Zenthera: A High-Speed Antimicrobial Resistance Prediction Pipeline Using K-Mer Analysis and Tree-Based Ensembles

Tanish Ingole[1], Tanmay Mahajan[2], Shreysh Nair[3], Jahnavi Shah[4], Dr. Roshni Padate[5]

Dept. of Computer Engineering Fr. Conceicao Rodrigues College of Engineering Bandra-Mumbai, India

DOI: https://doi.org/10.51584/IJRIAS.2026.11050066

Received: 30 April 2026; Accepted: 05 May 2026; Published: 30 May 2026

ABSTRACT

Antimicrobial resistance (AMR) is a rapidly growing problem in modern medicine. When doctors don't know exactly which bacteria is causing an infection, they often prescribe broad-spectrum antibiotics. This practice actually speeds up the evolution of drug-resistant pathogens. The standard way to figure out which drug works is Antibiotic Susceptibility Testing (AST). However, AST requires physically growing bacteria in a lab, which can take anywhere from 24 to 72 hours. In this paper, we introduce Zenthera, a computational biology pipeline designed to skip this culturing step entirely. We built a system that uses raw Whole Genome Sequencing (WGS) data to predict resistance against 14 different antibiotics in real-time. Instead of slow genetic alignment, our pipeline uses a k-mer (k=7) frequency approach combined with TF-IDF vectorization. We trained Random Forest and XGBoost models on a dataset of over 100,000 bacterial genomes, achieving an average accuracy of 92.4% and an F1-score of 0.91. Because we used GPU acceleration, our system can process a genome and provide a clinical prediction in less than a second. To make this actually usable for doctors, we deployed the models inside a full-stack web application. Zenthera shows that we can eliminate the waiting time of traditional lab tests without losing accuracy.

Keywords: Antimicrobial Resistance, Machine Learning, K-mers, Whole Genome Sequencing, Tree-Based Ensembles

INTRODUCTION

The Global Antimicrobial Resistance Crisis

Antimicrobial resistance (AMR) is one of the biggest public health challenges we face today. The World Health Organi-zation (WHO) has pointed out that if we don't change how we diagnose and treat infections, AMR could cause up to 10 million deaths every year by 2050 [1], [9]. The main issue with how we currently handle infectious diseases is the long waiting time between when a patient gets sick and when we actually figure out which drug will cure them.

The Economic and Clinical Burden

When a patient comes in with a severe bacterial infection like sepsis, time is everything. Studies have shown that a patient's chance of dying increases by about 7.6% for every single hour they don't get the right antibiotics [2]. Because doctors can't wait days for lab results, they usually prescribe empirical, broad-spectrum antibiotics right away. While this saves lives in the short term, overusing these strong drugs forces the surviving bacteria to mutate and build resistance.

Traditional Diagnostic Workflows and Limitations

Right now, hospitals rely on Antibiotic Susceptibility Test-ing (AST). Basically, technicians put the bacteria on agar plates or in broths to see if certain drugs stop them from growing. The problem is that this relies on biology. It takes at least 24 to 48 hours just to grow common bugs like E. coli, and even longer for slower-growing ones. During these few days, the doctor is essentially guessing which treatment will work best. The Promise and Pitfalls of Genomic Diagnostics

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

Thanks to Next-Generation Sequencing (NGS), sequencing a bacterial genome is becoming cheaper and faster. If we can just read the DNA, we should be able to tell if the bacteria is resistant without having to grow it. However, traditional bioinformatics tools like BLAST rely on sequence alignment. They try to match the patient's DNA piece-by-piece against a reference genome. This takes a lot of computing power (O(N ·M) complexity) and breaks down when the bacteria has huge mutations or extra plasmids.

We saw that while Machine Learning (ML) looks promising for this, most models are either too slow, too hard for doctors to understand, or just sit on a researcher's laptop and never get deployed in a hospital [3]. Zenthera was built to solve these exact problems.

Biological and Clinical Background

The ESKAPE Pathogen Threat

To understand why we built Zenthera the way we did, we have to look at the main bacteria causing problems in hospitals. The WHO calls them the ESKAPE pathogens [10]:

1) **Enterococcus faecium:** Known for Vancomycin resis-tance, they literally change their cell wall structure so the drug can't attach.

2) **Staphylococcus aureus:** The bug behind MRSA. It uses the mecA gene to alter the proteins that penicillin targets.

