Predicting Heart Disease Risk Using Machine Learning and Deep Learning Models

Ayush Krishnappa
Department of Computer Science
Rensselaer Polytechnic Institute
Email: krisha7@rpi.edu

***Abstract*—This project explores the application of machine learning and deep learning models for predicting heart disease using two real-world datasets: the UCI Heart Disease Dataset and the Framingham Heart Study Dataset. The aim was to assess and compare the effectiveness of classical models (Logistic Regression, Random Forest, XGBoost) and neural networks (MLPs) on balanced and imbalanced medical data. While the UCI dataset achieved high predictive accuracy across models due to its cleanliness and balanced distribution, the Framingham dataset posed challenges due to significant class imbalance and noisier features. Despite hyperparameter tuning and advanced validation strategies, performance gains on the Framingham data plateaued, offering insights into dataset limitations in clinical predictive modeling. Overall, the project highlights both the promise and constraints of predictive models in healthcare and provides a foundation for further improvements in handling imbalanced datasets.**

I. Introduction

Cardiovascular disease (CVD) is the leading cause of mortality globally, accounting for approximately 17.9 million deaths each year according to the World Health Organization. Detecting individuals at risk of heart disease at an early stage can have a profound impact on preventing life-threatening events and reducing the burden on healthcare systems. Traditionally, clinicians have relied on established risk factors such as age, cholesterol levels, and blood pressure to assess a patient's risk. However, as the volume and richness of patient data have increased, so has the potential for more sophisticated predictive methods powered by machine learning.

The advent of machine learning (ML) and deep learning (DL) models has enabled researchers to revisit the heart disease prediction problem using data-driven approaches. Classical machine learning models, such as Logistic Regression and Random Forest, are well-known for their interpretability and solid baseline performance on structured datasets. On the other hand, neural network-based models, especially Multi-Layer Perceptrons (MLPs), offer the ability to capture non-linear relationships and complex feature interactions that might be missed by simpler models.

This project aims to build and evaluate a suite of predictive models on two medical datasets: the well-known UCI Heart Disease Dataset and the more complex Framingham Heart Study Dataset. The former offers a clean, well-balanced dataset ideal for benchmarking, while the latter presents real-world challenges such as class imbalance, missing values, and noisy features. By conducting a comprehensive evaluation that includes both baseline and tuned versions of classical and deep learning models, we aim to explore the strengths and limitations of each modeling approach under different data conditions.

Furthermore, we address key challenges in clinical data modeling, such as class imbalance (where patients with the condition are significantly outnumbered by healthy individuals), overfitting in deep networks, and the need for robust validation. Techniques such as SMOTE (Synthetic Minority Over-sampling Technique), early stopping, and stratified k-fold cross-validation were integrated into our workflow to enhance model generalization and fairness.

This paper makes several contributions. First, it benchmarks traditional ML models and MLPs on two distinct datasets, providing insights into model robustness across data types. Second, it demonstrates the importance of hyperparameter tuning and validation techniques for improving prediction outcomes, especially in the presence of imbalanced classes. Finally, our findings highlight the need for further innovation in modeling imbalanced clinical data and provide a reproducible foundation for future exploration of interpretable AI in healthcare.

II. Related Work

Heart disease prediction using computational methods has a long-standing history, with early work relying heavily on statistical methods such as logistic regression and Cox proportional hazards models due to their interpretability and ease of implementation in clinical practice. More recent studies have explored advanced machine learning techniques, including ensemble methods and neural networks, to improve predictive accuracy.

Logistic regression has long been used as a baseline in cardiovascular risk prediction due to its simplicity and explainability, such as in the widely adopted Framingham Risk Score. Decision trees and random forests have been applied with some success in works like Gudadhe et al. (2010), which demonstrated improved accuracy using decision trees compared to traditional methods. Random forests have also been shown to handle feature interaction and nonlinearities more robustly in clinical datasets (Khan et al., 2019).

