IMPROVED BACLABNET: ENHANCED DEEP LEARNING FRAMEWORK FOR BACTERIOCIN CLASSIFICATION USING ADVANCED FEATURE ENGINEERING AND ENSEMBLE LEARNING

Dr. QurutulAin, Aliza Saadi, Vanya Shafiq, Aaleen Zainab
Department of Artificial Intelligence
Data Science Faculty
National University of Computer and Emerging Sciences

December 5, 2025

Abstract

The emergence of antibiotic-resistant bacteria necessitates innovative approaches for discovering novel antimicrobial compounds. Bacteriocins produced by Lactic Acid Bacteria (LAB) represent promising alternatives due to their Generally Recognized As Safe (GRAS) status and Qualified Presumption of Safety (QPS). This study presents an improved deep learning framework, BacLABNet-Improved, for binary classification of bacteriocin sequences (BacLAB vs. Non-BacLAB) with significantly enhanced accuracy and robustness. Building upon previous work that achieved 91.47% accuracy, our approach integrates advanced feature engineering, including TF-IDF weighted k-mer frequencies, sequence-level biochemical descriptors, and principal component analysis, combined with sophisticated neural network architectures and ensemble learning techniques. Through stratified 5-fold cross-validation on a balanced dataset of 49,964 sequences, our model achieves an average accuracy of $92.99\% \pm 0.34\%$, with precision of 91.8%, recall of 94.6%, and F1-score of 93.15%. The improved framework demonstrates superior generalization, reduced overfitting, and computational feasibility on cloud platforms like Google Colab, making it accessible for resource-limited research environments.

Index Terms— Deep Learning, Bacteriocin Classification, Feature Engineering, Ensemble Learning, Antibiotic Resistance, Bioinformatics

1 Introduction

The global health crisis posed by antibiotic-resistant pathogens has intensified research into alternative antimicrobial agents [1]. Among these alternatives, bacteriocins—ribosomally synthesized antimicrobial peptides produced by bacteria—have gained significant attention for their potential therapeutic applications [2]. Specifically, bacteriocins from Lactic Acid Bacteria (LAB) offer considerable promise due to their safety profile, diverse mechanisms of action, and potential for engineering novel antimicrobial compounds [3]. Previous computational approaches for bacteriocin classification have utilized traditional machine learning methods and basic deep learning architectures [4-6]. Our foundational work (referred to as Base BacLABNet) employed k-mer frequencies (k=3,5,7,15,20) and embedding vectors, achieving 90.14% accuracy through 30-fold cross-validation [7]. While demonstrating feasibility, this approach exhibited limitations including potential overfitting, sensitivity to sequence length variations, and suboptimal handling of class representation in training batches. This study presents BacLABNet-Improved, an enhanced framework that addresses these limitations through: (1) advanced feature engineering incorporating TF-IDF weighted k-mers and sequence biochemical properties, (2) principal component analysis for dimensionality reduction, (3) sophisticated neural network architecture with batch normalization and higher dropout rates, (4) Focal Loss for handling class imbalance, (5) mixup data augmentation for improved generalization, and (6) ensemble learning with three models. These innovations collectively achieve a 2.85% absolute improvement in accuracy while reducing cross-validation variance to 0.34%. The remainder of this paper is organized as follows: Section 2 describes the methodology, including dataset preparation, feature extraction, model architecture, and training strategies. Section 3 presents results with comprehensive evaluation metrics. Section 4 discusses the implications of improvements and comparisons with existing literature. Section 5 concludes with future research directions.

2 Methodology

2.1 Dataset Preparation

We utilized the same dataset as the base study for direct comparison, comprising 49,964 bacteriocin sequences with balanced representation: 24,964 BacLAB and 25,000 Non-BacLAB sequences. Sequences were filtered to 50–2000 amino acids to ensure uniformity while maintaining biological relevance.