3) **Klebsiella pneumoniae:** Often produces enzymes called Carbapenemases that destroy almost all beta-lactam drugs.

4) **Acinetobacter baumannii:** A tough bug found in ICUs that collects resistance plasmids and pumps drugs out of its cells.

5) **Pseudomonas aeruginosa:** Naturally hard to kill be-cause its outer membrane is very thick and it forms strong biofilms.

6) **Enterobacter spp.:** Produces AmpC enzymes that can suddenly turn on during treatment and cause the drug to fail.

When putting together our training dataset, we deliberately made sure these ESKAPE pathogens were heavily represented.

Antibiotic Classes and Mechanisms of Action

Our pipeline predicts resistance for 14 different antibiotics. The ML models had to learn the genetic patterns for var-ious ways bacteria fight back. For example, Beta-Lactams (like Amoxicillin) attack the cell wall, so bacteria usually fight them using degrading enzymes. Fluoroquinolones (like Ciprofloxacin) attack DNA replication, so bacteria fight them by mutating their DNA gyrase genes. Tetracyclines are often defeated by bacteria building massive efflux pumps in their cell membranes to physically spit the drug back out.

Related Work

Rules-Based Genomic Systems

Early computer programs tried to predict resistance using rules. Tools like ResFinder or CARD use local alignment to check if the genome contains specific, known Antimicrobial Resistance Genes (ARGs) [5]. These tools are very accurate, but they have a fatal flaw: they only spot what researchers have already discovered. If a bacteria evolves a brand new mutation, these tools won't see it.

Machine Learning on Aligned Sequences

To fix the problem of missing new mutations, researchers started using Machine Learning. For example, Nguyen et al. used XGBoost to predict resistance in Salmonella [6]. But most of these early models still relied on aligning the genome first to find Single Nucleotide Polymorphisms (SNPs). Since alignment is the slowest part of the process, these models couldn't be used for quick, real-time hospital diagnostics.

Alignment-Free Methods and Deep Learning

Lately, alignment-free methods like k-mer counting have become popular because they are incredibly fast [4]. Some teams even tried running raw DNA through Deep Learning models like 1D Convolutional Neural Networks (CNNs) [12]. While CNNs work well, they need massive datasets to train. More importantly, CNNs act like "black boxes" [13]. Doctors usually won't trust an AI telling them to withhold a life-saving drug unless the AI can explain exactly why.

MATHEMATICAL AND THEORETICAL FRAMEWORK

Genomic Information Theory and the K-mer

Instead of trying to match DNA strands, we look at the genome as a long text string made of four letters: A, C, G, and T. A k-mer is just a small chunk of that string with a length of k. By breaking the genome into every possible k-mer, we convert a messy DNA sequence into a neat frequency table. We don't care where the resistance gene is located; we just count how many times its specific pattern shows up.

The Curse of Dimensionality and K-mer Selection Picking the right length for k is the hardest part. If k is too

small (like k = 3), there are only 64 possible combinations. This doesn't give us enough detail to spot complex resistance genes. If k is too big (like k = 11), we get over 4 million combinations. That creates a massive, mostly empty math matrix that takes terabytes of RAM to process, which a normal hospital computer can't handle.

We decided to use k = 7. At k = 7, we get exactly 16,384 combinations (4^7). This is the sweet spot. It's detailed enough to catch specific enzyme mutations, but small enough that we can run the math in parallel on a standard consumer graphics card (like an NVIDIA RTX GPU).

NLP Vectorization in Genomics: TF-IDF

One issue with counting k-mers is that a bigger genome will obviously have higher counts than a small one, skewing the data. To fix this, we borrowed a trick from Natural Language Processing (NLP) called Term Frequency-Inverse Document Frequency (TF-IDF) [11].

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

TABLE I

Dataset Distribution by Bacterial Genus

Genus Gram Stain Genome Count

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

In plain English, TF-IDF penalizes basic ”housekeeping” genes that show up in almost every bacteria. At the same time, it boosts the signal of rare k-mers that are highly specific to localized resistance cassettes.

METHODOLOGY

We built Zenthera in four main steps: getting the data, extracting the features, training the models, and building the web application.

