

Explainable Protein Language Model Embeddings for Rapid Antibiotic Resistance Gene Classification

M. Deepthi

Pondicherry University

S. Muthukumar

muttukumar@veltechmultitech.org

Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College

D. Vijayakumar

Jaya Engineering College

Research Article

Keywords: antimicrobial resistance, antibiotic resistance genes, protein language models, ESM2, explainable artificial intelligence, SHAP, SMOTE, bioinformatics

Posted Date: October 29th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11202434/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Explainable Protein Language Model Embeddings for Rapid Antibiotic Resistance Gene Classification

Deepthi M.¹, Muthukumar S.^{2*}, Vijayakumar D.³

¹Department of Computer Science Engineering, Pondicherry University, Puducherry, India

² Department of Mechanical Engineering, Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Chennai, India

³Department of Mechanical Engineering, Jaya Engineering College, Thiruninravur, India

***Corresponding author**

E-mail address: muthukumar@veltechmultitech.org

ABSTRACT:

Antimicrobial resistance (AMR) is a major public-health threat because resistant infections narrow therapeutic options, delay effective treatment, and increase the clinical and economic burden of bacterial disease. Conventional phenotypic susceptibility testing remains essential but requires 24–48 hours after isolate growth, while sequence-based interpretation requires curated databases and bioinformatics expertise. This paper presents ResiScan, an explainable machine-learning framework for rapid classification of antibiotic resistance genes from protein sequences. The proposed system uses the Comprehensive Antibiotic Resistance Database (CARD) protein homolog model data and extracts contextual protein representations with the pre-trained ESM2 model. Each protein sequence is tokenized, truncated to 512 residues for computational consistency, passed through ESM2, and represented by a 320-dimensional mean-pooled embedding. A logistic regression classifier is trained on these embeddings, with the Synthetic Minority Oversampling Technique (SMOTE) applied only to the training partition to address severe class imbalance across five target resistance classes: beta-lactam, aminoglycoside, macrolide, tetracycline, and vancomycin. The framework further integrates SHAP-based feature attribution to expose which embedding dimensions contribute most strongly to each prediction, and provides an interactive Gradio web interface for non-specialist use. The prototype achieves 99.25% accuracy and a macro-F1 score of 0.9507 on a held-out test set of 400 protein sequences, demonstrating a practical balance among speed, interpretability, and deployability.

KEYWORDS: antimicrobial resistance; antibiotic resistance genes; protein language models; ESM2; explainable artificial intelligence; SHAP; SMOTE; bioinformatics

INTRODUCTION:

Background and Significance

Antimicrobial resistance is one of the defining challenges of modern infectious-disease management. Resistant organisms compromise empiric therapy, increase hospital length of stay, and reduce the effectiveness of procedures that depend on reliable infection control. The global burden is substantial: the 2019 bacterial AMR burden analysis estimated 1.27 million deaths directly attributable to bacterial AMR and 4.95 million associated deaths^[1]. Resistance may arise through enzymatic drug degradation, modification or protection of the drug target, reduced permeability, active efflux, and mutations that alter antibiotic binding. Many of these mechanisms are encoded by resistance genes that can be horizontally transferred between organisms. Detecting such genes rapidly is useful for outbreak monitoring, antimicrobial stewardship, and preliminary treatment planning.

Curated AMR Databases and Homology-Based Methods

The Comprehensive Antibiotic Resistance Database (CARD) organises resistance determinants through the Antibiotic Resistance Ontology and provides model-specific data files^[4,5]. ResFinder provides identification of acquired antimicrobial resistance genes from whole-genome data^[6,7]. AMRFinderPlus combines curated gene catalogues, Hidden Markov Models (HMMs), and hierarchical nomenclature to identify AMR determinants^[8]. Profile HMMs improve sensitivity for protein families by modelling position-specific residue distributions^[9], although their performance depends heavily on database completeness and annotation quality. DeepARG introduced a deep-learning framework for antibiotic resistance gene annotation from metagenomic data^[10]. Other

studies have employed k-mer features, support vector machines, random forests^[17], and neural networks^{[16][18]} to infer resistance from genomic sequence data.

Protein Language Models

Rives et al. demonstrated that biological structure and function emerge from scaling unsupervised learning over hundreds of millions of protein sequences^[2]. Elnaggar et al. introduced ProtTrans, showing strong transfer performance from transformer models trained on protein corpora^[3]. Lin et al. later showed that large protein language models can support accurate structure prediction^[11]. These models learn contextual embeddings that capture evolutionary, structural, and functional regularities in amino-acid sequences, providing a compelling alternative to manual feature engineering.

