[Thesis Title Here]

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of
Master of Science (MSc)
in
[Department Name]

[Your Name]
[Student ID]
[University Name]

Month, Year

Declaration

I hereby declare that the work presented in this thesis is my own and has not been submitted elsewhere for the award of any degree.

[Your Name]
[Date]

Abstract

Acknowledgement

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Antimicrobial resistance (AMR) represents one of the most pressing global health challenges of the 21st century. The World Health Organization has identified AMR as a top-ten global public health threat, with projections estimating that bacterial infections could result in approximately 10 million deaths annually by 2050 if current trends continue [35]. In 2019 alone, antibiotic-resistant bacteria directly caused 1.27 million deaths, with an additional 4.95 million deaths associated with drug-resistant infections [8]. Critical priority pathogens including *Klebsiella pneumoniae*, *Escherichia coli*, *Pseudomonas aeruginosa*, and *Salmonella enterica* account for a substantial proportion of these deaths, with carbapenem-resistant strains posing particular clinical challenges.

Traditional culture-based antimicrobial susceptibility testing (AST) remains the gold standard for determining resistance profiles but suffers from significant limitations. Most culturable bacteria require 24–48 hours of incubation for detection, with pathogen identification adding another 2–4 hours. If AMR is suspected, phenotypic AST extends the timeline by an additional 18–24 hours, resulting in a total turnaround time of 2–4 days [1]. This delay can prove critical in severe infections where rapid administration of appropriate antimicrobials significantly improves patient outcomes—the risk of death doubles if effective antibiotics are not administered within 24 hours in cases of bacteremia [22].

The advent of whole genome sequencing (WGS) and the establishment of comprehensive resistance gene databases have enabled computational approaches to AMR prediction. Machine learning (ML) methods have emerged as powerful tools for predicting resistance phenotypes from genomic data, offering the potential for rapid, accurate predictions that could guide empirical therapy selection before traditional

AST results become available [25]. This thesis presents a resistance gene-based machine learning framework that addresses critical methodological gaps in current AMR prediction approaches, demonstrating that optimized gene-based models can achieve performance competitive with or superior to whole-genome sequence-based methods while maintaining interpretability and computational efficiency.

Figure 1.1 illustrates the difference between traditional culture-based prediction and modern ML-based genomic prediction.

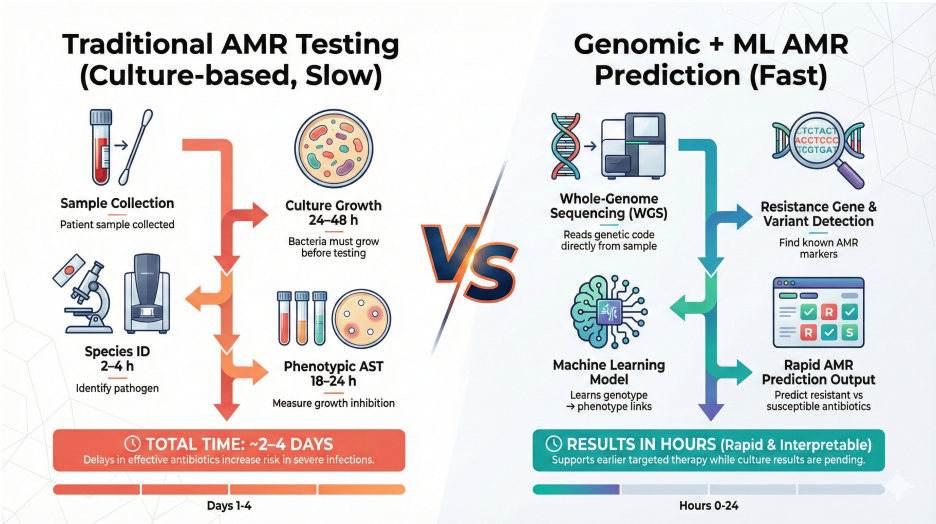

Figure 1.1: Traditional AMR testing vs genomic + ML AMR prediction

1.2 Antimicrobial Resistance: A Global Health Crisis

Antimicrobial resistance occurs when bacteria, viruses, fungi, and parasites evolve mechanisms to survive exposure to antimicrobial drugs that previously killed them or inhibited their growth. For bacteria, resistance mechanisms include enzymatic degradation (e.g., β-lactamases hydrolyzing penicillins and cephalosporins), modification of drug targets (e.g., mutations in DNA gyrase conferring fluoroquinolone resistance), efflux pump overexpression, and reduced membrane permeability [30].

Resistant infections lead to longer hospital stays, higher medical costs, and increased mortality. Patients with drug-resistant infections face a 64% higher risk of death compared to those with susceptible infections [21]. Beyond individual outcomes, AMR threatens the effectiveness of organ transplantation, cancer chemotherapy, and complex surgeries that rely on antimicrobial prophylaxis.

Economically, AMR poses global risks. The World Bank estimates that AMR could reduce global GDP by 1.1% to 3.8% by 2050, with annual costs reaching $3.4 trillion [12]. In the United States alone, resistant infections cost the healthcare system an estimated $20 billion annually, plus $35 billion in productivity losses [11].

1.3 Machine Learning Approaches for AMR Prediction

The application of machine learning to AMR prediction has evolved through distinct methodological paradigms over the past decade. Early approaches relied on rule-based systems that queried databases for known resistance genes and mutations. While these deterministic methods achieve high specificity for well-characterized mechanisms, they cannot identify novel resistance determinants absent from reference databases [18].

Classical ML models such as Random Forest, Support Vector Machines, Gradient Boosted Trees, and Logistic Regression have demonstrated strong predictive performance from genomic features, achieving accuracies ranging from 0.81 to 0.97 across pathogen–antibiotic combinations [28, 3].

Two dominant feature representation paradigms exist:

1. Resistance Gene-Based Features Binary matrices encoding presence/absence of known AMR genes, providing interpretability and computational efficiency but limited by database completeness.

2. Whole Genome Sequence (WGS)-Based Features High-dimensional features including k-mers, SNPs, and pan-genome content, capable of capturing novel mechanisms but computationally expensive and less interpretable.

Despite notable progress, gaps remain: simplistic binary encodings, weak feature selection strategies, limited handling of class imbalance, and overreliance on individual classifiers. These limitations motivate the enhanced framework proposed in this thesis.

1.4 Motivation

The motivation for this research stems from several critical observations in the current AMR prediction landscape:

- First, existing resistance gene-based approaches, while interpretable and efficient, often underperform compared to WGS-based methods. The Sunuwar & Azad framework [36] , a foundational gene-based approach, achieves F1-scores

around 0.90 but does not incorporate advanced feature engineering, hybrid selection strategies, or optimized ensemble architectures. The question arises: can methodological enhancements to the gene-based paradigm close the performance gap with WGS approaches while retaining interpretability?

- Second, WGS-based approaches, despite high reported accuracies, often exhibit a concerning accuracy-sensitivity tradeoff. The Noman et al. BioWeka framework [33] reports ≥98% accuracy but sensitivity as low as 62% for some antibiotics, meaning a substantial proportion of resistant isolates are misclassified as susceptible. In clinical contexts, such false negatives can lead to treatment failures with potentially fatal consequences. This raises the question: are high-accuracy WGS models actually superior when balanced performance metrics are considered?