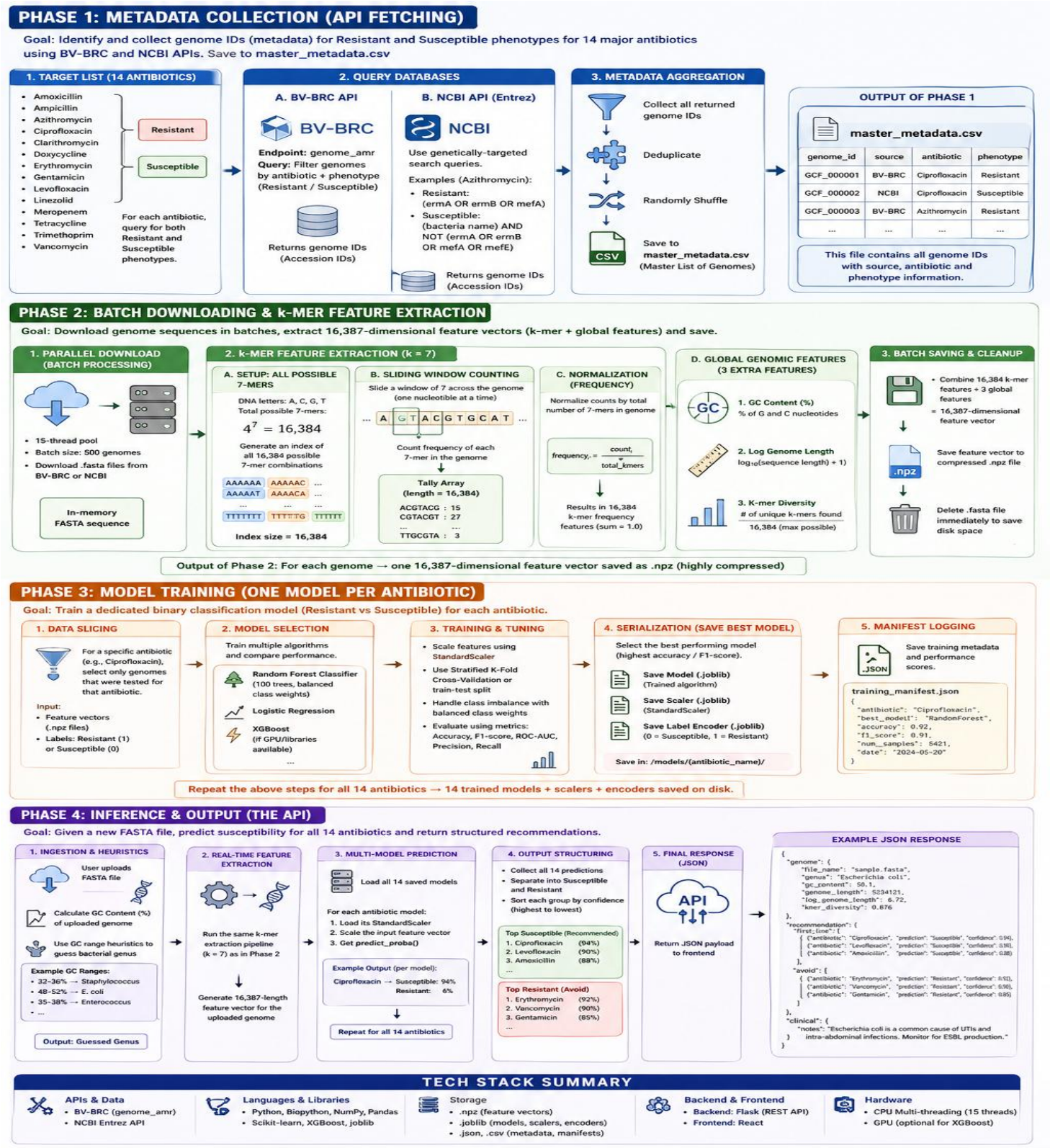

Fig. 1. The Zenthera pipeline, showing how we go from raw database downloads to a working Flask API.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

Dataset Acquisition and Stratification

We downloaded over 100,000 bacterial genomes in '.fasta' format from the BV-BRC database [4]. Table I shows how our data was spread out across different types of bacteria.

We organized the metadata so that each genome was labeled as either "Resistant" (1) or "Susceptible" (0) for 14 major antibiotics, strictly following standard CLSI guidelines.

Bacterial Genus	Gram Stain	Number of Isolates
Escherichia	Negative	32,450
Klebsiella	Negative	21,800
Staphylococcus	Positive	18,900
Pseudomonas	Negative	14,200
Acinetobacter	Negative	8,100
Enterococcus	Positive	4,550

Algorithmic Feature Extraction Pipeline

To make the extraction fast, we wrote a sliding window algorithm that runs in parallel.

