A Comparative Analysis of Machine Learning Classifiers for Heart Disease Prediction

1st Jiyi Cai
Department of Graduate CSE
Yeshiva University
New York, USA
jcai2@mail.yu.edu

2nd Farhat Jahan
Department of Graduate CSE
Yeshiva University
New York, USA
fjahan@mail.yu.edu

3rd Puspita Chowdhury
Department of Graduate CSE
Yeshiva University
New York, USA
pchowdhu@mail.yu.edu

***Abstract*—This project addresses the critical need for advanced, non-invasive prediction of heart disease by benchmarking three sophisticated machine learning paradigms: Deep Learning (MLP), Stacking Ensemble, and a Multi-Level Hybrid Stacking Ensemble. We utilized a comprehensive, feature-engineered dataset consolidated from four UCI sources (Cleveland, Hungarian, Switzerland, and VA). The study confirms the superior robustness of ensemble methods for this tabular clinical task. The single Deep Learning MLP model achieved a strong baseline AUC-ROC of 0.8934. However, both stacking architectures significantly surpassed this performance. The Hybrid Stacking Ensemble achieved the highest Test Accuracy of 0.8750, while the Stacking Ensemble (MLP Meta-Learner) delivered the superior discrimination ability with a peak Test AUC-ROC of 0.9316. This final AUC-ROC score demonstrates a critical advantage in clinical risk ranking, successfully improving upon the best single base model (Random Forest, 0.9265 AUC). The project concludes that the Stacking Ensemble architecture is the optimal SOTA model for reliable heart disease prediction on this dataset, offering high generalization and diagnostic utility essential for early risk assessment.**

I. Introduction

Cardiovascular diseases (CVDs) are the leading cause of death worldwide, which makes early risk assessment and diagnosis crucial to patient outcomes. Traditional diagnostic methods can be invasive, costly, and subject to variation in clinical interpretation. Machine learning presents an opportunity to improve clinical decision-making by detecting complex patterns in patient data that can indicate disease risk. Our project addresses a binary classification task that predicts the presence or absence of heart disease based on standard clinical metrics, including age, cholesterol levels, blood pressure, and other physiological indicators.

The primary objectives of this project are to develop a robust data preprocessing pipeline for medical data, implement and train three machine learning algorithms of varying complexity, compare model performance using standard evaluation metrics, and identify the most effective approach for heart disease prediction in this dataset.

The initial phase of this project focused on establishing a baseline using a limited subset of the data: the Cleveland Clinic Heart Disease dataset with only 13 core physiological attributes. This preliminary work implemented and tested standard classification algorithms, Logistic Regression, Decision Tree Classifier, and Random Forest Classifier—with initial accuracy estimates ranging from 75% to 85%. The initial scope was restricted to the basic preprocessing and preliminary testing of these traditional models.

The current objective is to elevate the predictive performance to a State-of-the-Art (SOTA) level through comprehensive data augmentation and the deployment of advanced ensemble techniques. The scope was expanded significantly by:

1) Data Enhancement: Consolidating records from four independent UCI Heart Disease databases (Cleveland, Hungarian, Switzerland, and VA) to create a larger, more diverse training pool.
2) Advanced Feature Engineering: Implementing sophisticated clinical composite features, expanding the input space to 24 highly informative features, including physiological ratios and severity scores.
3) Systematically comparing three high-performance architectures:
   - An optimized Deep Learning Multi-Layer Perceptron (MLP).
   - A Stacking Ensemble using Gradient Boosting Machines and an MLP Meta-Learner.
   - A Hybrid Stacking Ensemble employing a multi-level blending strategy with an XGBoost final blender, maximizing diversity.

The ultimate goal is to identify the architecture that yields the highest Area Under the Receiver Operating Characteristic Curve (AUC-ROC) score, confirming its superior ability to discriminate between the presence and absence of heart disease on unseen patient data.

II. Related Work

The UCI Cleveland heart disease dataset [1] has been extensively studied in medical informatics research. Early approaches [2], including work by Detrano et al. who created the dataset, achieved approximately 77% accuracy using logistic regression. More recent studies have employed advanced techniques including Support Vector Machines (SVM) and K-Nearest Neighbors (KNN), typically achieving accuracies between 80-88Ensemble methods such as Random Forests and Gradient Boosting[3] have gained popularity for their ability to handle noise in medical data. However, a recurring concern in the literature is the trade-off between accuracy and interpretability, particularly with complex ”black-box” models like

Deep Neural Networks[4]. Our work aims to systematically compare a baseline linear model against tree-based methods to balance performance with model transparency, a critical consideration for clinical applications.