- Third, the cumulative effect of resistance gene carriage remains unexplored as a predictive feature. Biological evidence suggests that isolates harboring multiple resistance genes may exhibit different resistance profiles than those with single genes due to synergistic effects, redundant protection pathways, or co-selection pressures. Yet no existing study incorporates an aggregate measure of resistance gene burden.

- Fourth, class imbalance is a common characteristic of AMR datasets, with many antibiotic-pathogen combinations exhibiting skewed resistant/susceptible ratios. Standard oversampling techniques may generate biologically implausible synthetic samples when applied to sparse binary gene matrices, and advanced hybrid methods combining oversampling with cleaning steps have received limited attention in this domain.

These observations motivate the development of an enhanced resistance gene-based framework that addresses each limitation through targeted methodological innovations: the R-Score for cumulative gene burden quantification, hybrid ANOVA-XGBoost feature selection, SMOTETomek for class imbalance handling, and the R-Blend weighted ensemble architecture.

1.5 Research Objectives

The primary objectives of this thesis are:

1. To develop and evaluate a resistance gene-based ML pipeline (R-Blend) for binary AMR prediction across multiple antibiotic–pathogen datasets.

2. To investigate whether an engineered cumulative gene-burden feature (R-Score) improves prediction performance.

3. To evaluate multiple hybrid feature selection variants (ANOVA + RF/XGB, union/intersection, different thresholds) and select an optimal configuration for resistance gene-based AMR prediction.

4. To compare several resampling strategies for class imbalance in resistance-gene data and select an effective method to improve resistant-class detection in the final model.

1.6 Challenges

The development of an effective resistance gene-based AMR prediction framework faces challenges across data and methodology dimensions.

1.6.1 Data-Related Challenges

- **Class imbalance**: Many antibiotic-pathogen datasets exhibit skewed resistant/susceptible ratios, which can bias naïve models toward the majority (susceptible) class and lead to poor detection of resistant isolates, the clinically critical class.

- **Sparse, high-dimensional gene matrices**: Resistance gene presence/absence data are binary, sparse, and often high-dimensional, making models prone to overfitting and sensitive to feature selection strategies.

- **Small dataset sizes**: Some antibiotic–pathogen combinations contain very few resistant isolates or very small sample sizes overall, leading to unstable estimates and occasional "perfect" scores that must be interpreted cautiously.

- **Genotype–phenotype mismatch**: The presence of a known resistance gene does not always translate into phenotypic resistance due to expression levels, regulatory mutations, or unknown modifiers, introducing label noise into the training data.

1.6.2 Methodological Challenges

- **Sensitivity vs. Overall Accuracy:** Clinically, missing resistant isolates (false negatives) is more serious than overcalling resistance. Designing models that

prioritize recall for the resistant class while maintaining acceptable precision and accuracy is non-trivial.

- **Identifying informative features under sparsity:** Identifying a compact but informative subset of genes from hundreds of candidates, without introducing data leakage or discarding subtle but important signals, is challenging in sparse binary spaces.

- **Handling Imbalance Safely:** Many resampling methods (e.g., SMOTE variants) can generate biologically implausible synthetic gene profiles in high-dimensional 0/1 data. Choosing a method that improves minority-class performance without distorting the data distribution requires careful evaluation.

- **Balancing Performance and Interpretability:** Powerful models like XGBoost can yield high accuracy but may operate as black boxes. The challenge is to design an approach that remains interpretable at the gene level while still being competitive with existing baselines.

1.6.3 Broader Considerations

Beyond the scope of this thesis, there are broader considerations for the eventual clinical deployment of AMR prediction models. These include generalizability across different hospitals and geographic regions, the need for prospective validation, integration with laboratory information systems, and ensuring clinician trust through transparent explanations. Additionally, resistance gene-based models inherently depend on the completeness of underlying gene annotations, meaning emerging or rare mechanisms may be underrepresented.

1.7 Contributions of This Thesis

This thesis makes the following contributions to the field of machine learning for antimicrobial resistance prediction:

- **R-Score**: A Novel Engineered Feature for Resistance Gene Burden. We introduce the R-Score, a normalized measure of cumulative resistance gene content capturing aggregate resistance-gene burden. Across the 12 Sunuwar et al. [13] datasets, adding R-Score improved average F1-score from 0.937 to 0.948 and average recall from 0.940 to 0.952, with larger gains observed on smaller datasets.

- **Hybrid ANOVA–XGBoost Feature Selection with Union-Based Integration:** We propose and evaluate hybrid feature selection variants that combine ANOVA F-test filtering ($p \leq 0.30$) with XGBoost importance (85% cumulative cutoff), using union-based integration to retain features supported by either linear or nonlinear relevance. The selected hybrid configuration consistently outperformed single-method selection across datasets.

- **Systematic Evaluation of SMOTETomek for Sparse AMR Gene Matrices:** We systematically compare multiple resampling strategies for imbalanced resistance-gene data and show that SMOTETomek provides the most stable improvement in resistant-class detection, outperforming standard over/undersampling and avoiding degradation seen in aggressive cleaning-based methods such as SMOTE-ENN.

- **R-Blend:** A Weighted Soft-Voting Ensemble Architecture. We develop and optimize R-Blend, a weighted soft-voting ensemble combining Decision Tree (weight=1), Logistic Regression (weight=1.5), and XGBoost (weight=1). The ensemble consistently improved balanced performance over individual base models and produced more stable results across heterogeneous datasets.

- **Comprehensive Benchmarking Against Gene-Based and WGS-Based Approaches:** We provide direct comparative evaluation on two benchmark settings.

    1. On the Sunuwar et al. datasets (12 antibiotic-pathogen combinations) R-Blend achieved average F1-score of 0.948, outperforming Sunuwar et al.'s [38] best classifiers by +6.3 percentage points (baseline F1: 0.885).
    2. On the Noman et al. dataset (12 antibiotics, P. aeruginosa): R-Blend achieved average F1-score of 0.959 and sensitivity of 94.1%, outperforming the WGS-based BioWeka approach by +12.6 pp F1 (BioWeka F1: 0.832) and +9.7 pp sensitivity (BioWeka: 84.4%). These results demonstrate that carefully engineered resistance gene-based models can achieve balanced performance superior to both existing gene-based frameworks and WGS-based approaches while maintaining interpretability and computational efficiency.

1.8 Organization of the Thesis

This thesis is organized into seven chapters as follows:

- **Chapter 1: Introduction**. This chapter provides an overview of the AMR crisis, introduces machine learning approaches for resistance prediction, presents the motivation and objectives of the research, discusses challenges, and outlines the contributions of this thesis.

- **Chapter 2: Theoretical Background**.This chapter presents the foundational concepts underlying this research, including antimicrobial resistance mechanisms, genomic data representation, machine learning algorithms employed, feature selection methods, class imbalance handling techniques, and ensemble learning approaches.

- **Chapter 3: Literature Review**.This chapter reviews the evolution of computational AMR prediction from rule-based detection to machine learning approaches. It covers AMR gene databases and detection tools, resistance gene-based and WGS-based prediction paradigms, and identifies critical gaps in existing methodologies that motivate the present work.

- **Chapter 4: Methodology**. This chapter presents the detailed methodology of the proposed framework, including data acquisition and preprocessing, R-Score feature engineering, hybrid ANOVA-XGBoost feature selection, SMOTETomek class imbalance handling, R-Blend ensemble architecture, experimental design, and evaluation metrics.