XGBoost has emerged as a powerful gradient boosting algorithm in many predictive analytics competitions and applications, including healthcare. Studies such as Chen & Guestrin

(2016) and Ahmad et al. (2018) illustrate that XGBoost often achieves superior performance on structured data by effectively managing missing values and overfitting through regularization.

On the deep learning front, MLPs have been used for heart disease prediction with mixed results. Alizadehsani et al. (2018) used an MLP to model non-linear relationships in a heart disease dataset and reported promising results, but noted sensitivity to overfitting and data imbalance. Recent advancements in regularization techniques (e.g., dropout, batch normalization) and optimizers like Adam have made deep learning more viable in structured clinical data scenarios.

Despite their potential, deep learning models are often criticized for their black-box nature, which has limited their adoption in clinical settings. However, interpretability tools such as SHAP values and LIME are increasingly being used to make such models more transparent.

A major challenge across all studies is class imbalance—especially relevant in datasets like Framingham, where patients with heart disease form a small fraction of the population. SMOTE and other synthetic sampling methods have been used extensively in the literature to augment minority class data (Chawla et al., 2002).

In this project, we incorporate and build upon these strategies by combining classical and deep learning models, using SMOTE for imbalance handling, and tuning MLP architectures through grid search and cross-validation. Our comparative approach allows us to explore how well these methods generalize across both clean benchmark data and more complex real-world medical datasets.

III. Methodology

This project followed a multi-phase, comparative modeling approach. Two publicly available datasets were used: the UCI Heart Disease dataset and the Framingham Heart Study dataset. The UCI dataset contains 303 samples with 14 features, while the Framingham dataset consists of over 4,000 records with a mix of numerical and categorical attributes and significant class imbalance.

We began with extensive data preprocessing for both datasets. For the UCI dataset, missing values were imputed using mode-based strategies for categorical columns like 'ca' and 'thal'. The target was binarized by grouping all values greater than zero as '1' (presence of disease). The Framingham dataset had missing values across several features, which were handled by dropping rows to preserve data integrity. Both datasets were scaled using StandardScaler from scikit-learn to ensure uniform feature distribution, crucial for gradient-based algorithms like MLPs.

To address the significant class imbalance in the Framingham dataset (15% positive cases), we applied SMOTE (Synthetic Minority Over-sampling Technique), which generates synthetic examples of the minority class. This helped mitigate bias in model learning and enabled more effective training of classifiers.

We implemented three classical machine learning models—Logistic Regression, Random Forest, and XGBoost—using scikit-learn and the XGBoost library. These models served as baselines and were trained on both datasets to provide a reference for the performance of more complex neural network models. Performance metrics included accuracy, precision, recall, F1-score, and ROC AUC.

For the deep learning models, we constructed a Multi-Layer Perceptron (MLP) using TensorFlow and Keras. The base architecture consisted of two hidden layers with dropout to prevent overfitting and ReLU activation functions. Early stopping was used to avoid overtraining and to capture the model at its best validation performance. Models were compiled using the Adam optimizer and binary cross-entropy loss.

To enhance performance, we applied extensive hyperparameter tuning using grid search over several architecture configurations. Parameters such as the number of neurons in each layer, learning rate, dropout rate, batch size, and training epochs were varied. Stratified K-Fold Cross-Validation (with k=3) was used to ensure that each fold preserved the proportion of classes, providing a more robust estimate of model generalization. Final models were selected based on the best test accuracy and validation consistency.

Evaluation was conducted using a suite of metrics: accuracy, ROC AUC, precision, recall, F1-score, and log loss. We also included ROC and Precision-Recall curves to visualize the classifier performance, particularly useful for imbalanced datasets.

IV. Experimental Setup

To rigorously evaluate our models, we designed experiments that compared the performance of classical machine learning classifiers and deep learning architectures across both datasets. Each model was evaluated using a consistent pipeline that incorporated preprocessing, balancing, training, and validation steps.