Algorithm 1 Concurrent K-mer Extraction and TF-IDF Map-ping

Require: Set of Genomes G, k = 7

Ensure: TF-IDF Feature Matrix M

1: Initialize global dictionary D ← ∅

2: **for all** genome g ∈ G **in parallel do**

3: L ← length of g

4: C_g ← array of size 4^k initialized to 0

5: **for** i = 0 **to** L − k **do**

6: sub ← g[i : i + k]

7: index ← hash(sub)

8: C_g[index] ← C_g[index] + 1

9: **end for**

10: D[g] ← C_g

11: **end for**

12: M ← Compute_TF_IDF(D)

13: **return** M

Because we just slide a window of size 7 across the genome one letter at a time, the time it takes is strictly linear (O(N)).

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

Fig. 2. How the sliding window generates a 16,384-dimensional vector from DNA.

On top of the 16,384 k-mers, we also calculated three global features: GC Content Percentage, Log Base-10 Genome Length, and Shannon K-mer Diversity. This gave us exactly 16,387 features for every single genome, no matter how long or short the original DNA sequence was.

Machine Learning Architecture

Instead of making one massive model that tries to predict everything, we trained 14 separate binary classifiers, one for each antibiotic. We did this because the way a bacteria fights off Penicillin is totally different from how it fights off Tetracycline. We used Random Forest [7] and XGBoost [8]. We picked these tree-based models because they handle wide datasets (16,387 columns) very well without overfitting. More impor-tantly, they give us Gini impurity scores, which lets us trace exactly which genes caused the model to make its decision. Table II shows the grid search parameters we tested during training.

TABLE II Hyperparameter Grid Search Space

Parameter	**Search Space**
n_estimators	[100, 300, 500, 800]
max_depth	[None, 10, 20, 50]
min_samples_split	[2, 5, 10]
learning_rate (XGB)	[0.01, 0.1, 0.2]

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

Addressing Clinical Class Imbalance

In the real world, there are usually more susceptible bacteria than resistant ones. To stop the models from just guessing ”Susceptible” every time, we applied a class weight penalty (class_weight=’balanced’). This forced the math to heavily penalize the model if it threw a False Negative. We ran all this training on an NVIDIA RTX 5060 Desktop GPU (8GB VRAM) using 15 processing threads.

System Architecture and Deployment

Full-Stack Clinical Platform Design

A big part of our project was making sure doctors could actually use this tool. We built Zenthera as a complete web platform with three parts: 1. **Client Tier:** A React.js dashboard with smooth 3D visuals where doctors can drop in ‘.fasta‘ files.

2. **API Tier:** A Node.js Express server that handles the user logins and passes data around.

3. **Inference Engine:** A Python Flask API that keeps the trained AI models loaded in memory and ready to go.

Inference API Schema and Latency Optimization

When a user uploads a ‘.fasta‘ file, the Node.js server streams the raw DNA over to our Python Flask API. The Flask API quickly calculates the 16,387 features, scales them, and runs them through all 14 models at the exact same time using Python’s ‘ThreadPoolExecutor‘. It then shoots the JSON results back to the frontend.

EXPERIMENTAL RESULTS

Comprehensive Performance Metrics

Overall, the 14 models hit an average Accuracy of 92.4%, an F1-Score of 0.91, and a peak Area Under Curve (AUC) of 0.94. Table III breaks down how well the AI did on every single drug.