- **Chapter 5: Experimental Results**.This chapter presents comprehensive experimental results including dataset-level performance, ablation studies quantifying the contribution of each pipeline component, and comparative analysis against the Sunuwar & Azad and Noman et al. approaches.

- **Chapter 6: Conclusion and Future Work**. This chapter summarizes the key findings, discusses limitations of the current work, and outlines directions for future research.

- **Chapter 7: References**.This chapter provides the complete list of references cited throughout the thesis.

1.9 Summary

Antimicrobial resistance poses a critical threat to global health, with millions of deaths attributed annually to drug-resistant infections. Traditional AST methods, while reliable, are too slow to guide early empirical therapy in severe infections. Machine

learning approaches offer the potential for rapid AMR prediction from genomic data, but existing methods suffer from methodological limitations including lack of aggregate feature engineering, primitive feature selection, inadequate class imbalance handling, and suboptimal ensemble architectures.

This thesis addresses these gaps through a comprehensive resistance gene-based machine learning framework incorporating four key innovations: the R-Score for cumulative gene burden quantification, hybrid ANOVA-XGBoost feature selection with union-based integration, SMOTETomek resampling for class imbalance, and the R-Blend weighted soft-voting ensemble. The framework is evaluated across 12 antibiotic–pathogen combinations spanning carbapenems, aminoglycosides, and clindamycin resistance in K. pneumoniae, E. coli/Shigella, P. aeruginosa, S. enterica, and C. jejuni.

The proposed approach achieves an average F1-score of 0.95, outperforming both the baseline gene-based framework of Sunuwar & Azad (+5.5 percentage points F1) and the WGS-based BioWeka approach of Noman et al. (+12.7 percentage points F1). These results demonstrate that carefully engineered resistance gene-based models can achieve balanced performance superior to whole-genome approaches while maintaining the interpretability and computational efficiency essential for practical application.

Chapter 2

THEORETICAL BACKGROUND

2.1 Overview

Antimicrobial Resistance (AMR) prediction from genomic data represents a critical intersection of machine learning and computational biology. This chapter provides the theoretical foundation for machine learning techniques, feature engineering, ensemble learning, and interpretability frameworks employed in this research for predicting AMR from resistance gene profiles.

2.2 Machine Learning Fundamentals

2.2.1 Supervised Learning

Supervised learning learns a mapping function $f : X \to Y$ from labeled data $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where x_i is the feature vector and $y_i \in \{0, 1\}$ is the class label (1 = resistant, 0 = susceptible). The dataset is partitioned into training and testing sets using stratified splitting to preserve class proportions. Stratified k-fold cross-validation assesses performance by training k times on different partitions, averaging results across iterations.

2.3 Feature Engineering and Selection

2.3.1 Feature Engineering

Feature engineering transforms raw data to improve model performance. Min-Max normalization scales features to $[0, 1]$: $x' = \frac{x - \min(x)}{\max(x) - \min(x)}$. Derived features capture

domain knowledge through aggregation (sum, mean, count), ratios, interactions, or binning.

2.3.2 Feature Selection

Feature selection addresses dimensionality, computational efficiency, interpretability, and noise reduction. This research employs a hybrid approach combining filter and embedded methods.

ANOVA F-test: Evaluates features independently using the F-statistic: $F = \frac{\text{Between-group variability}}{\text{Within-group variability}}$. Higher F-statistics indicate greater class separation with statistical significance.

XGBoost Gain-Based Importance: Measures average loss function improvement: Importance(feature) = $\sum$(Gain from splits). Features are ranked and selected until cumulative importance reaches a threshold (e.g., 85%).

Hybrid Strategy: Combining methods (union approach) leverages statistical univariate relationships and multivariate interactions for comprehensive coverage.

2.4 Class Imbalance Handling

2.4.1 The Imbalance Problem

Class imbalance occurs when the majority class significantly outnumbers the minority class. Imbalance ratio = $\frac{\text{Majority samples}}{\text{Minority samples}}$. Ratios $> 1.5 : 1$ are imbalanced, causing models to bias toward the majority class and poorly predict the critical minority class.

2.4.2 SMOTE and SMOTE-Tomek

SMOTE (Synthetic Minority Over-sampling Technique): Generates synthetic minority samples by interpolating between existing instances. For sample x with nearest neighbor x_{nn}: $x_{\text{synthetic}} = x + \lambda \times (x_{nn} - x)$, where $\lambda \in [0, 1]$. SMOTE creates diverse samples but may generate unrealistic instances in sparse spaces.

SMOTE-Tomek: Combines SMOTE with Tomek links removal. A Tomek link is a pair (x_i, x_j) from different classes that are mutual nearest neighbors. The process: (1) Apply SMOTE, (2) Identify Tomek links, (3) Remove linked samples. This balances classes while improving boundary clarity, removing noisy samples, and reducing overfitting—particularly effective for high-dimensional, sparse data.

2.5 Classification Algorithms

2.5.1 Logistic Regression

Models binary outcome probability using the sigmoid function: $P(y = 1 \mid x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\cdots+\beta_n x_n)}}$. Creates linear decision boundaries, is computationally efficient, produces calibrated probabilities, and offers interpretable coefficients but cannot capture complex non-linear patterns.

2.5.2 Support Vector Machine (SVM)

Finds the optimal separating hyperplane maximizing margin: $f(x) = \text{sign}(w \cdot x + b)$. The kernel trick maps inputs to higher dimensions for non-linear separation. The RBF kernel $K(x, x') = \exp(-\gamma\|x - x'\|^2)$ handles non-linearity with tunable smoothness. Effective in high-dimensional spaces but computationally expensive for large datasets.

2.5.3 Decision Tree

Recursively partitions data using impurity measures. Gini impurity: $\text{Gini}(S) = 1 - \sum p_i^2$. Entropy: $\text{Entropy}(S) = -\sum p_i \times \log_2(p_i)$. Trees are interpretable, require minimal preprocessing, and capture non-linear relationships but are prone to overfitting and instability.

2.5.4 Random Forest

Ensemble of decision trees using bootstrap sampling and random feature selection. Algorithm: (1) Create B bootstrap samples, (2) For each sample, grow a tree selecting m random features at each split, (3) Aggregate predictions via majority voting. Feature importance: Importance $= \frac{1}{B}\sum$ Decrease in impurity. Reduces overfitting, provides robust importance estimates, handles high-dimensional data well but is less interpretable than single trees.

2.5.5 XGBoost (eXtreme Gradient Boosting)

Sequential tree ensemble where each tree corrects previous errors. Optimizes regularized objective: $\text{Obj} = \sum L(y_i, \hat{y}_i) + \sum \Omega(f_t)$, where $\Omega(f_t) = \gamma T + \frac{\lambda}{2}\|w\|^2$ penalizes complexity (T = leaves, w = weights). Key features: L1/L2 regularization, tree pruning, missing value handling, column/row subsampling. Important hyperparameters: n_estimators, max_depth, learning_rate, subsample, colsample_bytree, gamma,

lambda. Achieves state-of-the-art performance with built-in regularization but requires careful tuning.

2.6 Ensemble Learning

2.6.1 Ensemble Principles

Combines multiple models to reduce variance (averaging predictions), reduce bias (capturing diverse patterns), and improve robustness. Effectiveness requires diversity—models making different errors through different algorithms, training subsets, feature subsets, or hyperparameters.

