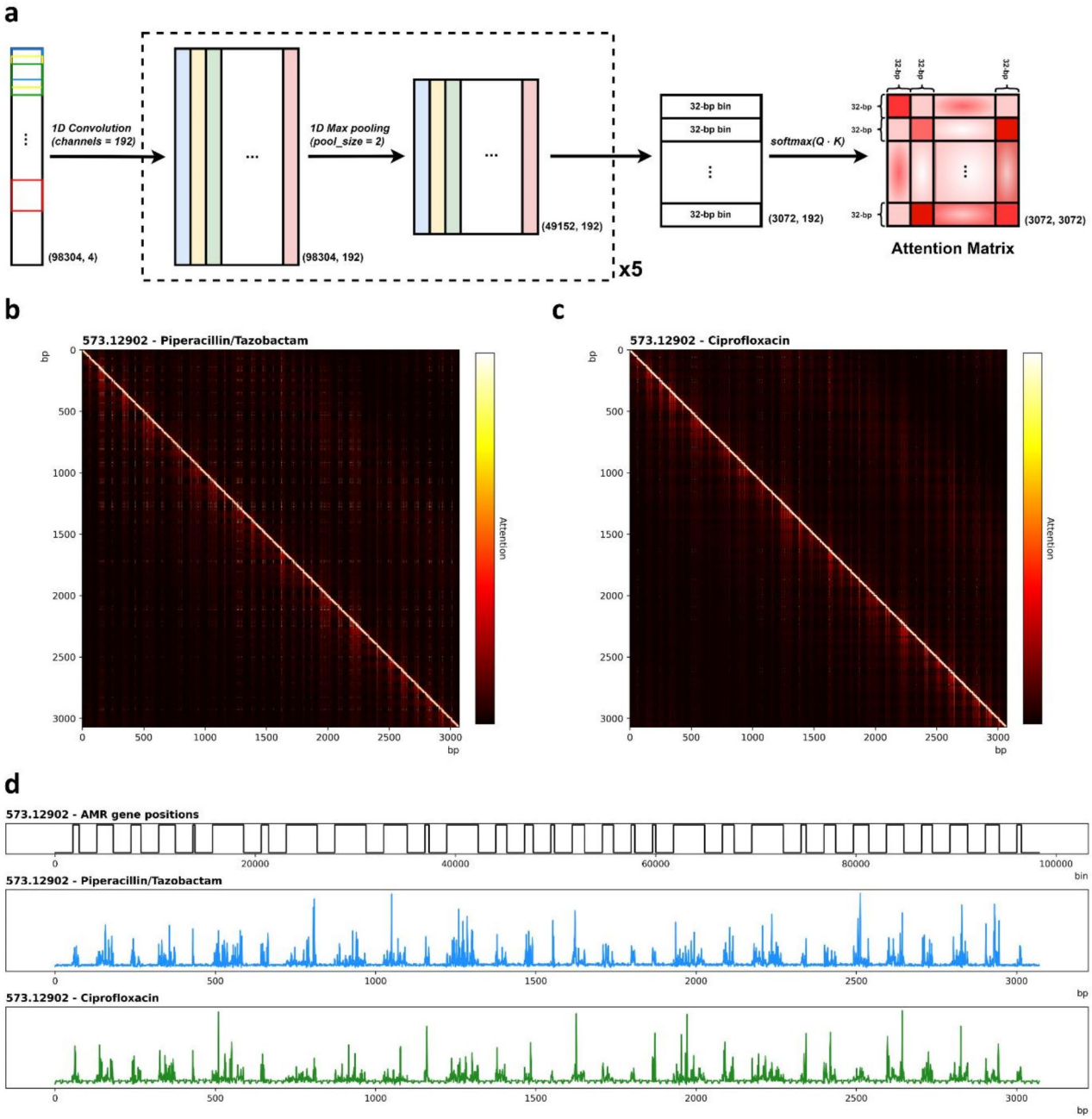

Fig. 4. Analysis of attention matrices of the Enformer-based model. (**a**) Transformation of input sequence features. After five convolution and pooling operations, the input sequence is compressed into a matrix with 1/32 the original number of rows. Since both convolution and pooling operations are one-dimensional, the sequential order of the original sequence is preserved. (**b**) Heatmap representation of the attention matrix for input 573.12902-Piperacillin/Tazobactam, showing distinct lines that indicate positions of high attention. (**c**) Heatmap representation of the attention matrix for input 573.12902-Ciprofloxacin. (**d**) Attention distribution depicted in a line graph. For AMR gene positions, plotted in black, high positions correspond to [AMR gene + SMILES] regions, while low positions correspond to ['N' padding + SMILES] regions. The mean attention for each column of the attention matrix 4b and 4c is plotted in blue and green respectively.

prescription. A substantial uptick in raw accuracy is anticipated to facilitate clinicians in leveraging these predictions for tailored patient decisions.