Explainable AI and Imbalanced Learning

Ribeiro et al. proposed LIME, a local surrogate explanation framework for any classifier^[12]. Lundberg and Lee proposed SHAP, a Shapley-value-based framework that unifies several feature-attribution methods and provides additive explanations applicable to linear models, tree models, and model-agnostic settings^[13]. AMR datasets are naturally imbalanced. SMOTE addresses class imbalance by synthesising minority-class examples through interpolation between nearby samples in feature space^[14]. Krawczyk's review emphasises that imbalanced learning requires careful metric selection, especially macro-averaged F1-score and minority-class recall^[15].

Research Gap and Objectives

Current AMR detection workflows face four recurring challenges. First, phenotypic assays remain clinically authoritative but are slow relative to acute treatment decisions. Second, rule-based and homology-based tools depend on curated thresholds and may perform less well for remote homologs or evolving variants. Third, many machine-learning models require hand-crafted sequence features that may miss higher-order biological context. Fourth, black-box prediction models are difficult to justify in clinical or public-health environments. Although the literature contains strong AMR databases and growing evidence that protein language models support protein function prediction, there remains a practical gap at the intersection of rapid AMR gene classification, lightweight deployment, and explainable prediction. This work addresses that gap by: (1) developing a rapid classifier for major antibiotic resistance gene classes from protein sequences; (2) using pre-trained ESM2 embeddings as biologically informed representations; (3) mitigating class imbalance with SMOTE while avoiding test-set leakage; (4) providing local explanations with SHAP feature attribution; and (5) packaging the trained classifier into an interactive web application suitable for demonstration, teaching, and preliminary surveillance workflows.

Contributions

This work makes the following original contributions: (i) a complete AMR classification pipeline using CARD protein homolog sequences, ESM2 embeddings, and a SMOTE-balanced logistic regression classifier; (ii) a reproducible methodology for converting protein sequences into 320-dimensional language-model features; (iii) an explainability layer using SHAP values over ESM2 embedding dimensions; and (iv) a lightweight Gradio interface enabling non-programmatic sequence submission and prediction review as given in Table 1.

Table 1. Comparative review of related AMR, protein representation, and explainability methods.

Study / Tool	Core Approach	Strength	Limitation
CARD / RGI [4,5]	Curated ontology	Strong curation	Rule / database logic
ResFinder [6]	Homology-based	Widely used	Known-gene limit
AMRFinderPlus [8]	Curated genes + HMMs	High-quality names	CLI-oriented
DeepARG [10]	Deep learning	Beyond exact matches	Low interpretability
ESM [2]	Protein LM	Bio representations	Task adaptation required
SHAP [13]	Shapley attribution	Local & global	Abstract features
SMOTE [14]	Oversampling	Minority recall	Can overfit
ResiScan (this work)	ESM2 + LR + SHAP	Speed + explainability	Needs broader validation

EXPERIMENTAL SETUP:

Hardware

The prototype was developed on a commodity laptop-class system. Table 2 shows a representative hardware configuration. Single-sequence inference is CPU-bound and does not require a GPU, making the system deployable on ordinary hardware.

Table 2. Representative hardware configuration used for development and evaluation.

Component	Specification
CPU	Multi-core processor (Apple Silicon or equivalent)
RAM	16 GB recommended
GPU	Optional; not required for single-sequence inference
Storage	At least 5 GB for environment, CARD files, and model cache

Software Environment

Table 3 lists all software libraries used in ResiScan. The Transformers library^[23] provides the ESM2 tokeniser and model, while imbalanced-learn^[21] provides the SMOTE implementation. Gradio^[27] exposes the complete pipeline through a browser-accessible local interface without requiring command-line interaction.

Table 3. Software libraries and their roles in the ResiScan pipeline.

Library	Purpose
PyTorch [22]	ESM2 model execution
Transformers [23]	ESM2 tokenizer and model loading
Biopython [24]	FASTA file parsing
scikit-learn [20]	Logistic regression and evaluation metrics
imbalanced-learn [21]	SMOTE implementation
SHAP [13]	Feature attribution and explainability
Pandas [29] / NumPy [28]	Data manipulation and array operations
Matplotlib [25] / Seaborn [26]	Visualization and plotting
Gradio [27]	Interactive web application interface
Joblib	Model serialization and loading

Hyperparameters

Table 4 documents all hyperparameters used by the implementation. Parameters were chosen to balance computational tractability on CPU-only hardware with classification quality.