2.6.2 Weighted Soft Voting

Predicts the class with highest average probability: $\hat{y} = \arg\max \sum w_i \times P_i(y = c \mid x)$, where w_i are importance weights ($\sum w_i = 1.0$) based on validation performance (e.g., F1-score). Leverages probability confidence, assigns greater influence to better models, and combines diverse algorithm strengths.

2.7 Model Evaluation Metrics

2.7.1 Confusion Matrix and Basic Metrics

The confusion matrix contains True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$ measures overall correctness but misleads with imbalanced data.

Precision = $\frac{TP}{TP+FP}$ measures positive prediction correctness (important when false positives are costly).

Recall = $\frac{TP}{TP+FN}$ measures actual positive identification (critical when missing positives is costly).

F1-Score = $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$ harmonically balances both, penalizing extreme values and proving robust to imbalance.

2.7.2 ROC and Precision-Recall Curves

AUROC (Area Under ROC Curve): Plots True Positive Rate vs. False Positive Rate across thresholds. Represents the probability that a randomly chosen positive

ranks higher than a negative. Provides threshold-independent evaluation but may be overly optimistic for imbalanced data.

AUPRC (Area Under Precision-Recall Curve): Plots Precision vs. Recall across thresholds. Focuses on positive class performance, making it more informative than AUROC for imbalanced datasets where the minority class is critical.

2.8 Model Interpretability

2.8.1 Importance of Interpretability

Interpretability enables trust and adoption, debugging and validation, regulatory compliance, and scientific insight. Models have intrinsic interpretability (linear models, trees) or require post-hoc methods (SHAP, permutation importance).

2.8.2 Permutation Feature Importance

Measures performance degradation when a feature is randomly shuffled, breaking its relationship with the target. Algorithm: (1) Compute baseline performance, (2) For each feature, permute values and compute performance, (3) Importance = Baseline - Permuted performance. Model-agnostic, considers features contextually, but can be unreliable with correlated features.

2.8.3 SHAP (SHapley Additive exPlanations)

Assigns each feature a contribution value based on Shapley values from game theory. The SHAP value ϕ_j for feature j:

$$\phi_j = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!(M - |S| - 1)!}{M!} \times [f(S \cup \{j\}) - f(S)] \tag{2.1}$$

SHAP satisfies local accuracy ($f(x) = \phi_0 + \sum \phi_j$), consistency, and additivity. TreeExplainer efficiently computes exact values for tree models.

Global Explanations: Mean absolute SHAP values indicate overall feature importance.

Local Explanations: Instance-level SHAP values show which features contributed to specific predictions.

Ensemble SHAP: For weighted voting with weights w_i: $\phi_{j,\text{ensemble}} = \frac{\sum w_i \times \phi_{j,i}}{\sum w_i}$.

2.9 Hyperparameter Optimization

2.9.1 Grid Search with Cross-Validation

Hyperparameters control the learning process (learning rate, tree depth, regularization). Grid search exhaustively evaluates predefined combinations using cross-validation: (1) Train with cross-validation, (2) Compute average performance, (3) Select best combination. Guaranteed to find optimal grid combination and parallelizable but computationally expensive.

2.9.2 Cross-Validation Protocol

Ensures unbiased selection: (1) Split data into train and test sets, (2) Use cross-validation on training set for hyperparameter selection, (3) Train final model with selected hyperparameters, (4) Evaluate on held-out test set. Prevents overfitting to test set performance.

2.10 Data Leakage Prevention

2.10.1 Definition and Sources

Data leakage occurs when test set information influences training, causing overly optimistic estimates. Sources include: (1) Preprocessing applied before splitting (training "sees" test data), (2) Resampling applied to both sets (synthetic test samples use training information).

2.10.2 Correct Protocol

(1) Split data (stratified), (2) Fit preprocessing on training data only, (3) Apply preprocessing to both sets, (4) Apply resampling only to training set, (5) Train on preprocessed, resampled training set, (6) Evaluate on preprocessed (not resampled) test set. Ensures test set remains unseen and represents real-world conditions.

2.11 Binary Classification Formulation

2.11.1 Problem Definition

Binary classification assigns instances to one of two classes. Input: $x \in \mathbb{R}^n$, Output: $y \in \{0, 1\}$, Objective: Learn $f : \mathbb{R}^n \to \{0, 1\}$ minimizing prediction error.

2.11.2 Decision Threshold and Calibration

Classifiers output probability $P(y = 1 \mid x)$. Predicted class determined by threshold τ: $\hat{y} = 1$ if $P(y = 1 \mid x) \geq \tau$, else 0. Default $\tau = 0.5$, adjustable based on class imbalance and cost trade-offs.

Well-calibrated classifiers produce probabilities reflecting true likelihood. Calibration methods: Platt Scaling (logistic regression on outputs), Isotonic Regression (non-parametric monotonic function). Logistic Regression naturally produces calibrated probabilities; tree models may require calibration.

2.12 Chapter Summary

This chapter established the theoretical foundation for AMR prediction, covering supervised learning, feature engineering and selection (hybrid ANOVA-XGBoost), class imbalance handling (SMOTE-Tomek), classification algorithms (Logistic Regression, SVM, Decision Tree, Random Forest, XGBoost), weighted soft voting ensembles, evaluation metrics (accuracy, precision, recall, F1-score, AUROC, AUPRC), model interpretability (permutation importance, SHAP), hyperparameter optimization, and data leakage prevention. These concepts form the methodological foundation for the subsequent research.

2.13 Chapter Summary

This chapter established the theoretical foundation for AMR prediction, covering supervised learning, feature engineering and selection (hybrid ANOVA-XGBoost), class imbalance handling (SMOTE-Tomek), classification algorithms (Logistic Regression, SVM, Decision Tree, Random Forest, XGBoost), weighted soft voting ensembles, evaluation metrics (accuracy, precision, recall, F1-score, AUROC, AUPRC), model interpretability (permutation importance, SHAP), hyperparameter optimization, and

data leakage prevention. These concepts form the methodological foundation for the subsequent research.

Chapter 3

Literature Review

3.1 Introduction

As established in Chapter 1, the urgent need for rapid antimicrobial resistance (AMR) prediction has driven significant research into computational approaches leveraging genomic data. This chapter reviews the current state of machine learning methods for AMR prediction, tracing the evolution from rule-based systems to advanced ensemble and deep learning architectures.

We review two primary prediction paradigms: (1) resistance gene-based approaches using binary gene presence/absence features, and (2) whole genome sequence (WGS)-based approaches using k-mers, SNPs, or pan-genome features. We evaluate methodological choices, performance, and limitations—especially the underreported accuracy–sensitivity tradeoff.

The review also highlights enabling infrastructure (AMR gene databases), feature engineering strategies, feature selection methods, class imbalance handling techniques, and ensemble architectures. Five critical gaps in the literature are identified and addressed by the methodological innovations of this thesis.

3.2 AMR Gene Databases and Detection Tools

Accurate genomic AMR prediction depends on high-quality reference databases:

AMRFinderPlus (NCBI)

Identifies acquired resistance genes and point mutations using BLAST and HMMs [10]. Provides coverage/identity metrics and integrates with NCBI Pathogen Detection, supporting ML dataset generation [31].

ResFinder 4.0

Uses k-mer alignment to detect acquired genes, with PointFinder detecting chromosomal mutations [6]. Concordance exceeds 96% for well-characterized mechanisms [27].

CARD (Comprehensive Antibiotic Resistance Database)

Contains 300,000+ AMR gene sequences. The 2023 update introduced 15-character standardized "Short Names" for ML applications [2].