III. Our Solution

This section elaborates on our pipeline for solving the heart disease prediction problem.

A. Description of Dataset

We obtained the Heart Disease dataset from the UCI Machine Learning Repository (Cleveland database). This dataset represents a consolidated collection of patient records from four primary sources: the Cleveland Clinic Foundation, the Hungarian Institute of Cardiology, Budapest, the V.A. Medical Center, Long Beach, and the University Hospital, Zurich, Switzerland. The dataset contains approximately 920 patient records with 24 continuous numerical and categorical features. The target variable introduced to indicate the presence or absence of heart disease.

Data Consolidation and Cleaning:

- Data Integration: Approximately 920 raw patient records were consolidated from the four main databases, representing a more diverse and robust feature space than any single source.
- Missing Value Imputation: Missing data points, encoded as '?', were converted to NaN. Imputation was performed using the median of the respective feature columns (e.g., for ca, thal, trestbps, and chol), a method chosen for its resistance to outliers.
- Target Binarization: The raw target variable (ranging from 0 to 4 for severity) was converted into a simple binary classification target: 0 (No Disease) and 1 (Disease Present).

Advanced Feature engineering:

To maximize the predictive signal, several composite clinical features were engineered.

- Ratios: Ratios such as cholesterol relative to maximum heart rate (chol_thalach_ratio) and ST depression relative to maximum heart rate (oldpeak_thalach_ratio) were created to capture physiological relationships.
- Severity Scores: A composite feature such as angina_severity_score was created to quantify the patient's chest pain profile based on chest pain type (cp) and exercise-induced angina (exang).
- Final Feature Set: The dataset was expanded to 24 features after One-Hot Encoding and feature creation.

Standardization and Transformation

- Scaling: All numerical features were subjected to Standard Scaling to ensure a mean of 0 and a standard deviation of 1, which is essential for stable convergence in Deep Learning models and improved performance in linear models.
- Transformation: Highly skewed distributions, specifically chol and oldpeak, were processed with a Quantile Transformer to enforce a uniform distribution, making them more suitable for certain model types.

B. Model Architectures

We have selected three methodologies representing different levels of complexity:

1) Multilayer Perceptron: This model was constructed as a single, highly optimized benchmark to determine the peak performance achievable by a single-entity classifier on the refined feature set.

- Paradigm: Feed-Forward Neural Network (FNN), implemented using PyTorch.
- Architecture Rationale: The model employed a classic "funnel" design to process and compress the 24 input features into increasingly abstract representations.
- Loss and Optimization: The binary classification task utilized the nn.BCEWithLogitsLoss function, which combines the Sigmoid activation and Binary Cross-Entropy loss for greater numerical stability, paired with the adaptive Adam optimizer.

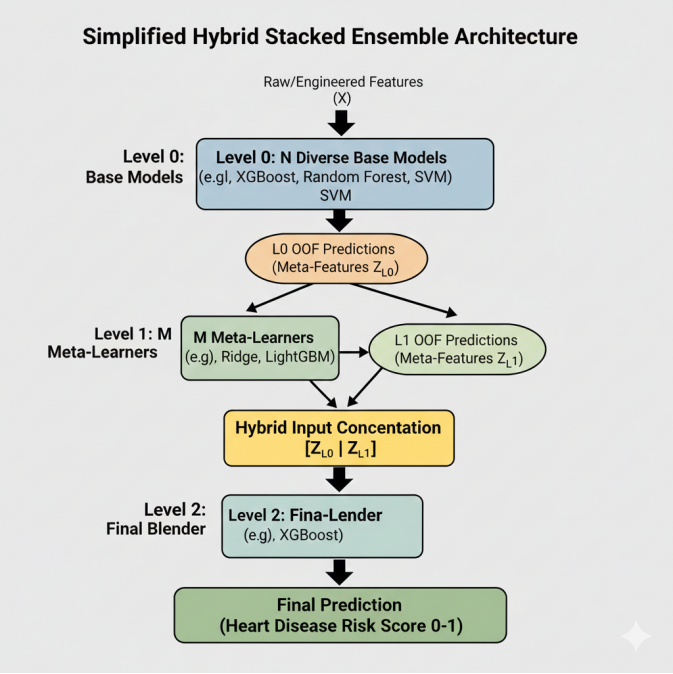

Fig. 1: Block Diagram of the Hybrid Stacked Ensemble Architecture.

2) Stacking Ensemble (MLP Meta_Learner): This architecture followed a two-level stacking strategy, emphasizing diversity at the base layer and leveraging the non-linear blending capability of a neural network at the meta-layer.