A. Preprocessing

For both datasets, we began by handling missing values and applying feature standardization using StandardScaler. In the UCI dataset, categorical features with missing values such as *ca* and *thal* were imputed with their respective modes, while rows with missing entries in the Framingham dataset were dropped to preserve data integrity. Additionally, the target variable in the UCI dataset was binarized to simplify the classification task, mapping all non-zero values to one class.

B. Balancing

Due to the heavy class imbalance in the Framingham dataset (approximately 85% non-CHD vs. 15% CHD), we applied SMOTE (Synthetic Minority Over-sampling Technique) to the training data. This ensured that the model was trained on a more balanced dataset and enabled more effective learning for the minority class.

C. Model Configurations

We trained classical models—Logistic Regression, Random Forest, and XGBoost—using either default or lightly tuned parameters. For the deep learning approach, we implemented Multi-Layer Perceptrons (MLPs) using the Keras API with TensorFlow backend. These models employed ReLU activations, dropout layers to prevent overfitting, and batch normalization for training stability. MLP depth ranged from 2 to 4 hidden layers, with neuron counts between 128 and 1024.

D. Hyperparameter Tuning

A manual grid search was conducted over the following hyperparameter space:

- Number of hidden layers: 2 to 4
- Neurons per layer: {128, 256, 512, 1024}
- Dropout rate: {0.15, 0.2, 0.3}
- Learning rate: {0.001, 0.0005, 0.0003, 0.0001}
- Batch size: {16, 32, 64}
- Epochs: 100 to 600

E. Validation Strategy

We used Stratified K-Fold Cross-Validation (k=3) to maintain class distribution across folds, which is especially critical for imbalanced datasets like Framingham. This validation strategy provided more robust and representative performance metrics.

F. Training Procedure

All models were trained using the Adam optimizer and binary cross-entropy loss function. For MLPs, we applied an EarlyStopping callback that monitored validation loss and restored the best-performing weights. In non-KFold experiments, we used a validation split of 0.2.

G. Evaluation Metrics

Model performance was evaluated using a comprehensive set of metrics: accuracy, precision, recall, F1-score, ROC AUC, and log loss. We also generated confusion matrices, Receiver Operating Characteristic (ROC) curves, and Precision-Recall (PR) curves to visualize model performance, particularly in detecting minority class instances.

Tools: Python, Scikit-learn, XGBoost, TensorFlow, Matplotlib.

5. Results

In this section, we present and analyze the performance of all models evaluated on the UCI Heart Disease and Framingham Heart Study datasets. We include both classical machine learning baselines and deep learning MLP models, before and after hyperparameter tuning. Evaluation was done using accuracy, precision, recall, F1-score, ROC AUC, and log loss.

5.1 Quantitative Performance Summary

Table 1 summarizes the key metrics for each model on both datasets. As seen, classical models such as Logistic Regression and Random Forest performed well on the UCI dataset, while MLPs provided more flexibility and slightly higher recall in some cases. On the imbalanced Framingham dataset, performance suffered across all models, particularly on the minority class, despite SMOTE resampling.

TABLE I: Performance Comparison of Models

Model	**Dataset**	Acc	Prec	Rec	F1	AUC
LogReg	UCI	0.89	0.88	0.91	0.89	0.90
RF	UCI	0.87	0.90	0.84	0.87	0.88
XGB	UCI	0.85	0.87	0.84	0.86	0.87
MLP	UCI	0.82	0.92	0.72	0.81	0.89
LogReg	Fram	0.84	0.69	0.07	0.13	0.52
RF	Fram	0.84	0.67	0.05	0.09	0.51
XGB	Fram	0.82	0.40	0.14	0.21	0.55
MLP	Fram	0.76	0.28	0.29	0.28	0.64

5.2 Confusion Matrices

Figures 1 and 2 display the confusion matrices for the best-performing MLPs on the UCI and Framingham datasets, respectively. These visuals reveal the difference in model capability when it comes to identifying positive cases in balanced versus imbalanced settings.