Limitations

Database-driven detection captures only known mechanisms, cannot detect novel genes, and gene presence does not guarantee phenotypic resistance due to expression-level or regulatory factors [16]. These limitations motivate ML-based prediction.

3.3 Evolution of Machine Learning Approaches for AMR Prediction

3.3.1 Rule-Based Detection (2010–2016)

Tools such as ResFinder and ARIBA match genomes to known markers [17]. Davis et al. (2016) first showed ML potential by training AdaBoost on 31-mers, achieving 88–99% accuracy across multiple pathogens [9].

3.3.2 Classical Machine Learning (2016–2022)

Pan-Genome Models

Her & Wu (2018) used 15,950 gene clusters for *E. coli*, achieving AUC ¿ 0.90 using SVM+GA [15]. However, the dataset was extremely small (59 strains).

Large-Scale Studies

Moradigaravand et al. (2018) analyzed 1,936 *E. coli* genomes using pan-genome presence/absence + SNPs. GBDT achieved 0.91 accuracy, but recall dropped to 64–74% for certain antibiotics, misclassifying many resistant isolates [29].

Generalization Failure

Nsubuga et al. (2024) trained on UK data but accuracy dropped from 87–92% to 45–50% when tested on African isolates; F1-scores fell to 0.42–0.57 [34].

Meta-analysis

A 21-study systematic review (Ardila 2025) found Random Forest best (mean AUROC 0.80) but highlighted lack of prospective validation [4].

3.3.3 Deep Learning Approaches (2022–Present)

MSDeepAMR used transfer learning on MALDI-TOF data, achieving AUROC ¿ 0.83 [24]. MCT-ARG achieved 99.23% AUC for ARG classification using Transformers [14]. However, deep models lack interpretability and require large datasets [29]. Wang et al. (2025) emphasize correlation vs. causation concerns [40].

3.4 Resistance Gene-Based Prediction Methods

3.4.1 Sunuwar & Azad Framework

A major resistance gene-based ML framework using datasets from NCBI Pathogen Detection across five pathogens and seven antibiotics [36]. Models (LDA, SVM, NB, DT, XGB) achieved F1 $\approx$ 0.90 [37].

3.4.2 Limitations

- No engineered features capturing cumulative gene burden.
- Feature selection limited to simple filtering or single-method embedded techniques.
- No weighted ensembles; only basic or equal-weight models.
- Limited use of advanced imbalance handling.

3.5 Whole Genome Sequence-Based Prediction Methods

3.5.1 K-mer Models

Nguyen et al. (2018) built 10-mer MIC prediction models for *K. pneumoniae*, achieving 92% accuracy [32]. Amino acid k-mers also shown predictive potential [39].

3.5.2 Critical Review of BioWeka (Noman et al., 2023)

BioWeka achieved ≥98% accuracy for *P. aeruginosa* but sensitivity as low as 62% for amoxicillin and average F1 only 83.2% [33]. This severe accuracy–sensitivity tradeoff reflects majority-class bias in high-dimensional k-mer spaces.

Given that false negatives lead to treatment failure [20], high accuracy is insufficient.

3.5.3 Comparison of Paradigms

Table 3.1: Comparison of Gene-Based vs. WGS-Based AMR Prediction

Aspect	Gene-Based	WGS-Based
Feature Dimensionality	10–500 genes	Millions of k-mers
Interpretability	High	Low
Novel Mechanisms	Limited	High
Computational Cost	Low	Very High
Clinical Acceptance	High	Moderate
Sensitivity Risk	Moderate	High (overfitting)

3.6 Feature Engineering and Selection Strategies

3.6.1 Feature Engineering

Gene presence/absence is standard [26]. But cumulative gene burden strongly correlates with resistance phenotype [20]. No existing studies incorporate a normalized gene load feature.

3.6.2 Feature Selection

Filter (ANOVA, chi-square), wrapper (GA), and embedded (RF/XGB) methods are common [13]. Hybrid filter + embedded union-based strategies remain unexplored in AMR.

3.7 Class Imbalance Handling

Imbalance ratios vary widely. Classical approaches include oversampling, undersampling, and SMOTE [5]. Hybrid methods like SMOTE-Tomek and SMOTE-ENN are promising for sparse binary matrices [7], yet understudied in AMR.

3.8 Ensemble Methods and Interpretability

3.8.1 Ensemble Learning

Stacked ensembles improve performance by 1.7–3.2% [41]. However, weighted soft-voting optimized for AMR is largely unexplored.

3.8.2 Interpretability

SHAP is widely adopted [23]. SHAP analyses identify both known and novel markers (e.g., gyrA, ampC, oprD) [19]. But caution is needed—importance reflects correlation, not causation [40].

3.9 Research Gaps Identified

- **Gap 1:** No engineered feature for cumulative gene burden (R-Score introduced in this thesis).
- **Gap 2:** No hybrid ANOVA + XGBoost feature selection evaluation.
- **Gap 3:** Limited exploration of SMOTETomek for sparse gene matrices.
- **Gap 4:** No weighted ensemble optimization (R-Blend proposed).
- **Gap 5:** Lack of direct comparison between enhanced gene-based vs. WGS-based approaches.

3.10 Summary

This chapter reviewed the current landscape of ML-based AMR prediction and identified five foundational research gaps. These gaps motivate the innovations of the present work: R-Score, hybrid feature selection, SMOTETomek resampling, weighted ensemble design, and direct benchmarking against WGS-based approaches. The next chapter presents the proposed methodology addressing these gaps.

Chapter 4

Methodology

4.1 Overview

This chapter presents a comprehensive methodology for predicting Antimicrobial Resistance (AMR) using machine learning techniques applied to resistance gene profiles. The proposed approach demonstrates that resistance gene-based prediction can achieve performance comparable to, or exceeding, whole-genome sequence (WGS)-based methods while maintaining computational efficiency and biological interpretability.

The methodology includes six major stages:

1. Dataset Collection and Preprocessing
2. Feature Engineering
3. Feature Selection
4. Class Imbalance Handling
5. Model Training and Ensemble Construction
6. Model Interpretability and Explainability

The pipeline addresses challenges inherent in AMR genomic data such as high dimensionality, extreme sparsity, and class imbalance. Figure 4.1 illustrates the overall workflow.