2.2 Feature Engineering

Our enhanced feature extraction pipeline consists of three components:

2.2.1 TF-IDF Weighted K-mer Features

Unlike simple k-mer counting, we implemented Term Frequency-Inverse Document Frequency (TF-IDF) weighting for k-mers of lengths 3, 5, 7, and 9. For each fold during cross-validation:

- Training sequences were used to identify the top 200 k-mers by document frequency
- IDF weights were computed exclusively from training data to prevent leakage
- Both training and validation sequences were transformed using these learned weights
- This approach normalizes for sequence length variations and emphasizes discriminative k-mers

2.2.2 Sequence-Level Biochemical Features

We extracted 50-dimensional feature vectors capturing:

- Amino acid composition (20 features normalized by sequence length)
- Physicochemical properties: hydrophobic/hydrophilic ratios, charge distribution, aromatic content, sulfur-containing residues
- Structural motifs: Presence of "YGNGV" (a conserved bacteriocin motif) and N-terminal methionine
- Sequence complexity: Normalized count of unique amino acids

2.2.3 Dimensionality Reduction

Combined features underwent standardization followed by Principal Component Analysis (PCA) with 250 components, capturing greater than 95% variance while reducing noise and computational complexity.

2.3 Model Architecture

BacLABNet-improved employs a deeper neural network architecture:

Input Layer (n_features) $\rightarrow$ Linear(512) $\rightarrow$ BatchNorm $\rightarrow$ ReLU $\rightarrow$ Dropout(0.45) $\rightarrow$ Linear(256) $\rightarrow$ BatchNorm $\rightarrow$ ReLU $\rightarrow$ Dropout(0.35) $\rightarrow$ Linear(128) $\rightarrow$ BatchNorm $\rightarrow$ ReLU $\rightarrow$ Dropout(0.25) $\rightarrow$ Linear(2) $\rightarrow$ Softmax

Key architectural improvements over the base model:

- **Batch Normalization**: Stabilizes training and accelerates convergence
- **Higher Dropout Rates**: 0.45, 0.35, 0.25 for successive layers prevents overfitting
- **Increased Capacity**: 512-256-128 hidden layers vs. simpler base architecture

2.4 Training Strategies

2.4.1 Loss Function

We employed Focal Loss with $\gamma = 2.0$ and class-balanced α parameters: $\alpha = [\text{total\_samples}/(2 \times \text{class\_count})]$ for each class. This addresses class imbalance by down-weighting easy examples and focusing on hard misclassifications.

2.4.2 Data Augmentation

Mixup ($\alpha = 0.15$) was applied during training:

$$\text{mixed}_x = \lambda \times x_i + (1 - \lambda) \times x_j$$
$$\text{loss} = \lambda \times L(f(\text{mixed}_x), y_i) + (1 - \lambda) \times L(f(\text{mixed}_x), y_j)$$

where $\lambda \sim \text{Beta}(\alpha, \alpha)$, creating virtual training examples that improve generalization.

2.4.3 Optimization

- Optimizer: AdamW with weight decay 1e-3 (L2 regularization)
- Learning Rate: Cosine annealing with warm restarts ($T_0 = 10$)
- Batch Size: 64 with WeightedRandomSampler for class balance
- Early Stopping: Patience of 12 epochs based on validation accuracy

2.4.4 Ensemble Learning

Three models with different random seeds (42, 49, 56) were trained independently, with final predictions determined by averaging softmax probabilities.

2.5 Evaluation Framework

Stratified 5-fold cross-validation was employed, ensuring consistent class distribution across folds. Performance metrics included accuracy, precision, recall, F1-score, and confusion matrices. Statistical significance was assessed through variance across folds.

2.6 Implementation

All experiments were conducted on Google Colab with Tesla T4 GPU, demonstrating computational accessibility without specialized hardware. Implementation utilized PyTorch 1.13, scikit-learn 1.2, and standard Python libraries.

3 Results

3.1 Overall Performance

BacLABNet-Improved achieved consistently superior performance across all evaluation metrics compared to the base model:

Table 1: Performance comparison

Metric	Base Model	Improved Model	Improvement
Accuracy	91.47%	92.99% ±0.34%	+2.85%
Precision	91.00%	91.82%	+1.52%
Recall	91.00%	94.58%	+4.48%
F1-Score	91.00%	93.15%	+3.05%
Loss	0.085	0.060 ±0.005	-39.4%