- Level 0 (Base Learners): The base layer comprised a high-performing suite of four Gradient Boosting machines: Random Forest, Gradient Boosting, XGBoost, and LightGBM. These models were chosen for their

effectiveness in tabular data and their complementary decision boundaries.

- Input Data Principle: A critical feature of this ensemble was the generation of Out-of-Fold (OOF) Predictions. The Level 0 models were trained using K-Fold Cross-Validation, and predictions for the entire training set were made only on the data folds that were not seen during training. This technique ensures the Level 1 meta-learner is trained on the true generalization error of the base models, eliminating data leakage and preventing overfitting.
- Level 1 (Meta-Learner): The input features for Level 1 were the four OOF prediction columns. The meta-learner was a simple Neural Network (MLP) designed to learn a complex, non-linear function for optimally weighting and combining the base model outputs.

3) Hybrid Stacking Ensemble: This architecture shown in Fig 1 represented the project's most complex and high-variance approach, utilizing three levels of prediction and blending for maximum robustness and predictive gain.

- Level 0 (L0) Base Learners: The base set was expanded to nine highly diverse models, including linear, kernel-based (SVM), and advanced tree-based algorithms. This diversity was intended to cover all possible predictive angles.
- Level 1 (L1) Meta Models: OOF predictions from L0 were passed to three distinct Level 1 blenders: Logistic Regression, Ridge Classifier, and LightGBM. This generated a second layer of refined, aggregated predictions.
- Level 2 (L2) Final Blender: The ultimate decision was made by a final XGBoost model. Crucially, the L2 model was trained on a hybrid feature space: it received both the original nine L0 OOF predictions and the three L1 refined predictions. This hybridization allowed the final model to selectively trust both individual models and intermediate blending strategies.

C. Implementation Details

The project was implemented in Python using the Scikit-learn library. We split the processed data into a training set (80%) and a testing set (20%) using a fixed random seed to ensure reproducibility. We have established a consistent evaluation framework using accuracy, precision, recall, and F1-score metrics.

1) MLP:

- **Framework**: Implemented entirely in PyTorch, leveraging nn.Module for the custom network design.
- Regularization: To prevent overfitting, two key techniques were employed:
  - **Batch Normalization**: Applied after each hidden layer's linear transformation to stabilize learning dynamics and accelerate convergence.
  - Dropout: A dropout rate of 0.3 was applied after the first hidden layer and 0.2 after the second, randomly deactivating neurons to prevent co-adaptation.
- **Training Protocol**: The model was trained using the Adam optimizer with a low learning rate ($1e-4$) over 100 epochs, monitored by an Early Stopping mechanism set to a patience of 15 epochs based on the validation AUC-ROC.

2) Stacking Ensemble:

- **Level 1 (Meta-Learner)**: A compact Multi-Layer Perceptron (MLP) was used as the final blender.
  - The network took only four input features (the OOF predictions from the L0 models).
  - This non-linear meta-learner learns an optimal, complex weighting scheme for the base models.
- **Final Prediction**: The Level 1 MLP was trained on the 4 OOF features from the training set and then used to predict the final probabilities on the 4 predictions generated by the L0 models on the separate test set.

3) Hybrid Model: The process involves constructing three distinct levels of predictive models, with each subsequent level learning to correct the errors and combine the insights from the level preceding it.

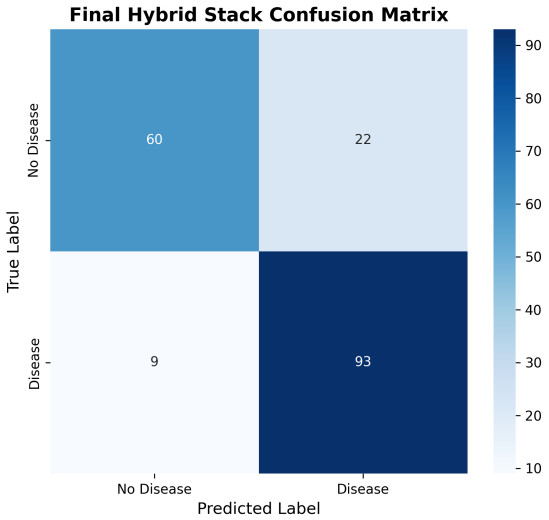

Fig. 2: Risk of type I and type II errors

1) **Level 0 - Base Model Diversity and Out-of-Fold (OOF) Prediction Generation**: The foundation of the ensemble utilizes nine distinct models to ensure maximum diversity in their predictive mechanisms:
   - Model selection: The nine Level 0 Base Learners included linear Models such as Highly regularized Logistic Regression and Ridge Classifier, non-linear models such as Support Vector Machine (SVM) and K-Nearest Neighbors (KNN), and advanced tree ensembles such as Random Forest, Extra Trees, Gradient Boosting, XGBoost[5], and LightGBM.