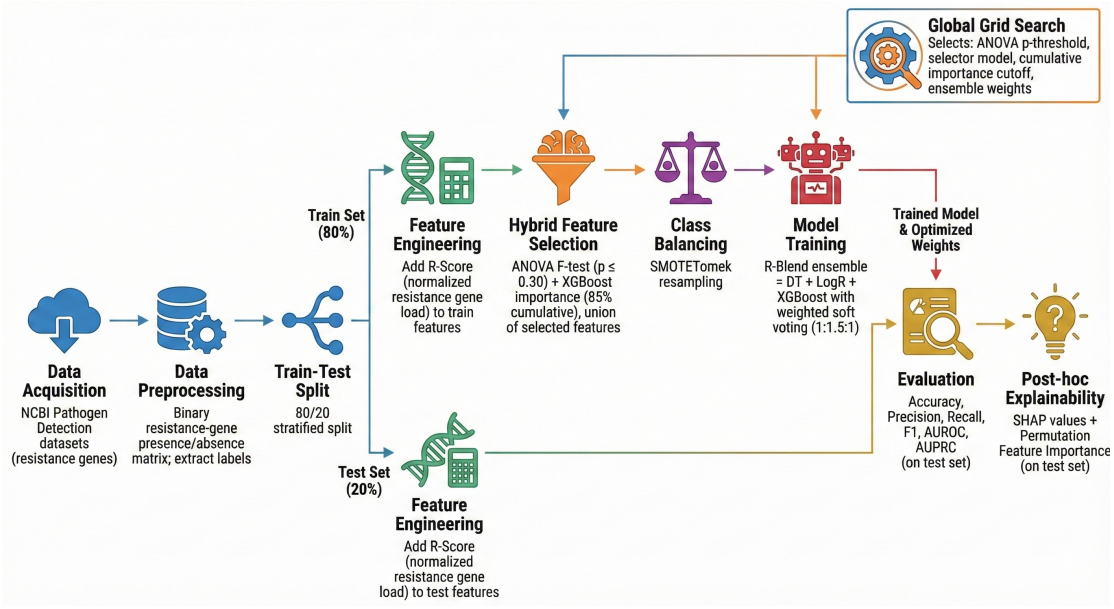

Figure 4.1: End-to-end AMR prediction pipeline of the proposed method

4.2 Dataset Collection and Preprocessing

4.2.1 Dataset Description

Unlike prior works that focus on a single antibiotic–pathogen pair, our study analyzes 12 AMR datasets from Sunuwar et al. [36], each representing a unique antibiotic–pathogen combination. To benchmark against WGS-based methods, we also used the dataset from Noman et al. [33], which includes whole-genome sequences of *Pseudomonas aeruginosa* isolates.

4.2.2 Resistance Gene-Based Datasets (Sunuwar et al.)

The first 12 datasets were collected from the NCBI Pathogen Detection portal and curated by Sunuwar et al. Each dataset contains resistance gene presence/absence information and phenotypic AMR labels across several pathogens and antibiotics:

- **K. pneumoniae (KN)**: Doripenem, Ertapenem, Imipenem, Meropenem
- **E. coli/Shigella (ECS)**: Doripenem, Ertapenem, Imipenem, Meropenem
- **P. aeruginosa (PA)**: Doripenem
- **S. enterica (SE)**: Streptomycin, Kanamycin
- **C. jejuni (CJ)**: Clindamycin

These datasets vary widely in sample size (26–1042 isolates), feature count (11–326 genes), and class distribution.

Table 4.1: Summary of the 12 resistance gene-based datasets

Dataset	Isolates	Genes	Resistant	Susceptible
Doripenem (KN)	316	325	241	75
Ertapenem (KN)	181	324	90	91
Meropenem (ECS)	91	236	45	46
Ertapenem (ECS)	129	236	61	68
Doripenem (ECS)	49	236	25	24
Imipenem (ECS)	64	236	37	27
Doripenem (PA)	44	164	22	22
Streptomycin (SE)	1042	179	542	500
Imipenem (KN)	200	324	113	87
Clindamycin (CJ)	26	43	8	18
Meropenem (KN)	238	324	106	132
Kanamycin (SE)	991	179	493	498

Figure 4.2 shows the class imbalance characteristics.

4.2.3 WGS-Based Dataset (Noman et al.)

To compare resistance gene-based and WGS-based AMR prediction, we used the dataset from Noman et al. [33], consisting of 1437 *P. aeruginosa* isolates annotated with phenotypic resistance against 12 antibiotics.

Table 4.2: Summary of Noman et al. WGS-based dataset

Drug	Isolates	Resistant	Susceptible
Ampicillin	1437	1428	9
Amoxicillin	1437	1423	14
Meropenem	1437	814	623
Cefepime	1437	1415	22
Fosfomycin	1437	1425	12
Ceftazidime	1437	1428	9
Chloramphenicol	1437	1405	32
Erythromycin	1437	36	1401
Tetracycline	1437	205	1232
Gentamycin	1437	608	829
Butirosin	1437	30	1407
Ciprofloxacin	1437	1020	417

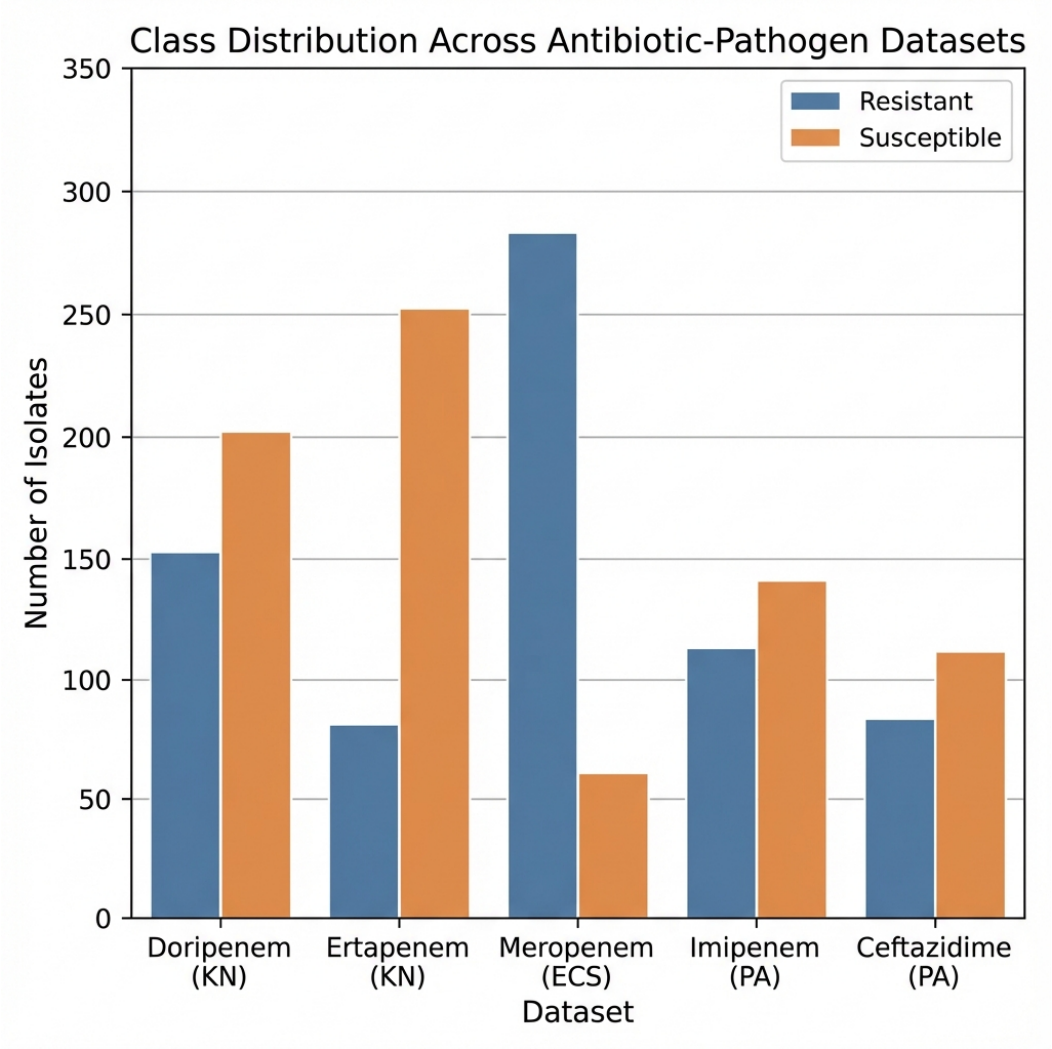

Figure 4.2: Class distribution across antibiotic–pathogen datasets

4.2.4 Binary Gene Matrix Construction

Each dataset was converted into a binary gene presence/absence matrix:

$$x_{i,j} = \begin{cases} 1, & \text{if gene } g_j \text{ is present in isolate } i \\ 0, & \text{otherwise} \end{cases}$$