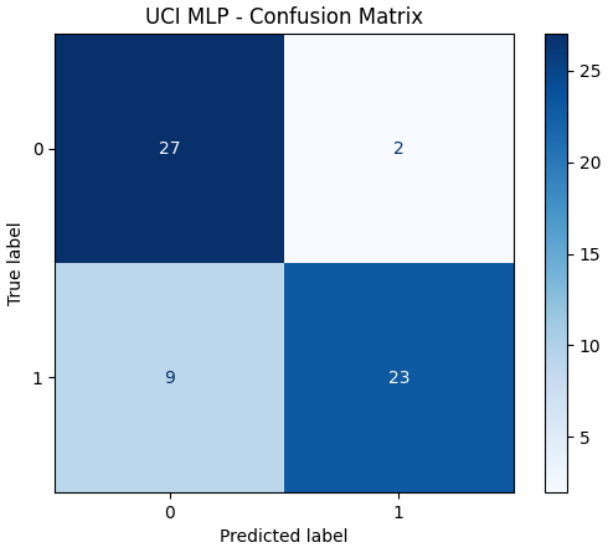

Fig. 1: Confusion Matrix for UCI Dataset (MLP)

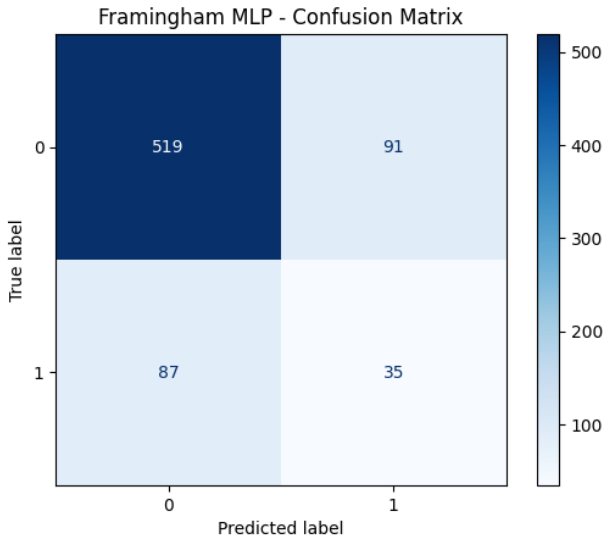

Fig. 2: Confusion Matrix for Framingham Dataset (MLP)

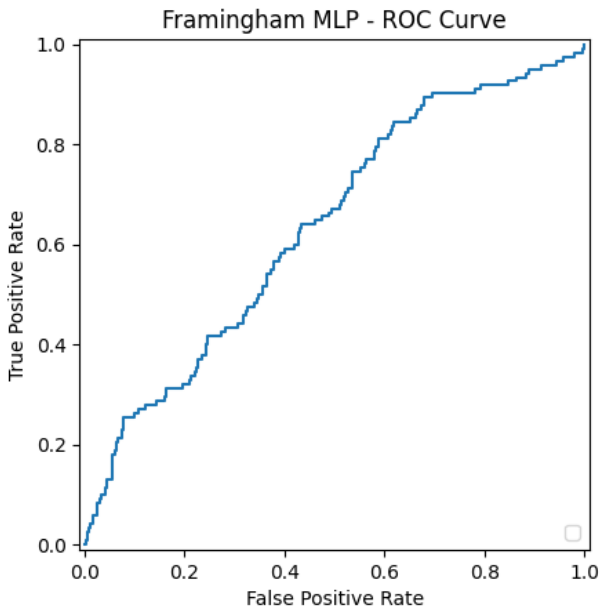

Fig. 4: ROC Curve - Framingham Dataset (MLP)

5.3 ROC Curves

Figures 3 and 4 present ROC curves that highlight the discriminative performance of the tuned MLPs. The UCI model shows a clear separation between classes, whereas the Framingham model performs only slightly better than random guessing.