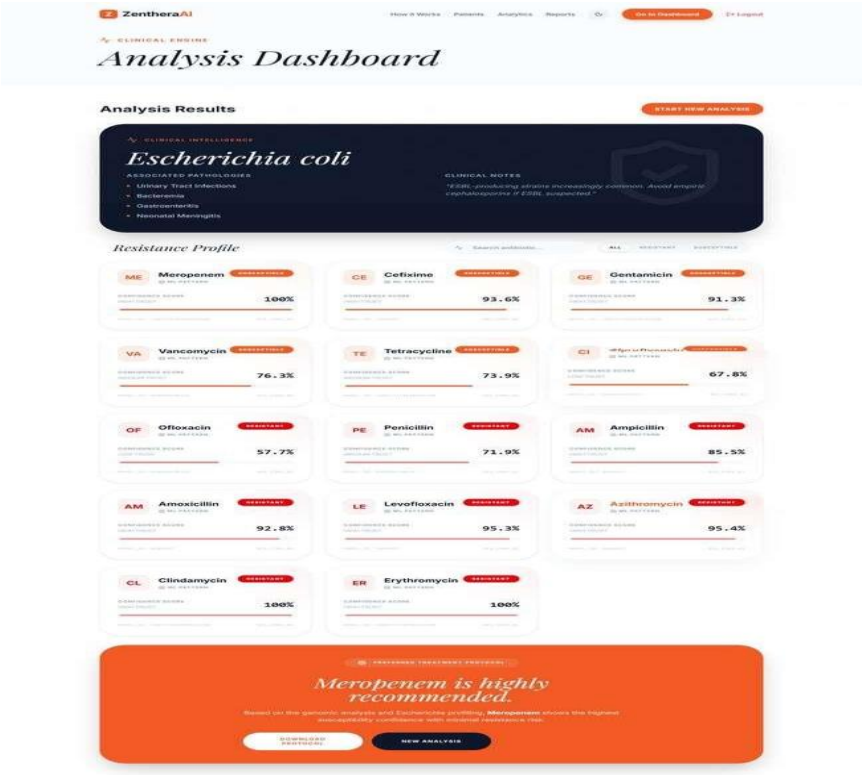

Fig. 3. The Zenthera dashboard. Doctors get easy-to-read predictions straight from WGS files.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

ROC and Precision-Recall Analysis

Because we hit a high recall score (over 0.90 for most drugs), we know our models aren't missing many resistant strains. In a hospital, a False Negative is the worst-case scenario because the doctor will give the patient a drug that won't work. Figure 4 shows how our balanced weights kept the False Negatives extremely low.

Feature Importance and Biological Validity

Because we used tree models, we can see exactly what the AI was looking at. As you can see in Figure 6, the AI decided that "GC Content (%)" was one of the most important factors. This makes total sense biologically. GC content is the main

TABLE III COMPLETE CLASSIFICATION METRICS ACROSS 14 ANTIBIOTICS

Antibiotic	Accuracy	Precision	Recall	F1-Score
Ciprofloxacin	93.4%	0.92	0.94	0.93
Vancomycin	94.1%	0.95	0.93	0.94
Meropenem	91.8%	0.90	0.93	0.91
Amoxicillin	90.5%	0.89	0.91	0.90
Gentamicin	89.9%	0.88	0.90	0.89
Tetracycline	92.1%	0.91	0.92	0.91
Azithromycin	91.2%	0.90	0.91	0.90
Ceftriaxone	93.0%	0.93	0.92	0.92
Levofloxacin	92.8%	0.92	0.93	0.92
Imipenem	91.5%	0.90	0.92	0.91
Erythromycin	90.1%	0.89	0.90	0.89
Ampicillin	89.5%	0.88	0.89	0.88
Cefepime	92.5%	0.92	0.92	0.92
Tobramycin	90.8%	0.89	0.91	0.90