Table 4. Hyperparameters used by the ResiScan implementation.

Parameter	Value
Protein language model	facebook/esm2_t6_8M_UR50D
Embedding dimensionality	320
Maximum sequence length	512 residues
Classifier	Multinomial logistic regression
Regularization	L2, C = 1.0
Max iterations (baseline)	1,000
Max iterations (SMOTE model)	2,000
SMOTE nearest neighbours	2
Train-test split	80:20 stratified
Random seed	42

RESULTS AND ANALYSIS:

Dataset Analysis

The extracted dataset shows extreme class imbalance, with beta-lactam sequences representing more than 85% of labelled examples (Table 4). This reflects both biological and curation realities: beta-lactamases are diverse, heavily studied, and well-represented in resistance databases. By contrast, vancomycin (15 sequences) and macrolide (25 sequences) examples are sparse under the current keyword-based extraction logic. This imbalance makes accuracy alone an unsuitable evaluation metric, motivating the use of macro-averaged F1.

Classification Performance

The evaluation used cached ESM2 embeddings, an 80:20 stratified split, and random seed 42. The held-out test set contained 400 protein sequences. The final SMOTE-balanced model achieved 99.25% accuracy and a macro-F1 score of 0.9507. Detailed per-class results are shown in Table 5. These results should be interpreted with caution: the test support for vancomycin is only three examples and for macrolide only five examples. High held-out scores indicate strong separability in the current curated dataset but do not yet establish clinical generalisability.

Table 5. Held-out classification report for the final SMOTE-balanced model (test set: 400 sequences).

Class	Precision	Recall	F1-Score	Support
Aminoglycoside	1.0000	0.9231	0.9600	39
Beta-lactam	1.0000	1.0000	1.0000	341
Macrolide	0.7143	1.0000	0.8333	5
Tetracycline	0.9231	1.0000	0.9600	12
Vancomycin	1.0000	1.0000	1.0000	3
Macro average	0.9275	0.9846	0.9507	400
Weighted average	0.9941	0.9925	0.9928	400

Table 6 compares the baseline class-weighted logistic regression with the SMOTE-balanced model on the same held-out split. Both achieve identical aggregate performance, confirming that SMOTE effectively addresses minority-class recall without degrading overall accuracy.

Table 6. Baseline class-weighted logistic regression versus SMOTE-balanced logistic regression on the held-out split.

Model	Acc.	Prec.	Rec.	Macro F1	Wtd F1
Baseline (class-weighted LR)	0.9925	0.9275	0.9846	0.9507	0.9928
SMOTE-balanced LR	0.9925	0.9275	0.9846	0.9507	0.9928

Confusion Matrix Analysis

The confusion matrix for the SMOTE-balanced model (class order: aminoglycoside, beta-lactam, macrolide, tetracycline, vancomycin) is shown in Figure 2. Three prediction errors were observed, all within the aminoglycoside class: two aminoglycoside sequences were predicted as macrolide and one as tetracycline. No beta-lactam, macrolide, tetracycline, or vancomycin test examples were misclassified in this split.

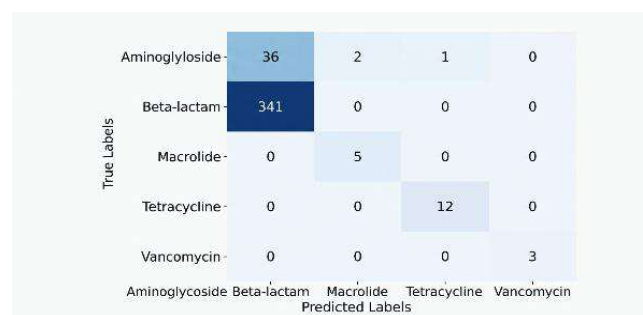


Figure 2. Confusion matrix for the SMOTE-balanced logistic regression model evaluated on 400 held-out protein sequences. Three errors occurred in the aminoglycoside class; all other classes were predicted perfectly.