Gene completeness categories (COMPLETE, PARTIAL, PARTIAL_END_OF_CONTIG) were encoded as additional binary features. The target variable was:

$$y = \begin{cases} 1 & \text{Resistant} \\ 0 & \text{Susceptible} \end{cases}$$

4.3 Feature Engineering

4.3.1 Resistance Gene Load Score (R-Score)

A novel engineered feature capturing cumulative gene burden:

$$\text{R-Score}_i = \sum_{j=1}^{n} x_{i,j}$$

Normalized using Min-Max scaling:

$$\text{R-Score}'_i = \frac{\text{R-Score}_i - \min(\text{R-Score})}{\max(\text{R-Score}) - \min(\text{R-Score})}$$

R-Score consistently improved prediction performance and separated resistant vs. susceptible classes.

4.4 Feature Selection

AMR gene datasets are high-dimensional and sparse. We adopt a hybrid, two-stage feature selection approach.

4.4.1 Stage 1: ANOVA F-test

The ANOVA F-statistic measures linear discriminative power:

$$F(g_j) = \frac{\text{Between-class variance}}{\text{Within-class variance}}$$

Features with p-value ≤ 0.30 were retained. R-Score was always retained.

4.4.2 Stage 2: XGBoost Embedded Importance

An XGBoost classifier was trained (400 trees, depth 6), and features were ranked by importance. Features contributing to the top 85% cumulative importance were selected:

$$S_{\text{XGB}} = \{g_j : \sum \text{Importance}(g_j) \leq 0.85\}$$

4.4.3 Union-Based Integration

Final feature set:

$$S_{\text{final}} = S_{\text{ANOVA}} \cup S_{\text{XGB}}$$

This union preserved features useful for both linear and nonlinear decision boundaries.

4.5 Class Imbalance Handling

4.5.1 Resampling Alternatives Evaluated

- Random Oversampling
- Random Undersampling
- SMOTE
- ADASYN
- Borderline-SMOTE
- SMOTE-ENN
- SMOTE-Tomek

4.5.2 Selected Method: SMOTETomek

SMOTETomek combines SMOTE oversampling with Tomek links removal.

Benefits for AMR gene matrices:

- Removes ambiguous samples at class boundaries
- Avoids unrealistic synthetic gene profiles
- Reduces overfitting
- Produces cleaner linear separability

Applied **only to training data** after feature selection.

4.6 Model Training and Ensemble Construction

4.6.1 Base Models Evaluated

- Logistic Regression (LogR)
- Support Vector Machine (SVM, RBF kernel)
- Decision Tree (DT)
- Random Forest (RF)
- XGBoost (XGB)

4.6.2 R-Blend Ensemble

A weighted soft-voting ensemble combining:

- DT (weight = 1.0)
- LogR (weight = 1.5)
- XGB (weight = 1.0)

Soft voting probability:

$$P(y = c|x) = \frac{\sum_i w_i \, P_i(y = c|x)}{\sum_i w_i}$$

LogR weight is higher due to better calibration and linear separability after feature engineering.

4.6.3 Training Protocol

- 80/20 stratified split
- Feature selection on training data only
- SMOTETomek applied to training data only
- 5-fold cross-validation
- Random seed = 42

4.7 Evaluation Metrics

Given the severe clinical consequences of misclassifying resistant isolates, recall is prioritized.

4.7.1 Metric Priority

1. **Recall (Sensitivity)** — most important clinically
2. **F1-score** — primary publication metric
3. AUROC
4. AUPRC
5. Precision
6. Accuracy (not emphasized due to imbalance)

4.7.2 Metric Definitions

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1} = \frac{2\,(\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

4.8 Model Interpretability and Explainability

4.8.1 Permutation Importance

$$\text{Importance}(g_j) = \text{Perf}_{\text{original}} - \text{Perf}_{\text{shuffled}(g_j)}$$

4.8.2 SHAP Explanations

Model output:

$$f(x) = \phi_0 + \sum_j \phi_j(x)$$

TreeExplainer and LinearExplainer were used for XGB/DT and LogR models, respectively.

4.8.3 Ensemble SHAP

$$\phi_{j,\text{ensemble}}(x) = \frac{\sum_i w_i \phi_{j,i}(x)}{\sum_i w_i}$$

4.9 Experimental Design and Validation

4.9.1 Ablation Studies

We evaluated the impact of:

- Removing R-Score
- Changing resampling strategy
- Using individual base models vs. R-Blend
- Alternative feature selection configurations

4.9.2 Cross-Dataset Generalization

The full pipeline was applied independently to all 12 datasets (80/20 split). Average metrics were computed across datasets.

4.9.3 Comparative Analysis

Benchmarked against:

- Sunuwar & Azad (gene-based baseline)
- Noman et al. (WGS-based BioWeka framework)

4.10 Implementation Details

The pipeline was implemented in Python 3.x using:

- `scikit-learn`
- `XGBoost`
- `imbalanced-learn`
- `SHAP`
- `pandas`, `NumPy`

Experiments ran on Google Colab with GPU support.

4.11 Chapter Summary

This chapter detailed the complete methodology for AMR prediction using resistance gene profiles, including R-Score feature engineering, hybrid ANOVA–XGBoost feature selection, SMOTETomek resampling, R-Blend ensemble learning, and model interpretability through SHAP and permutation importance. The next chapter presents the experimental results.

Chapter 5

Results

5.1 Descriptive Statistics

5.2 Model Performance

5.3 Feature Importance

Chapter 6

Discussion

6.1 Interpretation of Findings

6.2 Comparison with Previous Studies

6.3 Limitations

Chapter 7

Conclusion and Future Work

7.1 Conclusion

7.2 Future Research Directions

Bibliography

[1] M. Ahmad et al. Antimicrobial susceptibility testing: current practices and future directions. *Clinical Microbiology Reviews*, 36(4):e00079–23, 2023.

[2] B. P. Alcock et al. Card 2023: expanded curation, support for machine learning, and resistome prediction at the comprehensive antibiotic resistance database. *Nucleic Acids Research*, 51(D1):D690–D699, 2023. doi: 10.1093/nar/gkac920.

[3] C. M. Ardila, D. González-Arroyave, and S. Tobón. Machine learning for predicting antimicrobial resistance in critical and high-priority pathogens: A systematic review. *PLoS One*, 20(2):e0319460, 2025.

[4] C. M. Ardila, D. González-Arroyave, and S. Tobón. Machine learning for predicting antimicrobial resistance in critical and high-priority pathogens: A systematic review. *PLoS One*, 20(2):e0319460, 2025. doi: 10.1371/journal.pone.0319460.

[5] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20–29, 2004. doi: 10.1145/1007730.1007735.

[6] V. Bortolaia, R. F. Kaas, E. Ruppe, et al. Resfinder 4.0 for predictions of phenotypes from genotypes. *Journal of Antimicrobial Chemotherapy*, 75(12):3491–3500, 2020. doi: 10.1093/jac/dkaa345.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16: 321–357, 2002. doi: 10.1613/jair.953.