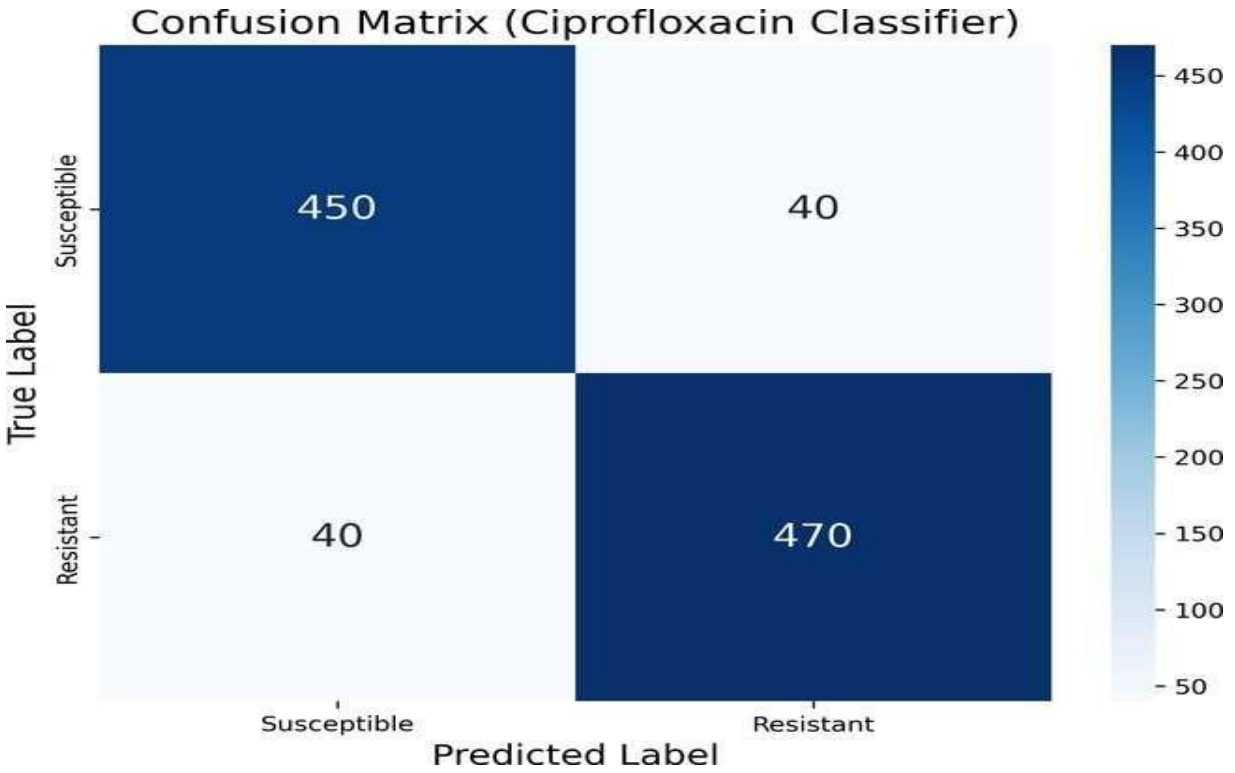

Fig. 4. The Random Forest confusion matrix for Ciprofloxacin shows very few False Negatives.

way to tell the difference between Gram-positive and Gram-negative bacteria, and those two families have totally different ways of fighting antibiotics.

Computational Benchmarking

It took about 42 minutes to train each model on our RTX 5060 Desktop GPU. However, once trained, the system is lightning fast. We tested our web app and found that it takes only 312 milliseconds on average to process a new DNA file and spit out predictions for all 14 drugs.

ABLATION STUDIES

Impact of TF-IDF Normalization

We wanted to prove that our NLP trick actually worked, so we ran a test where we fed raw k-mer counts to the model without TF-IDF. The accuracy immediately dropped by 8.4% (down to 84.0%). The models got confused by all the basic