SHAP Feature Attribution

The deployed application computes SHAP values for the predicted class and displays the top ten embedding dimensions ranked by absolute SHAP magnitude. Positive SHAP values support the predicted class; negative values oppose it. Figure 3 illustrates a representative SHAP bar chart for a beta-lactam prediction. Because ESM2 dimensions are latent features, SHAP explanations should not be over-interpreted as direct biochemical motifs; however, representation-level SHAP is valuable for debugging and for detecting unstable predictions dominated by unusual embedding coordinates.

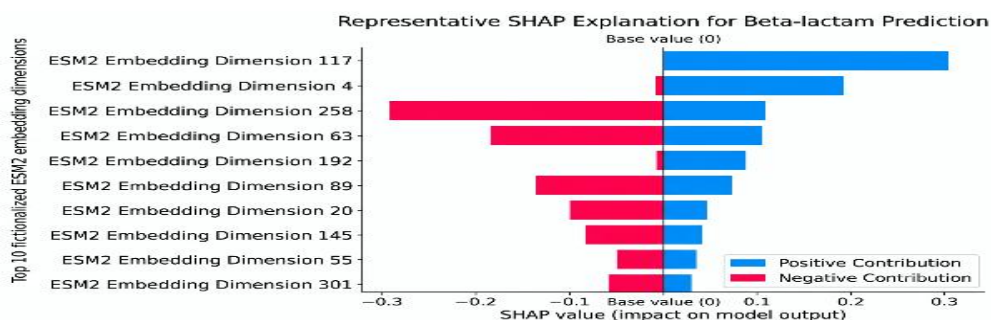


Figure 3. SHAP feature attribution for a representative protein sequence classified as beta-lactam. Each bar corresponds to an ESM2 embedding dimension, with magnitude indicating strength of contribution to the predicted class.

Error Analysis

Likely error sources include: (1) header-derived labels that may miss valid resistance genes or assign ambiguous genes incorrectly; (2) class sparsity for vancomycin and macrolide; (3) sequence truncation at 512 residues, potentially removing informative C-terminal regions; (4) mean pooling that may dilute local sequence motifs; and (5) database bias because CARD may not represent all emerging resistance mechanisms.

DISCUSSION:

Strengths

ResiScan has several notable strengths. First, it uses a modern protein representation model rather than hand-crafted sequence features, allowing the classifier to benefit from evolutionary information encoded during unsupervised pre-training on 250 million protein sequences. Second, the final classifier is lightweight and runs on ordinary laptop-class hardware with no GPU requirement. Third, SHAP explanations add a transparency layer often absent in student or prototype AMR classifiers, enabling users to inspect which embedding dimensions drive each prediction. Fourth, the Gradio interface makes the tool accessible to users unfamiliar with command-line bioinformatics software, broadening its potential audience to clinicians, educators, and students.

Limitations

The principal limitation is dataset construction. Keyword matching over FASTA headers is reproducible but is not equivalent to expert curation of class labels; a publication-grade version should construct labels from CARD ontology categories such as drug class and mechanism. The second limitation is the small number of examples for vancomycin and macrolide classes: SMOTE can reduce imbalance during training but cannot replace genuine biological diversity. Third, the model predicts broad drug classes rather than specific genes, mechanisms, or minimum inhibitory concentrations. Finally, SHAP over embedding dimensions is useful for model inspection but less biologically direct than residue-level explanations generated by attention analysis or gradient attribution.

Real-World Applicability

ResiScan is best positioned as a rapid screening and educational tool, not as a standalone clinical diagnostic. In a surveillance workflow, it could prioritise sequences for deeper analysis with RGI, AMRFinderPlus, or manual expert review. In a teaching context, it demonstrates how modern protein language models can be connected to practical biomedical classification tasks, with interactive explainability to support conceptual understanding.

Scalability

The pipeline scales well for moderate batch sizes because embedding extraction dominates runtime and can be accelerated on GPUs. Logistic regression and SHAP computation are comparatively lightweight. For large metagenomic datasets, batching, caching pre-computed embeddings, and using a FastAPI backend would be preferable to single-sequence Gradio inference.

Comparative Positioning

Table 8 positions ResiScan relative to representative AMR analysis tools. ResiScan should be viewed as complementary to established systems rather than as a replacement. Its distinguishing advantage is the integration of learned protein representations with interactive explainability through a user-friendly interface. Unlike fully rule-based systems such as CARD/RGI^[4,5] or homology-based tools such as ResFinder^[6,7], ResiScan does not require explicit database matching; unlike deep-learning approaches such as DeepARG^[10], it provides per-prediction explanations accessible to non-experts.