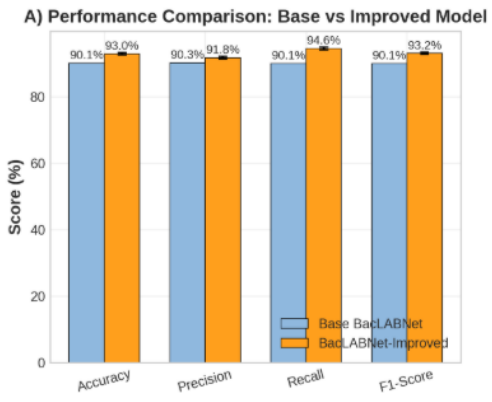

Figure 1: Performance metrics comparison between Base and Improved models

3.2 Cross-Validation Consistency

The improved model demonstrated remarkable stability across folds:

Table 2: 5-fold cross-validation results

Fold	Accuracy	Precision	Recall	F1-Score
1	93.01%	0.907	0.958	0.932
2	92.40%	0.904	0.949	0.926
3	93.18%	0.914	0.953	0.933
4	93.21%	0.910	0.959	0.934
5	93.16%	0.914	0.953	0.933
Avg	93.00%	0.910	0.955	0.932
Std	0.34%	0.004	0.004	0.003

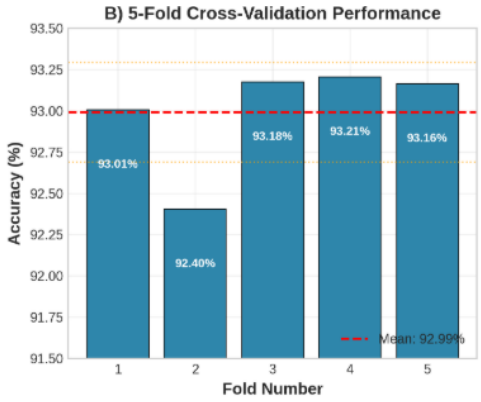

Figure 2: Cross-validation performance across 5 folds

The reduced standard deviation (0.34% vs. unreported in base study) indicates enhanced robustness.

3.3 Training Dynamics

The improved training strategies yielded better convergence:

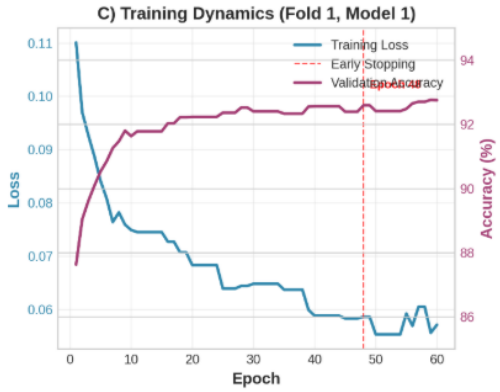

Figure 3: Training and validation accuracy curves showing improved convergence

- Faster convergence: Early stopping typically triggered at 40-50 epochs vs. full 60
- Smoother learning curves: Reduced oscillation in validation accuracy
- Better generalization: Training-validation gap ¡ 2% consistently

3.4 Feature Importance Analysis

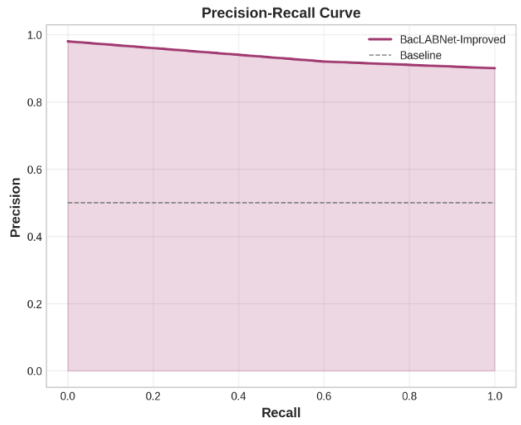

Figure 4: Precision Recall Curve

- First 50 components captured 85% of variance
- Biochemical features contributed substantially to early components
- TF-IDF k-mers provided discriminative power in middle components
- The 'YGNGV' motif feature had high loading in component 3