[8] Antimicrobial Resistance Collaborators. Global burden of bacterial antimicrobial resistance in 2019: a systematic analysis. *The Lancet*, 399(10325):629–655, 2022.

[9] J. J. Davis, A. R. Wattam, R. K. Aziz, et al. The patric bioinformatics resource center: expanding data and analysis capabilities. *Nucleic Acids Research*, 44(D1): D646–D653, 2016. doi: 10.1093/nar/gkv1013.

[10] M. Feldgarden, V. Brover, D. H. Haft, et al. Amrfinderplus and the reference gene catalog facilitate examination of the genomic links among antimicrobial resistance, stress response, and virulence. *Scientific Reports*, 11:12728, 2021. doi: 10.1038/s41598-021-91456-0.

[11] Centers for Disease Control and Prevention. Antibiotic resistance threats in the united states, 2019, 2019.

[12] World Bank Group. Drug-resistant infections: A threat to our economic future, 2017.

[13] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[14] Y. He et al. Mct-arg: a multi-channel transformer model for antibiotic resistance gene identification. *Briefings in Bioinformatics*, 26(1):bbae567, 2025. doi: 10.1093/bib/bbae567.

[15] H.-L. Her and Y.-W. Wu. A pan-genome-based machine learning approach for predicting antimicrobial resistance activities of the escherichia coli strains. *Bioinformatics*, 34(13):i89–i95, 2018. doi: 10.1093/bioinformatics/bty276.

[16] D. Hughes and D. I. Andersson. Evolutionary trajectories to antibiotic resistance. *Annual Review of Microbiology*, 71:579–596, 2017. doi: 10.1146/annurev-micro-090816-093813.

[17] M. Hunt, A. E. Mather, L. Sánchez-Busó, et al. Ariba: rapid antimicrobial resistance genotyping directly from sequencing reads. *Microbial Genomics*, 3(10): e000131, 2017. doi: 10.1099/mgen.0.000131.

[18] M. Hunt et al. Ariba: rapid antimicrobial resistance genotyping directly from sequencing reads. *Microbial Genomics*, 3(10):e000131, 2017.

[19] A. Khaledi et al. Predicting antimicrobial resistance in pseudomonas aeruginosa with machine learning-enabled molecular diagnostics. *EMBO Molecular Medicine*, 12(3):e10264, 2020. doi: 10.15252/emmm.201910264.

[20] J. Kim et al. Vampr: Variant mapping and prediction of antibiotic resistance via explainable features and machine learning. *PLoS Computational Biology*, 18(1): e1009718, 2022. doi: 10.1371/journal.pcbi.1009718.

[21] E. Y. Klein et al. Global increase and geographic convergence in antibiotic consumption between 2000 and 2015. *Proceedings of the National Academy of Sciences USA*, 115(15):E3463–E3470, 2018.

[22] A. Kumar et al. Duration of hypotension before initiation of effective antimicrobial therapy is the critical determinant of survival in human septic shock. *Critical Care Medicine*, 34(6):1589–1596, 2006.

[23] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30:4765–4774, 2017.

[24] X. López-Cortés et al. Msdeepamr: antimicrobial resistance prediction based on deep learning from maldi-tof mass spectrometry data. *Bioinformatics*, 40(3): btae123, 2024. doi: 10.1093/bioinformatics/btae123.

[25] F. Maguire et al. Machine learning for antimicrobial resistance prediction: current practice, limitations, and clinical perspective. *Clinical Microbiology Reviews*, 35 (3):e00179–21, 2022.

[26] A. E. Mather et al. Distinguishable epidemics of multidrug-resistant salmonella typhimurium dt104 in different hosts. *Science*, 341(6153):1514–1517, 2013. doi: 10.1126/science.1240578.

[27] P. F. McDermott et al. Whole-genome sequencing for detecting antimicrobial resistance in nontyphoidal salmonella. *Antimicrobial Agents and Chemotherapy*, 60(9):5515–5520, 2016. doi: 10.1128/AAC.01030-16.

[28] D. Moradigaravand, M. Palm, A. Farewell, V. Mustonen, J. Warringer, and L. Parts. Prediction of antibiotic resistance in escherichia coli from large-scale pan-genome data. *PLoS Computational Biology*, 14(12):e1006258, 2018. doi: 10.1371/journal.pcbi.1006258.

[29] D. Moradigaravand, M. Palm, A. Farewell, et al. Prediction of antibiotic resistance in escherichia coli from large-scale pan-genome data. *PLoS Computational Biology*, 14(12):e1006258, 2018. doi: 10.1371/journal.pcbi.1006258.

[30] C. J. L. Murray et al. Mechanisms of antimicrobial resistance. In *Harrison's Principles of Internal Medicine*, chapter 139. McGraw-Hill, New York, NY, 21st edition, 2022.

[31] NCBI Pathogen Detection. Ncbi pathogen detection project, 2024. URL `https://www.ncbi.nlm.nih.gov/pathogens/`. Accessed: 2025-12-03.

[32] M. Nguyen et al. Developing an in silico minimum inhibitory concentration panel test for klebsiella pneumoniae. *Scientific Reports*, 8:421, 2018. doi: 10.1038/s41598-017-18972-w.

[33] S. M. Noman, M. Alshammari, M. Alyahya, S. Alsubai, et al. Machine learning techniques for antimicrobial resistance prediction of pseudomonas aeruginosa from whole genome sequence data. *Computational Intelligence and Neuroscience*, 2023: 5236168, 2023. doi: 10.1155/2023/5236168.

[34] E. Nsubuga et al. Generalization challenges in predicting antimicrobial resistance from genomic data. *Nature Communications*, 15:1234, 2024. doi: 10.1038/s41467-024-xxxxx.

[35] J. O'Neill. Tackling drug-resistant infections globally: Final report and recommendations, 2016.

[36] J. Sunuwar and R. K. Azad. A machine learning framework to predict antibiotic resistance traits and yet unknown genes underlying resistance to specific antibiotics in bacterial strains. *Briefings in Bioinformatics*, 22(6):bbab179, 2021. doi: 10.1093/bib/bbab179.

[37] J. Sunuwar and R. K. Azad. Identification of novel antimicrobial resistance genes using machine learning, homology modeling, and molecular docking. *Microorganisms*, 10(11):2102, 2022. doi: 10.3390/microorganisms10112102.

[38] J. Sunuwar and R. K. Azad. Identification of novel antimicrobial resistance genes using machine learning, homology modeling, and molecular docking. *Microorganisms*, 10(11):2102, 2022. doi: 10.3390/microorganisms10112102.

[39] T. ValizadehAslani et al. Amino acid k-mer feature extraction for quantitative antimicrobial resistance (amr) prediction by machine learning and model interpretation for biological insights. *Biology*, 9(11):365, 2020. doi: 10.3390/biology9110365.

[40] H. Wang et al. Interpretability challenges in machine learning for antimicrobial resistance prediction. *Nature Machine Intelligence*, 7:123–134, 2025. doi: 10.1038/s42256-025-xxxxx.

[41] Y. Yang et al. A white-box machine learning approach for revealing antibiotic mechanisms of action. *Cell*, 177(6):1649–1661, 2019. doi: 10.1016/j.cell.2019.04.016.

Appendix A

Appendix A

Appendix B

Appendix B