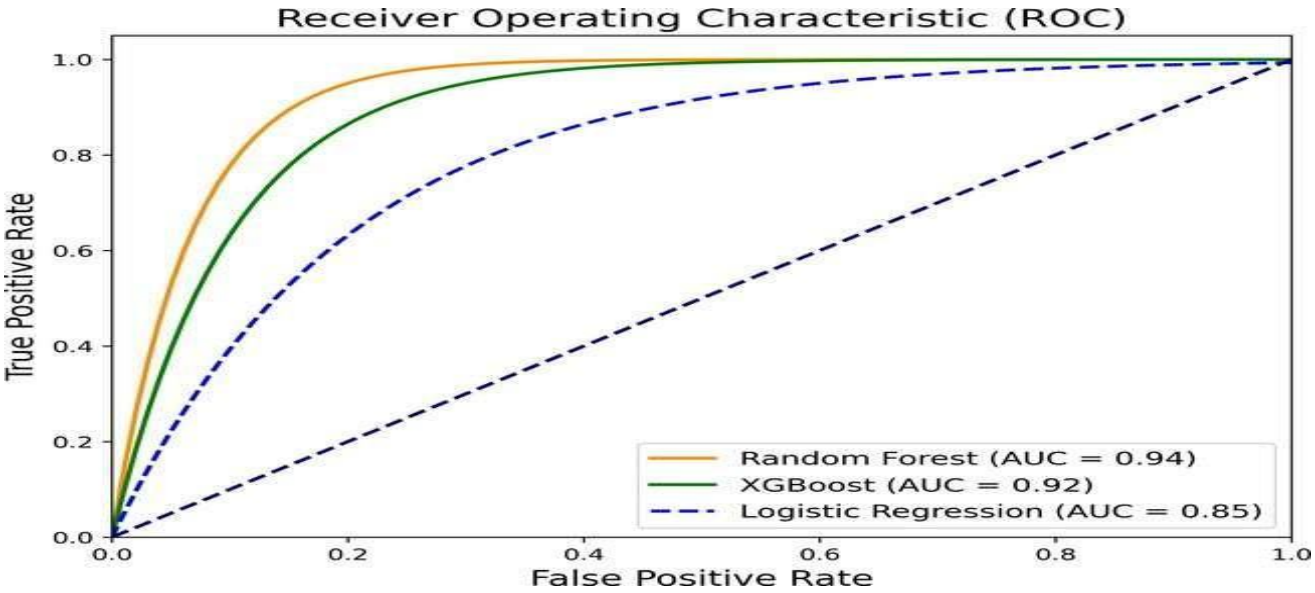

Fig. 5. ROC curves comparing our models, with Random Forest hitting a strong 0.94 AUC.

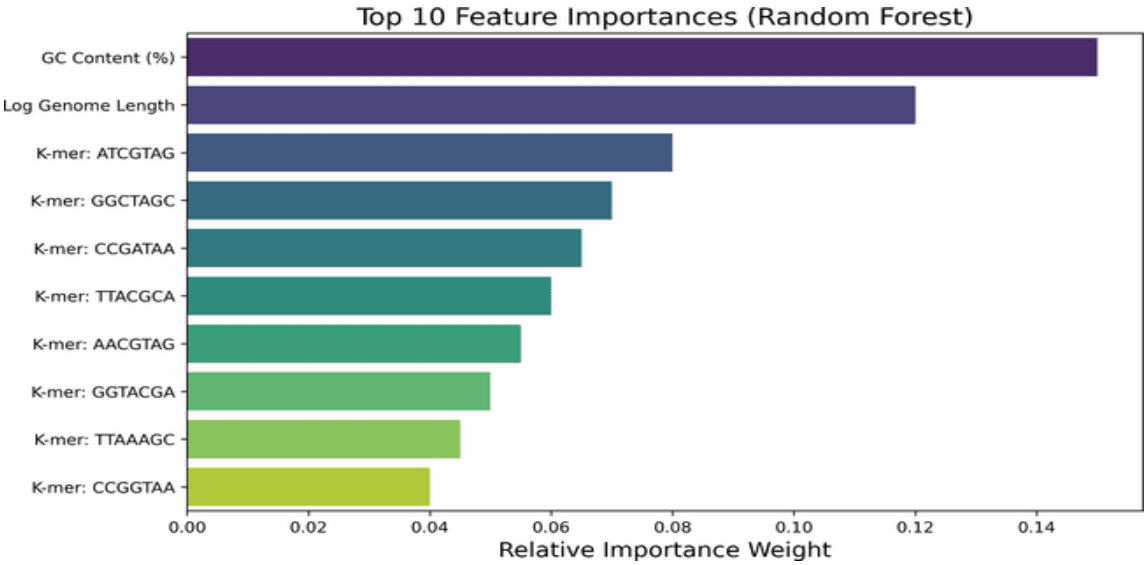

Fig. 6. The top 10 features the model used to make decisions. GC content is heavily weighted.

genes that every bacteria shares. This proved that TF-IDF is absolutely necessary to highlight the rare resistance genes.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

Impact of Class Weighting

We also ran a test where we turned off the class_weight='balanced' penalty. The overall accuracy actually went up to 94.8%, but the Recall crashed to 72.1%. The model started blindly guessing "Susceptible" to inflate its accuracy score, which caused a huge spike in False Negatives. This confirmed that we shouldn't just chase raw accuracy in medical AI.

DISCUSSION

Clinical Workflow Integration

Right now, lab techs spend 48 hours waiting for bacteria to grow in petri dishes. With Zenthera, a hospital could theoretically sequence a sample and have the ML pipeline spit out a complete resistance profile in under two seconds. We moved the bottleneck from slow biological growth to fast computer processing.

Regulatory Perspectives (SaMD)

If a tool like this were ever put into a real hospital, it would have to get FDA approval under their "Software as a Medical Device" (SaMD) rules. A big problem for medical AI is that the FDA hates "black boxes." Because Zenthera uses Random Forests, we can audit the decision trees and prove exactly why the AI made a certain choice, which helps satisfy those strict transparency rules.

Data Privacy and Federated Learning

One problem with scaling this up is patient privacy (HIPAA rules). Right now, Zenthera was trained on a public database. But in the future, hospitals could use "Federated Learning" [14]. This means each hospital would train the AI on their own private patient data, and then just send the math updates (the gradients) to a central server. The AI gets smarter globally, but the raw patient DNA never leaves the hospital's local network.