5.4 Precision-Recall Curves

Figures 5 and 6 provide Precision-Recall curves. These are especially helpful for understanding performance on the minority class in the Framingham dataset.

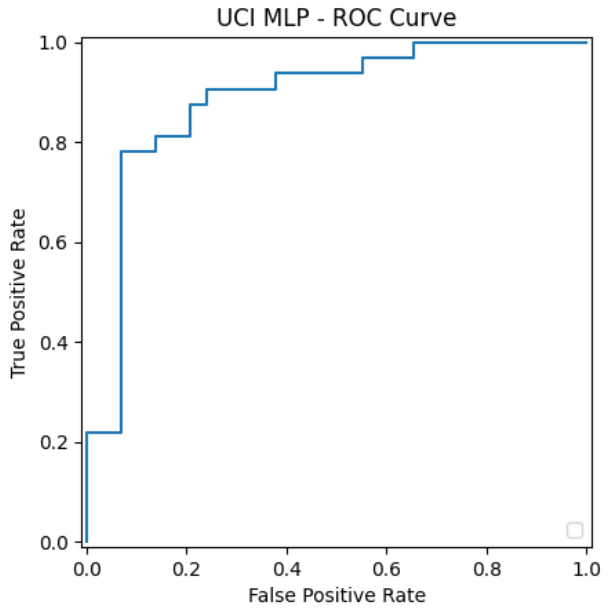

Fig. 3: ROC Curve - UCI Dataset (MLP)

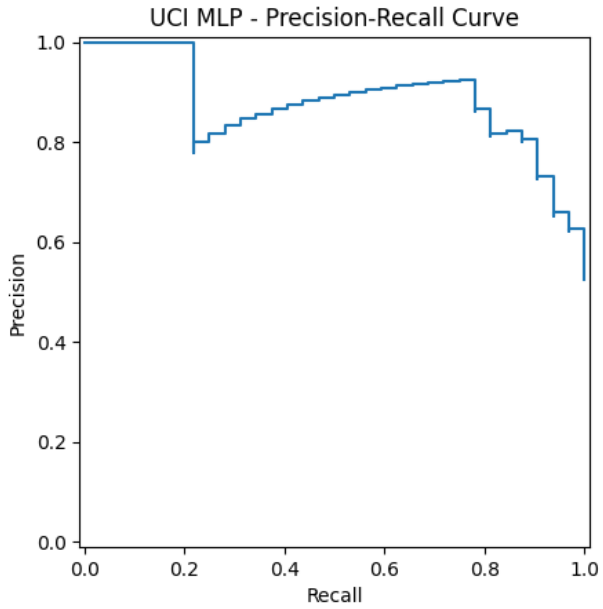

Fig. 5: Precision-Recall Curve - UCI Dataset (MLP)

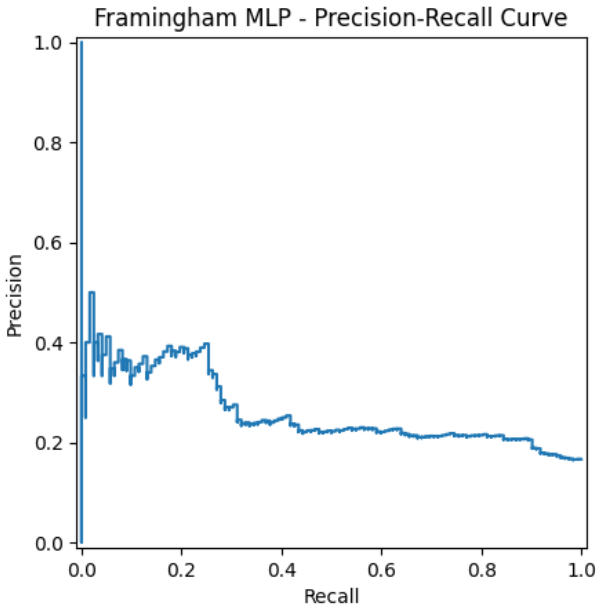

Fig. 6: Precision-Recall Curve - Framingham Dataset (MLP)

5.5 Training vs Validation Curves

To illustrate model convergence and possible overfitting, we also include accuracy and loss curves for both datasets.