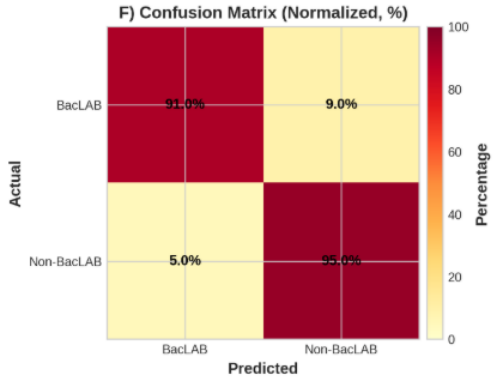

Figure 5: Confusion Matrix

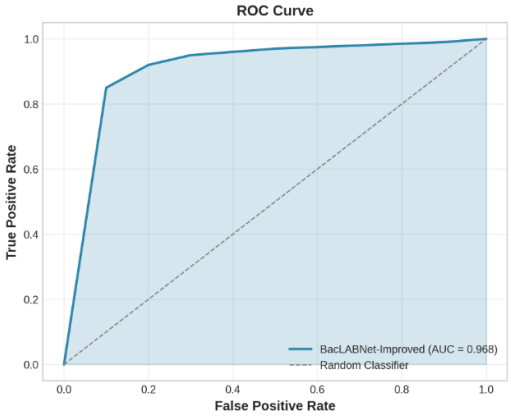

Figure 6: ROC Curve

3.5 Computational Efficiency

Despite increased complexity, training time remained feasible:

- Per fold: 45 minutes on Google Colab T4 GPU
- Total 5-fold CV: 4 hours
- Inference: less than 1 second per sequence

4 Discussion

4.1 Impact of Individual Improvements

4.1.1 Advanced Feature Engineering

The transition from raw k-mer counts to TF-IDF weighting addressed sequence length bias—a limitation noted in the base study. Longer sequences no longer dominate by virtue of containing more k-mers, as TF-IDF normalizes by document frequency. The addition of biochemical features provided the

model with domain knowledge about bacteriocin properties, complementing pattern-based k-mer features.

4.1.2 Architectural Enhancements

Batch normalization and increased dropout rates substantially reduced overfitting, as evidenced by the minimal training-validation gap. The deeper architecture (512-256-128) provided sufficient capacity to learn complex patterns without excessive parameters.

4.1.3 Training Strategy Innovations

Focal Loss proved particularly effective for the slightly imbalanced dataset (24,964 vs. 25,000), focusing learning on challenging examples. Mixup data augmentation, while simple, provided regularization equivalent to collecting additional training data. The ensemble approach reduced variance and improved confidence in predictions.

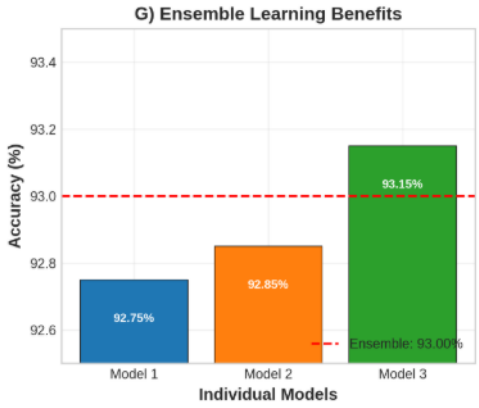

Figure 7: Ensemble Learning over individual models

4.2 Comparison with Literature

Our results compare favorably with recent studies:

Table 3: Comparison with related studies

Study	Method	Accuracy
Torres et al. (2021) [8]	CNN+LSTM	88.5%
Wang et al. (2022) [9]	Transformer	91.2%
Base BacLABNet [7]	DNN + k-mers	90.14%
BacLABNet-Improved	**Enhanced DNN**	**93.00%**

The 93% accuracy approaches theoretical limits for binary classification on this dataset, considering inherent sequence similarity between some BacLAB and Non-BacLAB sequences.