To evaluate the model's capability in predicting susceptibility profiles, VME and ME rates were assessed. The CNN-based model demonstrated VME rates comparable to the state-of-the-art XGBoost model. Notably, the CNN-based model significantly outperformed the 8-mer CNN model across all antibiotics (Table 2). The ME rate was higher than that of the XGBoost model. This discrepancy is likely due to the imbalance in the dataset,

with resistant data points outnumbering susceptible ones by over a factor of three. Additionally, the multi-class classification approach may be inherently more vulnerable to the effects of such imbalances. Despite these issues, we used the dataset without balancing to maintain consistency with previous studies by Nguyen et al.[7] and Nguyen et al.[16]. Utilizing the same datas allows for a direct comparison of the performance of our novel approaches with those used in earlier research. Future studies should aim to train models on more enriched and balanced datasets to effectively address this issue. Despite having a higher ME rate compared to the XGBoost model, VME is considered a more critical error than ME in clinical applications.

The CNN approach is not without limitations. The extraction of significant 10-mers to structure the genome data matrix relies on preceding XGBoost models. This prolongs the data processing pipeline and renders the model cumbersome for iterative enhancements. Relying on XGBoost's feature importance to select 10-mers adds an additional layer of opacity, impeding straightforward model interpretation. As such, extended model architectures suffer from diminished explainability and interpretability.

To address these constraints of the CNN-based model, we turned our attention to Enformers. By deploying Enformers, genomic sequences were used without extensive feature engineering, enhancing model explainability. While the Enformer-based model failed to surpass the state-of-the-art model in 1-tier accuracy, it delivered satisfactory performance in raw accuracy. It presents a noteworthy improvement, marking an 11% enhancement (Table 1).

Several factors may account for the model's subdued performance relative to the 10-mer CNN approach. Most prominently, the Enformer-based model incorporated only AMR gene data. Despite the Enformer's architecture, the entire WGS of Kpn exceeded the memory capacity of our hardware configuration. In selectively extracting AMR genes, potentially crucial data, such as AMR gene regulatory regions or even unidentified AMR genes, might have been overlooked. Furthermore, streamlining the genomic data to include only AMR genes increases the model's vulnerability to low genomic diversity within the dataset. Nguyen et al.[7] reported the phylogenetic tree of *Klebsiella pneumoniae* strains in the dataset, constructed by aligning seven housekeeping genes (*phoE*, *tonB*, *rpoB*, *pgi*, *gapA*, *mdh*, and *infB*), and analyzed the AMR gene content of each strain. They also examined the variability in MIC values across 20 antibiotics within the same MLST types. Their findings indicated that in the dataset closely related strains can exhibit significant diversity in AMR gene content, and nearly clonal strains of the same MLST type can have varying MIC values for the same antibiotic. For example, 27 isolates of the same MLST type had five different MIC values for Piperacillin/Tazobactam, while 56 isolates in another MLST type had seven different MIC values. Moreover, among the 1,667 clinical isolates, 1,133 were in the top five most common MLST types. This lack of genomic diversity across isolates may negatively impact the effectiveness of MIC prediction models, particularly the Enformer model, which relies on a streamlined genomic dataset consisting only of AMR genes. This approach increases the likelihood of similar inputs for strains with different MIC values for the same antibiotic. Additionally, the model's structural simplicity could have influenced its performance. The decision to implement five convolutional blocks and two transformer blocks was, in part, a concession to hardware constraints. As such, future endeavors that incorporate supplementary DNA regions, like the AMR gene's regulatory regions, combined with more advanced hardware to examine deeper architectures, could elevate the accuracy of the Enformer-based model.

Albeit these limitations, introducing the Enformer model in this study was aimed at demonstrating the potential of holistic models that require minimal preprocessing and can handle longer genomic sequences directly. The Enformer model was proposed as a solution to the extensive and somewhat arbitrary preprocessing pipeline required by the CNN-based model. Training 10 XGBoost models requires approximately 24 h with the hardware setup utilized in this study. Additionally, the process of splitting the WGS data into 10-mers, a number chosen partly due to hardware constraints, and then selectively extracting portions of these 10-mers fragments the genomic sequence data, adding an extra layer of complexity and opacity to the model. The ability to analyze long sequences holistically provides a more transparent model architecture and reduces the need for pre-selecting features. Analysis of attention matrices of the Enformer-based model highlights the model's transparency and interpretability. The Enformer-based model exemplifies a holistic and interpretable approach that AMR prediction model research should strive to advance. The model's promising performance, even when trained exclusively on AMR gene regions, indicates its potential to replace traditional ML or CNN models in the near future. Future research should focus on enhancing the Enformer model by including additional relevant genomic regions, such as regulatory regions, and utilizing more advanced hardware configurations to fully exploit its capabilities. In a high-stakes environment such as clinical decision-making, where outcomes can directly impact patient health, the ability of clinicians to understand and trust AI-driven recommendations is crucial[35]. Therefore, future iterations of MIC prediction models must prioritize the development of interpretable frameworks that clinicians can navigate and utilize with confidence.