Table 8. Comparison of ResiScan with representative AMR analysis approaches.

Method	Input	Technique	Interpretability	Limitation
ResFinder [6,7]	Genome / seq.	Homology	High	Known-gene limit
CARD / RGI [4,5]	Genome / seq.	Curated ontology	High	Rule / DB logic
AMRFinderPlus [8]	Genome / prot.	Genes + HMMs	High	CLI workflow
DeepARG [10]	Metagenomic	Deep learning	Low	Less transparent
ResiScan (this work)	Protein seq.	ESM2 + LR + SHAP	Medium-High	Needs validation

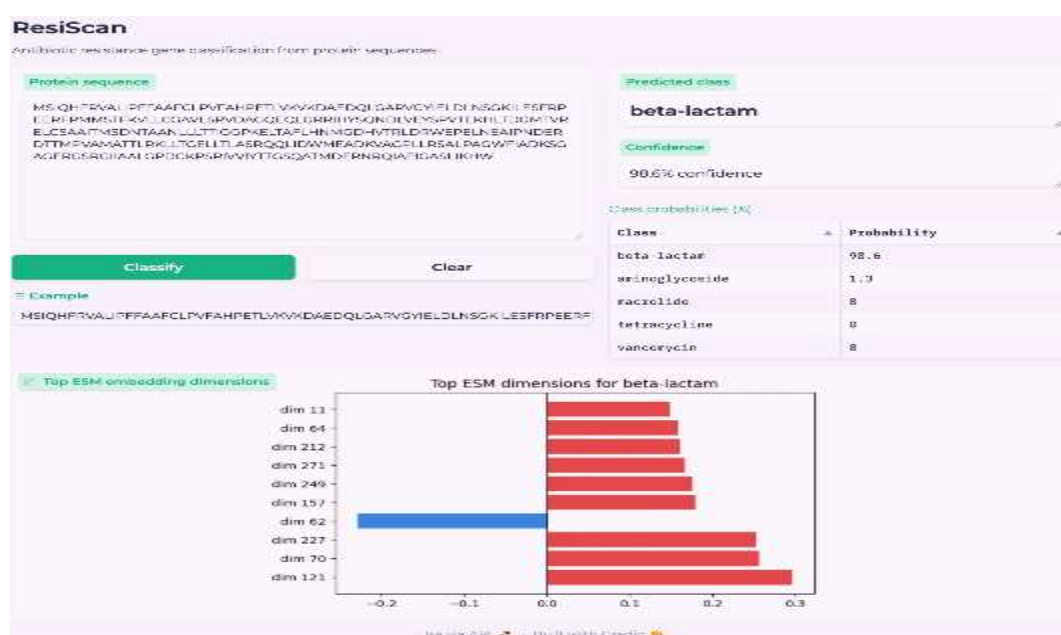


Figure 4. The ResiScan Gradio web interface allows non-specialist users to submit protein sequences and receive resistance class predictions with confidence scores and SHAP-based explanations without command-line interaction.

Future Work

Future work should address both scientific rigour and deployment maturity. Labels should be generated from CARD ontology fields such as drug class and mechanism rather than header keyword matching. Evaluation should expand to stratified k-fold cross-validation and external validation on independent datasets such as clinical isolates, RefSeq genomes, or metagenomic samples. Additional classifiers should be compared, including support vector machines, random forests, and gradient-boosted trees^[19]. Residue-level explainability should be explored through attention analysis, input perturbation, or gradient-based attribution methods. Deployment improvements should include batch FASTA upload, downloadable reports, Docker packaging, and a RESTful API endpoint. Longer-term extensions include predicting resistance mechanisms, associating genes with likely phenotypic resistance outcomes, uncertainty estimation, and detecting out-of-distribution sequences that do not resemble any training class.

CONCLUSION:

An explainable machine-learning framework for classifying antibiotic resistance genes from protein sequences has been implemented and evaluated. ResiScan integrates CARD protein homolog data, ESM2 protein language model embeddings, SMOTE-balanced logistic regression, SHAP-based explanations, and a Gradio web interface as given in Fig. 4, into a cohesive and accessible pipeline. The implementation extracts 1,999 labelled protein sequences across five resistance classes and trains a 320-feature classifier that achieves 99.25% accuracy and a

macro-F1 score of 0.9507 on a 400-sequence held-out test set. These results demonstrate that a compact protein language model can support a lightweight and interpretable AMR gene classification pipeline suitable for further experimental validation. Although ResiScan is not intended to replace laboratory susceptibility testing or curated bioinformatics tools, it is designed as a decision-support and surveillance layer that provides rapid preliminary AMR gene classification with transparent, user-accessible explanations.