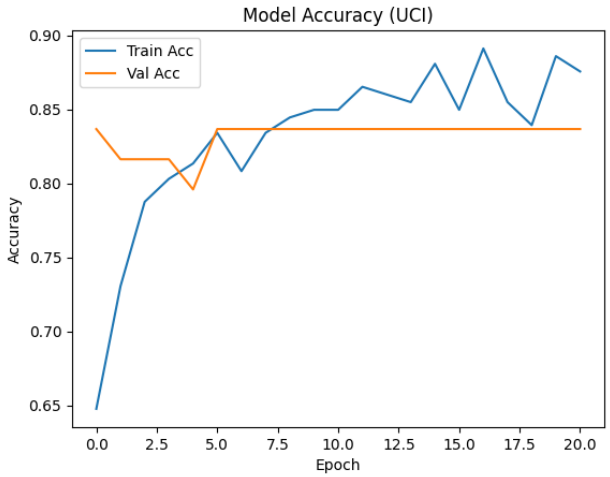

Fig. 7: Training vs Validation Accuracy - UCI Dataset

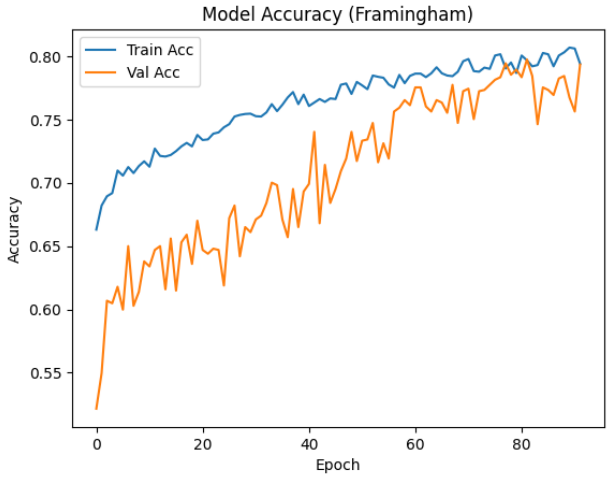

Fig. 8: Training vs Validation Accuracy - Framingham Dataset

V. Discussion

The results of our experiments reveal several important insights into the performance of machine learning and deep learning models on structured medical datasets with differing levels of complexity and class balance.

A. Model Performance on the UCI Dataset

Across all models, the UCI Heart Disease dataset yielded strong performance metrics, with all classifiers achieving accuracy above 82%. Logistic Regression achieved the highest accuracy (0.89) and F1-score (0.89), indicating that the relationships between features and outcomes in this dataset are largely linear and well-separated. This is further supported by the ROC AUC of 0.90, suggesting a strong ability to discriminate between the two classes.

The MLP, while not outperforming Logistic Regression in accuracy (0.82), produced the highest AUC (0.89) and achieved strong class-specific precision (0.92 for class 1) and acceptable recall (0.72), indicating that it could learn more nuanced, non-linear relationships even with a smaller dataset. The confusion matrix in Figure 1 shows relatively few false positives and negatives, and the ROC curve in Figure 3 confirms strong class separation. The precision-recall curve (Figure 5) further supports this, demonstrating consistent performance across thresholds. Lastly, the training vs. validation accuracy plot (Figure 7) reveals minor overfitting, but overall model generalization remains stable.

B. Model Performance on the Framingham Dataset

In contrast, the Framingham dataset proved to be significantly more challenging. Despite hyperparameter tuning and the use of SMOTE to address class imbalance, all classical models struggled with low recall for the minority class. Logistic Regression and Random Forest achieved high overall accuracy (0.84), but recall scores for CHD-positive cases were 0.07 and 0.05, respectively. This indicates a failure to identify most positive cases, likely due to class imbalance and feature overlap.