Data availability

The datasets and codes generated for the CNN-based model are available at https://github.com/ByeonggyuRyu/CNN_MIC_Prediction. The datasets and codes generated for the Enformer-based model are available at https://github.com/ByeonggyuRyu/Enformer_MIC_Prediction.

Received: 26 March 2024; Accepted: 9 October 2024
Published online: 29 October 2024

References

1. Ashurst, J. V. & Dawson, A. in *StatPearls* (2023).
2. Navon-Venezia, S., Kondratyeva, K. & Carattoli, A. Klebsiella pneumoniae: a major worldwide source and shuttle for antibiotic resistance. *FEMS Microbiol. Rev.* **41**, 252–275. https://doi.org/10.1093/femsre/fux013 (2017).

3. Meletis, G. Carbapenem resistance: overview of the problem and future perspectives. *Ther. Adv. Infect. Dis.* **3**, 15–21. https://doi.org/10.1177/2049936115621709 (2016).
4. Jorgensen, J. H. & Ferraro, M. J. Antimicrobial susceptibility testing: a review of general principles and contemporary practices. *Clin. Infect. Dis.* **49**, 1749–1755. https://doi.org/10.1086/647952 (2009).
5. Tetz, G. & Tetz, V. Evaluation of a New Culture-based AtbFinder Test-System employing a Novel Nutrient Medium for the selection of optimal antibiotics for critically ill patients with Polymicrobial infections within 4 h. *Microorganisms*. **9**https://doi.org/10.3390/microorganisms9050990 (2021).
6. Laxminarayan, R. et al. Antibiotic resistance-the need for global solutions. *Lancet Infect. Dis.* **13**, 1057–1098. https://doi.org/10.1016/S1473-3099(13)70318-9 (2013).
7. Nguyen, M. et al. Developing an in silico minimum inhibitory concentration panel test for Klebsiella pneumoniae. *Sci. Rep.* **8**, 421. https://doi.org/10.1038/s41598-017-18972-w (2018).
8. Drouin, A. et al. Predictive computational phenotyping and biomarker discovery using reference-free genome comparisons. *BMC Genom.* **17**https://doi.org/10.1186/s12864-016-2889-6 (2016).
9. Davis, J. J. et al. Antimicrobial Resistance Prediction in PATRIC and RAST. *Sci. Rep.* **6**, 27930. https://doi.org/10.1038/srep27930 (2016).
10. Coelho, J. R. et al. The use of machine learning methodologies to analyse antibiotic and biocide susceptibility in Staphylococcus aureus. *PLoS One*. **8**, e55582. https://doi.org/10.1371/journal.pone.0055582 (2013).
11. Stoesser, N. et al. Predicting antimicrobial susceptibilities for Escherichia coli and Klebsiella pneumoniae isolates using whole genomic sequence data. *J. Antimicrob. Chemother.* **68**, 2234–2244. https://doi.org/10.1093/jac/dkt180 (2013).
12. Niehaus, K. E., Walker, T. M., Crook, D. W., Peto, T. E. A. & Clifton, D. A. Machine learning for the prediction of antibacterial susceptibility in Mycobacterium tuberculosis. *2014 Ieee-Embs International Conference on Biomedical and Health Informatics (Bhi)*, 618–621 (2014).
13. Bradley, P. et al. Rapid antibiotic-resistance predictions from genome sequence data for Staphylococcus aureus and Mycobacterium tuberculosis. *Nat. Commun.* **6**, 10063. https://doi.org/10.1038/ncomms10063 (2015).
14. Pesesky, M. W. et al. Evaluation of Machine Learning and rules-based approaches for Predicting Antimicrobial Resistance profiles in Gram-negative Bacilli from whole genome sequence data. *Front. Microbiol.* **7**, 1887. https://doi.org/10.3389/fmicb.2016.01887 (2016).
15. Jeukens, J. et al. Genomics of antibiotic-resistance prediction in Pseudomonas aeruginosa. *Ann. N Y Acad. Sci.* **1435**, 5–17. https://doi.org/10.1111/nyas.13358 (2019).
16. Nguyen, Q. H. et al. eMIC-AntiKP: estimating minimum inhibitory concentrations of antibiotics towards Klebsiella pneumoniae using deep learning. *Comput. Struct. Biotechnol. J.* **21**, 751–757. https://doi.org/10.1016/j.csbj.2022.12.041 (2023).
17. Avsec, Ž. et al. Effective gene expression prediction from sequence by integrating long-range interactions. *Nat. Methods*. **18**, 1196–1203 (2021).
18. Long, S. W. et al. Population Genomic Analysis of 1,777 Extended-Spectrum Beta-Lactamase-Producing < i > Klebsiella pneumoniae Isolates, Houston, Texas: Unexpected Abundance of Clonal Group 307. *mBio* 8, (2017). https://doi.org/10.1128/mbio.00489-17.