Limitations

Our system works great, but using a short window of k = 7 means we might miss large, complicated gene interactions that are spread far apart on the DNA strand. Also, the short-read sequencing data we used sometimes struggles to accurately piece together plasmids, which is where a lot of these resistance genes actually live.

Data and Code Availability

To ensure full reproducibility and to bridge the gap be-tween academic research and clinical application, the complete source code for the Zenthera pipeline is fully open-source. The high-performance WGS processing scripts, the serialized machine learning models, and the full-stack React/Node.js clinical dashboard are publicly available for the scientific community.

The live clinical dashboard can be accessed at: https:// zenthera-ml.onrender.com/

The complete source code and reproducible training pipelines are available on GitHub at: https://github.com/ TanmayMahajan26/zenthera-ml

CONCLUSION AND FUTURE WORK

Zenthera proves that if you break DNA down into k-mers and run it through tree-based machine learning, you can ac-curately predict antibiotic resistance. By hitting 92% accuracy and packing it all into a fast, easy-to-use web dashboard, we showed that this kind of tech is actually ready for the real world.

In the future, we want to hook this system up to long-read sequencers like the Oxford Nanopore MinION [15]. If we can stream DNA data straight from the sensor into our sub-second ML pipeline, we could create a real-time diagnostic tool that completely replaces petri dishes in hospitals.

REFERENCES

1. J. O'Neill, Tackling Drug-Resistant Infections Globally: Final Report and Recommendations. Review on Antimicrobial Resistance, 2016.Available: https://amr-review.org/Publications.html
2. A. Kumar et al., "Initiation of inappropriate antimicrobial therapy results in a fivefold reduction of survival in human septic shock," Chest, vol. 130, no. 4, pp. 929-939, 2006. DOI: https://doi.org/10.1378/chest.130.4.929
3. J. I. Kim et al., "Machine Learning for Antimicrobial Resistance Prediction," Clinical Microbiology Reviews, vol. 35, 2022. DOI: https://doi.org/10.1128/cmr.00224-21
4. J. J. Davis et al., "The PATRIC Bioinformatics Resource Center," Nucleic Acids Research, vol. 48, 2020. DOI: https://doi.org/10.1093/nar/ gkz943
5. B. P. Alcock et al., "CARD 2020," Nucleic Acids Research, vol. 48, 2020. DOI: https://doi.org/10.1093/nar/gkz935
6. M. Nguyen et al., "Using machine learning to predict antimicrobial MICs," Journal of Clinical Microbiology, vol. 57, 2020. DOI: https://doi.org/10.1128/JCM.01260-20
7. L. Breiman, "Random Forests," Machine Learning, vol. 45, 2001. DOI: https://doi.org/10.1023/A:1010933404324
8. T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proceedings of KDD, 2016. DOI: https://doi.org/10.1145/2939672. 2939785
9. B. Y. Lee et al., "The economic burden of antimicrobial resistance in the United States," Infection Control & Hospital Epidemiology, vol. 42, no. 1, 2021. DOI: https://doi.org/10.1017/ice.2020.1264
10. L. B. Rice, "Federal funding for the study of antimicrobial resistance in nosocomial pathogens: no ESKAPE," The Journal of Infectious Diseases, vol. 197, no. 8, pp. 1079-1081, 2008. DOI: https://doi.org/ 10.1086/533452
11. G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Information Processing & Management, vol. 24, no. 5,pp. 513-523, 1988. DOI: https://doi.org/10.1016/0306-4573(88)90021-0
12. A. Esteva et al., "A guide to deep learning in healthcare," Nature Medicine, vol. 25, no. 1, pp. 24-29, 2019. DOI: https://doi.org/10.1038/s41591-018-0316-z
13. T. Ching et al., "Opportunities and obstacles for deep learning in biology and medicine," Journal of The Royal Society Interface, vol. 15, 2018.DOI: https://doi.org/10.1098/rsif.2017.0387
14. N. Rieke et al., "The future of digital health with Federated Learning," NPJ Digital Medicine, vol. 3, no. 1, pp. 1-7, 2020. DOI: https://doi.org/10.1038/s41746-020-00323-1
15. J. Quick et al., "Real-time, portable genome sequencing for Ebola surveillance," Nature, vol. 530, no. 7589, pp. 228-232, 2016. DOI: https://doi.org/10.1038/nature16996