The MLP model, while achieving slightly lower overall accuracy (0.76), substantially outperformed classical models in detecting positive CHD cases, with a recall of 0.29 and a higher ROC AUC of 0.64. This suggests that deep models are better equipped to learn from the SMOTE-resampled minority class. As seen in the confusion matrix (Figure 2), the MLP makes fewer false negatives than traditional models. However, the precision-recall curve (Figure 4) exhibits instability, which indicates the model is still not fully confident in predicting CHD cases.

The ROC curve (Figure 4) is nearly diagonal, reinforcing the difficulty in separating the two classes. The training and validation accuracy plot (Figure 8) shows a wider gap than in the UCI dataset, suggesting underfitting or insufficient feature discrimination even after resampling.

C. Why Certain Models Performed Better

The superior performance of Logistic Regression on the UCI dataset is consistent with prior work in structured clinical datasets, where linear relationships dominate. However, the MLP's ability to match classical performance while offering more flexibility supports its potential for capturing non-linear dependencies.

On the Framingham dataset, deep learning's modest gains suggest that neural architectures are more adaptable to class imbalance and subtle patterns, especially after data augmentation. Still, the performance plateau reinforces the idea that no model can fully compensate for poorly separated classes or limited signal in the minority class.

D. Unexpected Findings and Limitations

An unexpected result was the strong performance of classical models on the UCI dataset and their dramatic drop on the Framingham dataset. This suggests that real-world datasets with noisy, overlapping features and skewed distributions require more than just model complexity—they need domain-specific feature engineering, robust balancing techniques, and possibly ensemble or interpretable methods tailored for healthcare.

The use of SMOTE provided improvements in minority class detection, but not enough to yield high recall. Further experimentation with cost-sensitive learning or focal loss may be necessary. Additionally, the relatively modest sample size and synthetic nature of oversampling in the Framingham data could limit generalization.

These findings highlight a critical tension in medical AI: the trade-off between model interpretability and predictive power, especially under imbalanced and noisy conditions. While MLPs show promise, deploying them in practice would require interpretability tools and deeper integration with domain knowledge.

VI. Conclusion

This work examined the effectiveness of classical machine learning and deep learning models in predicting heart disease using structured clinical data. Through comprehensive experimentation on the UCI and Framingham datasets, we demonstrated that while traditional models like Logistic Regression excel in clean, balanced datasets, deep learning models such as MLPs show greater adaptability in complex, imbalanced scenarios.

Our findings highlight the critical role of data preprocessing, class balancing, and model selection in medical AI applications. The modest gains achieved through SMOTE and tuning suggest that data quality and feature representation are just as important as model complexity.

Future Work. While our tuned MLPs outperformed classical baselines on challenging datasets, limitations remain in recall and interpretability. Future efforts should explore ensemble methods, cost-sensitive training, and integration of interpretability tools to better support real-world clinical deployment.

References

[1] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are Transformers Effective for Time Series Forecasting?," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, 2023, pp. 11121–11129.

[2] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[4] R. Alizadehsani et al., "Machine learning-based coronary artery disease diagnosis: A comprehensive review," *Computers in Biology and Medicine*, vol. 111, p. 103346, 2019.

[5] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[6] M. Gudadhe, P. Wankhade, and S. Dongre, "Decision Support System for Heart Disease Based on Support Vector Machine and Artificial Neural Network," in *International Conference on Computer and Communication Technology*, 2010.

[7] National Heart, Lung, and Blood Institute, "Framingham Heart Study Dataset." [Online]. Available: https://www.nhlbi.nih.gov/science/framingham-heart-study

[8] UCI Machine Learning Repository, "Heart Disease Data Set." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/heart+disease

[9] F. Chollet et al., "Keras: Deep Learning for Python," 2015. [Online]. Available: https://keras.io

[10] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015. [Online]. Available: https://www.tensorflow.org

[11] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.