19. Long, S. et al. Whole-genome sequencing of human clinical Klebsiella pneumoniae isolates reveals misidentification and misunderstandings of Klebsiella pneumoniae, Klebsiella variicola, and Klebsiella quasipneumoniae. *Msphere*. **2**, 101128mspheredirect00290–101128mspheredirect00217 (2017).
20. Bankevich, A. et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *J. Comput. Biol.* **19**, 455–477. https://doi.org/10.1089/cmb.2012.0021 (2012).
21. Wattam, A. R. et al. Improvements to PATRIC, the all-bacterial Bioinformatics Database and Analysis Resource Center. *Nucleic Acids Res.* **45**, D535–D542. https://doi.org/10.1093/nar/gkw1017 (2017).
22. VanOeffelen, M. et al. A genomic data resource for predicting antimicrobial resistance from laboratory-derived antimicrobial susceptibility phenotypes. *Brief. Bioinform.* **22**, bbab313 (2021).
23. Deorowicz, S., Kokot, M., Grabowski, S. & Debudaj-Grabysz, A. KMC 2: fast and resource-frugal k-mer counting. *Bioinformatics*. **31**, 1569–1576. https://doi.org/10.1093/bioinformatics/btv022 (2015).
24. Kim, S. et al. PubChem 2023 update. *Nucleic Acids Res.* **51**, D1373–D1380. https://doi.org/10.1093/nar/gkac956 (2023).
25. Jannik Bjerrum, E. S. M. I. L. E. S. Enumeration as data augmentation for neural network modeling of molecules. arXiv:1703.07076 (2017). https://ui.adsabs.harvard.edu/abs/2017arXiv170307076J
26. RDKit Open-source cheminformatics. (2022). https://www.rdkit.org
27. Hirohara, M., Saito, Y., Koda, Y., Sato, K. & Sakakibara, Y. Convolutional neural network based on SMILES representation of compounds for detecting chemical motif. *BMC Bioinform.* **19**, 83–94 (2018).
28. Du, L. et al. Genome sequencing and comparative genome analysis of 6 hypervirulent Klebsiella pneumoniae strains isolated in China. *Arch. Microbiol.* **203**, 3125–3133. https://doi.org/10.1007/s00203-021-02263-0 (2021).
29. P Alcock, B. et al. CARD 2023: expanded curation, support for machine learning, and resistome prediction at the Comprehensive Antibiotic Resistance Database. *Nucleic Acids Res.* **51** (D690-D699). https://doi.org/10.1093/nar/gkac920 (2023).
30. Food & Administration, D. Guidance for industry and FDA. Class II special controls guidance document: antimicrobial susceptibility test (AST) systems. *Center for Devices and Radiological Health, Food and Drug Administration, US Department of Health and Human Services, Silver Spring, MD* (2009).
31. Jorgensen, J. H. Selection criteria for an antimicrobial susceptibility testing system. *J. Clin. Microbiol.* **31**, 2841–2844. https://doi.org/10.1128/jcm.31.11.2841-2844.1993 (1993).
32. Wayne, P. CLSI Performance Standards for Antimicrobial Susceptibility Testing. *CLSI Document Clinical Laboratory Standards Institute (CLSI): Wayne, PA, USA* (2017).
33. Goenka, S. D. et al. Accelerated identification of disease-causing variants with ultra-rapid nanopore genome sequencing. *Nat. Biotechnol.* **40**, 1035–1041 (2022).
34. Owen, M. J. et al. Rapid sequencing-based diagnosis of thiamine metabolism dysfunction syndrome. *N. Engl. J. Med.* **384**, 2159–2161 (2021).
35. Giordano, C. et al. Accessing Artificial Intelligence for clinical decision-making. *Front. Digit. Health*. **3**, 645232. https://doi.org/10.3389/fdgth.2021.645232 (2021).

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korean Government (2019M3E5D6063871, 2021M3H9A2097443, and 2022R1A2C1006609).

Author contributions

B.R. conceptualized the study, executed the experimental work, analyzed the data, and authored the manuscript. W.J. provided oversight for the research outcomes and revised the manuscript. D.K. oversaw the research findings and reviewed and edited the manuscript.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-024-75973-2.

Correspondence and requests for materials should be addressed to D.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.