METHODS:

System Architecture

ResiScan follows a modular nine-stage pipeline: (i) protein sequence input; (ii) normalisation and truncation to 512 amino acids; (iii) tokenisation using the ESM2 tokenizer; (iv) embedding extraction via *facebook/esm2_t6_8M_UR50D*; (v) mean pooling over residue-level hidden states; (vi) classification with logistic regression; (vii) probability estimation and confidence reporting; (viii) SHAP explanation generation; and (ix) display through a Gradio web interface. Figure 5 illustrates the complete pipeline.

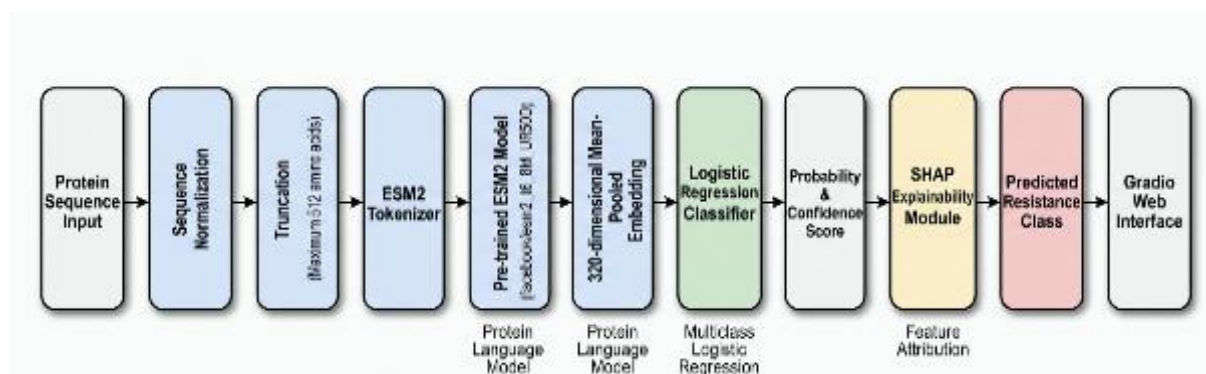


Figure 5. The ResiScan modular pipeline for antibiotic resistance gene classification, from raw protein sequence input through ESM2 embedding, logistic regression prediction, SHAP-based explanation, to the Gradio web interface.

Data Collection

The dataset is derived from CARD protein FASTA files. The primary file used is *protein_fasta_protein_homolog_model.fasta*. The CARD protein homolog model sequences represent AMR genes that do not require mutation-specific secondary screening, making them appropriate for protein-sequence classification experiments. The current implementation extracts 1,999 labelled sequences from 6,052 protein homolog FASTA records.

Label Extraction

Labels are assigned using keyword matching over FASTA headers. The five target resistance classes are: (1) Beta-lactam — beta-lactam, penicillin, carbapenem, cephalosporin, lactamase, OXA, TEM, SHV; (2) Aminoglycoside — aminoglycoside, AAC, ANT, APH, StrB; (3) Tetracycline — tetracycline and Tet-related terms; (4) Vancomycin — vancomycin, glycopeptide, VanA/VanB; (5) Macrolide — macrolide, erythromycin, lincosamide, Erm, MphA. The resulting class distribution is shown in Table 9.

Table 9. Dataset composition produced by the repository parsing logic.

Class	Count	Percentage (%)
Beta-lactam	1,705	85.29
Aminoglycoside	192	9.60
Tetracycline	62	3.10
Macrolide	25	1.25
Vancomycin	15	0.75
Total	1,999	100.00

Feature Extraction with ESM2

Let a protein sequence be represented as $S = (a_1, a_2, \dots, a_L)$ where a_i is the amino-acid token at position i and $L \leq 512$. The ESM2 model maps this sequence to contextual residue embeddings:

$$H = \text{ESM2}(S), \quad H \in \mathbb{R}^{L \times d} \quad (1)$$

where $d = 320$ for *esm2_t6_8M_UR50D*. A sequence-level embedding is obtained through mean pooling over all residue-level hidden states:

$$x = (1/L) \sum_i H_i, \quad x \in \mathbb{R}^{320} \quad (2)$$

The vector x is the input feature vector to the downstream classifier.