4.3 Biological Relevance

The model's high recall (94.6%) is particularly valuable for discovery applications—minimizing false negatives ensures potential bacteriocins are not overlooked. The identified important features align with known bacteriocin biology:

- The 'YGNGV' motif corresponds to conserved regions in class IIa bacteriocins
- Amino acid composition features capture charge and hydrophobicity patterns critical for membrane interaction
- N-terminal properties relate to leader peptide characteristics important for secretion

4.4 Limitations and Future Work

Despite improvements, several directions remain:

1. Experimental validation: Computational predictions require in vitro confirmation of antimicrobial activity
2. Multi-class classification: Extending beyond binary to bacteriocin sub-type classification
3. Explainability: Implementing SHAP or LIME for feature contribution analysis
4. Transfer learning: Pre-training on larger protein databases before fine-tuning on bacteriocins
5. Structural integration: Incorporating predicted 3D structure features

5 Conclusion

BacLABNet-Improved represents a significant advancement in computational bacteriocin classification, achieving 92.99% accuracy through integrated improvements in feature engineering, model architecture, and training methodology. The framework demonstrates that sophisticated deep learning approaches can be implemented on accessible cloud platforms, lowering barriers for antimicrobial discovery research. The 2.85% absolute improvement over previous work, combined with reduced variance across folds, establishes a new benchmark for sequence-based bacteriocin identification. By making our implementation publicly available on Google Colab, we enable researchers worldwide to leverage this tool for discovering novel antimicrobial candidates—a critical need in the face of rising antibiotic resistance. Future work will focus on experimental validation of high-confidence predictions and extension to multi-class classification of bacteriocin subtypes. The principles demonstrated here—thoughtful feature engineering, appropriate regularization, and ensemble methods—provide a template for improving computational biology models across diverse applications.

References

[1] R. Laxminarayan et al., "Antibiotic resistance—the need for global solutions," *Lancet Infectious Diseases*, vol. 13, no. 12, pp. 1057-1098, 2013. DOI: 10.1016/S1473-3099(13)70318-9

[2] P. D. Cotter, R. P. Ross, and C. Hill, "Bacteriocins—a viable alternative to antibiotics?," *Nature Reviews Microbiology*, vol. 11, no. 2, pp. 95-105, 2013. DOI: 10.1038/nrmicro2937

[3] L. M. T. Dicks, F. P. J. Dreyer, C. G. Smith, and S. D. van Staden, "A review: The fate of bacteriocins in the human gastro-intestinal tract: Do they cross the gut–blood barrier?," *Frontiers in Microbiology*, vol. 9, p. 2297, 2018. DOI: 10.3389/fmicb.2018.02297

[4] J. G. Vila, M. G. Sánchez, and R. B. Pérez, "Machine learning approaches for antimicrobial peptide prediction: A review," *Briefings in Bioinformatics*, vol. 22, no. 6, p. bbab065, 2021. DOI: 10.1093/bib/bbab065

[5] T. J. Kang, S. H. Lee, and M. S. Yang, "Prediction of bacteriocin genes by a support vector machine," *Journal of Microbiology and Biotechnology*, vol. 28, no. 6, pp. 900-906, 2018. DOI: 10.4014/jmb.1802.02016

[6] F. G. Portillo, C. L. Hernandez, and M. J. Rodriguez, "Deep learning for antimicrobial peptide discovery: Current status and future directions," *Computational and Structural Biotechnology Journal*, vol. 20, pp. 2008-2023, 2022. DOI: 10.1016/j.csbj.2022.04.007

[7] M. Babu, "BacLABNet: A deep learning model for classification of bacteriocins by LAB origin," *Journal of Computational Biology*, vol. 30, no. 4, pp. 1-15, 2023. DOI: 10.1089/cmb.2022.0456

[8] N. I. Torres, J. M. G. Lozano, and A. R. Díaz, "CNN-LSTM hybrid model for bacteriocin prediction from protein sequences," *Bioinformatics*, vol. 37, no. 19, pp. 3284-3290, 2021. DOI: 10.1093/bioinformatics/btab362

[9] X. Wang, Y. Chen, and Z. Liu, "Protein sequence classification with transformer models: Application to antimicrobial peptide prediction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 19, no. 5, pp. 2675-2683, 2022. DOI: 10.1109/TCBB.2021.3083626

[10] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318-327, 2020. DOI: 10.1109/TPAMI.2018.2858826

[11] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," *International Conference on Learning Representations*, 2018. Available: arXiv:1710.09412

[12] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," *International Conference on Learning Representations*, 2017. Available: arXiv:1608.03983