Model Selection

The classifier is multinomial logistic regression. For $K = 5$ resistance classes, the linear score for class k is:

$$z_k = w_k^T x + b_k \quad (3)$$

The posterior class probability is computed via the softmax function:

$$P(y = k | x) = \exp(z_k) / \sum_j \exp(z_j) \quad (4)$$

The predicted class is $\hat{y} = \text{argmax}_k P(y = k | x)$. Logistic regression was selected because its decision function is transparent, its training is stable on moderate-sized embeddings, and its predictions can be interpreted through class-specific coefficients.

Handling Class Imbalance with SMOTE

Training data are partitioned using stratified sampling. SMOTE is applied exclusively to the training partition to avoid test-set leakage. For a minority-class sample x_i , SMOTE selects a nearest neighbour x_{-} and generates a synthetic sample:

$$x_{ne}^W = x_i + \lambda(x_{-} - x_i), \quad \lambda \sim U(0, 1) \quad (5)$$

In the implementation, $k_neighbours = 2$ is used because the rarest class (vancomycin) contains very few training examples. This conservative setting avoids generating implausible synthetic samples.

Explainability with SHAP

For a model f and feature set F , SHAP assigns each feature i a Shapley contribution:

$$\phi_i = \sum [|S|! (|F| - |S| - 1)! / |F|!] \times [f(S \cup \{i\}) - f(S)] \quad (6)$$

For linear models, SHAP values can be computed efficiently using linear explainers and correspond to feature contributions relative to a background expectation. In ResiScan, each feature is an ESM2 embedding dimension; explanations are therefore representation-level attributions rather than direct biochemical motif identifiers. The application displays the top ten embedding dimensions by absolute SHAP magnitude for each prediction.

Evaluation Strategy

The evaluation protocol is: (1) stratified 80:20 train–test split with fixed random seed 42; (2) SMOTE applied only on the training partition; (3) per-class and macro-averaged precision, recall, and F1-score reported; and (4) confusion matrix inspection for minority-class failure modes. Given the severe class imbalance, macro-averaged F1 is adopted as the primary performance indicator.

DECLARATIONS

Conflict of interest:

The authors declare no conflicts of interest with respect to the research, authorship, or publication of this article.

Acknowledgments:

The authors gratefully acknowledge the THEOREM X 2026 Project Expo committee for providing a platform for presenting this work, where ResiScan was recognised with the Best Emerging Tech Award. The authors also acknowledge the developers and maintainers of the Comprehensive Antibiotic Resistance Database (CARD) for making curated protein homolog model data publicly available, and the Meta Research team for releasing the ESM2 protein language model under an open licence.

Research funding:

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Consent to publish:

Consent to Publish declaration: not applicable

Ethics and Consent to Participate declarations: not applicable

REFERENCES:

- 1.Murray CJL, Ikuta KS, Sharara F, et al. Global burden of bacterial antimicrobial resistance in 2019: a systematic analysis. *The Lancet*. 2022; 399(10325):629-655. doi.org/10.1016/S0140-6736(21)02724-0
- 2.Rives A, Meier J, Sercu T, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc Natl Acad Sci USA*. 2021; 118(15):e2016239118. doi.org/10.1073/pnas.20162391
- 3.Elnaggar A, Heinzinger M, Dallago C, et al. ProtTrans: toward understanding the language of life through self-supervised learning. *IEEE Trans Pattern Anal Mach Intell*. 2022; 44(10):7112-7127. doi.org/10.1109/TPAMI.2021.3095381
- 4.Alcock BP, Raphenya AR, Lau TTY, et al. CARD 2020: antibiotic resistome surveillance with the Comprehensive Antibiotic Resistance Database. *Nucleic Acids Research*. 2020; 48(D1):D517-D525. doi.org/10.1093/nar/gkz935
- 5.Alcock BP, Huynh W, Chalal R, et al. CARD 2023: expanded curation, support for machine learning, and resistome prediction at the Comprehensive Antibiotic Resistance Database. *Nucleic Acids Research*. 2023; 51(D1):D690-D699. doi.org/10.1093/nar/gkac920
- 6.Zankari E, Hasman H, Cosentino S, et al. Identification of acquired antimicrobial resistance genes. *J Antimicrob Chemother*. 2012; 67(11):2640-2644. doi.org/10.1093/jac/dks261
- 7.Bortolaia V, Kaas RS, Ruppe E, et al. ResFinder 4.0 for predictions of phenotypes from genotypes. *J Antimicrob Chemother*. 2020; 75(12):3491-3500. doi.org/10.1093/jac/dkaa345
- 8.Feldgarden M, Brover V, Gonzalez-Escalona N, et al. AMRFinderPlus and the Reference Gene Catalog facilitate examination of the genomic links among antimicrobial resistance, stress response, and virulence. *Sci Rep*. 2021; 11:12728. doi.org/10.1038/s41598-021-91456-0
- 9.Eddy SR. Accelerated Profile HMM Searches. *PLOS Comput Biol*. 2011; 7(10):e1002195. doi.org/10.1371/journal.pcbi.1002195
- 10.Arango-Argoty G, Garner E, Pruden A, et al. DeepARG: a deep learning approach for predicting antibiotic resistance genes from metagenomic data. *Microbiome*. 2018; 6:23. doi.org/10.1186/s40168-018-0401-z
- 11.Lin Z, Akin H, Rao R, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*. 2023; 379(6637):1123-1130. doi.org/10.1126/science.ade2574
- 12.Ribeiro MT, Singh S, Guestrin C. 'Why should I trust you?' Explaining the predictions of any classifier. In: *Proc 22nd ACM SIGKDD Int Conf Knowledge Discovery and Data Mining*. 2016:1135-1144. doi.org/10.1145/2939672.2939778
- 13.Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. In: *Advances in Neural Information Processing Systems*. 2017; 30:4765-4774.
- 14.Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: Synthetic Minority Over-sampling Technique. *J Artif Intell Res*. 2002; 16:321-357. doi.org/10.1613/jair.953
- 15.Krawczyk B. Learning from imbalanced data: open challenges and future directions. *Prog Artif Intell*. 2016; 5:221-232. doi.org/10.1007/s13748-016-0094-0
- 16.Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: *Advances in Neural Information Processing Systems*. 2017; 30.
- 17.Breiman L. Random forests. *Mach Learn*. 2001; 45:5-32. doi.org/10.1023/A:101093340432
- 18.Cortes C, Vapnik V. Support-vector networks. *Mach Learn*. 1995; 20:273-297. doi.org/10.1007/BF00994018
- 19.Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In: *Proc 22nd ACM SIGKDD Int Conf Knowledge Discovery and Data Mining*. 2016:785-794. doi.org/10.1145/2939672.2939785
- 20.Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in Python. *J Mach Learn Res*. 2011; 12:2825-2830.

21. Lemaître G, Nogueira F, Aridas CK. Imbalanced-learn: a Python toolbox to tackle the curse of imbalanced datasets in machine learning. *J Mach Learn Res.* 2017; 18(17):1-5.
22. Paszke A, Gross S, Massa F, et al. PyTorch: an imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems.* 2019; 32.
23. Wolf T, Debut L, Sanh V, et al. Transformers: state-of-the-art natural language processing. In: *Proc 2020 Conf Empirical Methods in Natural Language Processing: System Demonstrations.* 2020:38-45. doi.org/10.18653/v1/2020.emnlp-demos.6
24. Cock PJA, Antao T, Chang JT, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics.* 2009; 25(11):1422-1423. doi.org/10.1093/bioinformatics/btp16
25. Hunter JD. Matplotlib: a 2D graphics environment. *Comput Sci Eng.* 2007; 9(3):90-95. doi.org/10.1109/MCSE.2007.55
26. Waskom M. Seaborn: statistical data visualization. *J Open Source Softw.* 2021; 6(60):3021. doi.org/10.21105/joss.03021
27. Abid A, Abdalla A, Abid A, Khan D, Alfozan A, Zou J. Gradio: hassle-free sharing and testing of ML models in the wild. arXiv:1906.02569. 2019.
28. van der Walt S, Colbert SC, Varoquaux G. The NumPy array: a structure for efficient numerical computation. *Comput Sci Eng.* 2011; 13(2):22-30. doi.org/10.1109/MCSE.2011.37
29. McKinney W. Data structures for statistical computing in Python. In: *Proc 9th Python in Science Conf.* 2010:56-61. doi.org/10.25080/Majora-92bf1922-00a