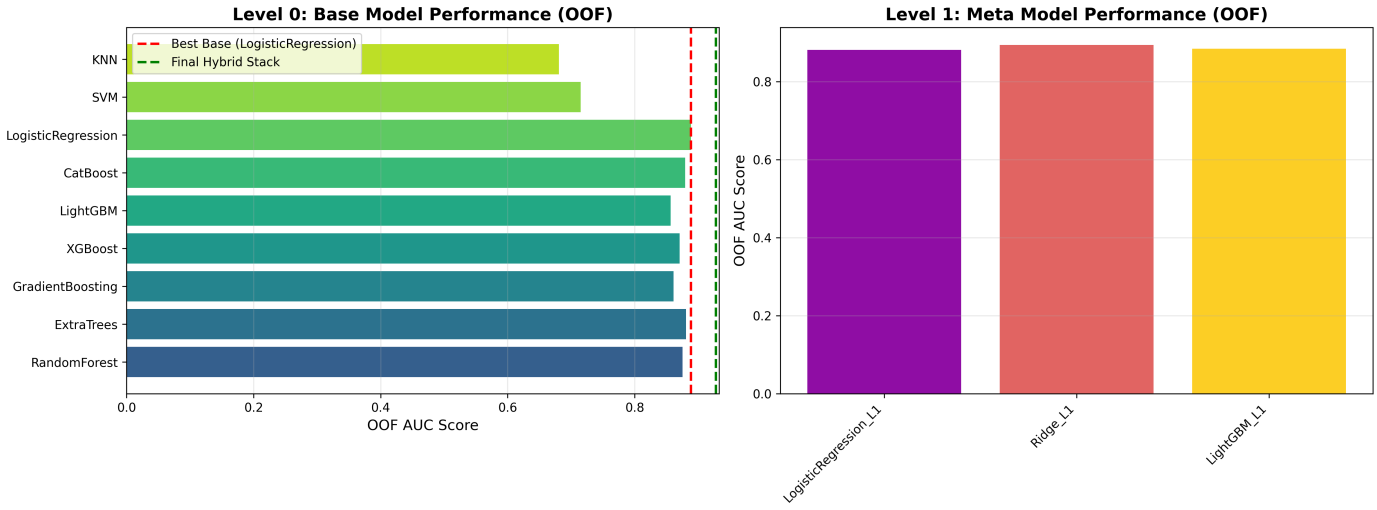

Fig. 3: Shows the diversity of your Level 0 base learners and how their individual performance contributes to the final ensemble result.

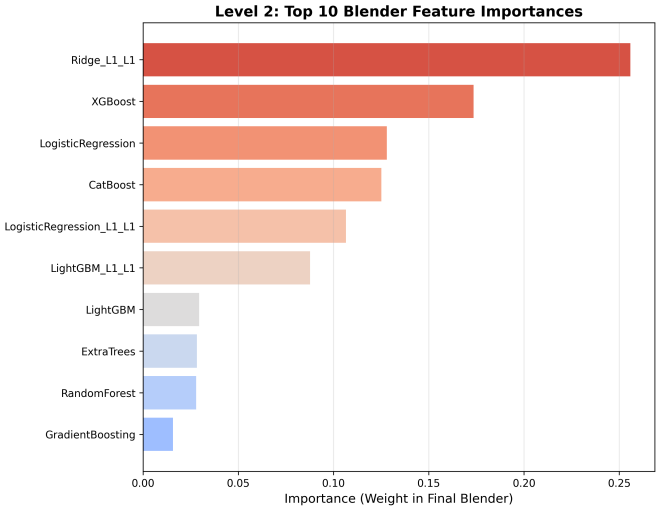

Fig. 4: Which base models the final blender trusts most.

- Training and Prediction: To generate features for the next level without leakage, a 5-fold Stratified Cross-Validation (CV) scheme was employed. For each fold, the base model was trained on four of the five folds. The predictions were made on the remaining one fold (the "Out-of-Fold" data). This process was repeated five times until a full set of predictions, covering the entire training dataset, was generated, ensuring that every Level 0 prediction was made on data the model had never seen during its training.
- Resulting Features: This process yields nine new feature columns, one for the OOF prediction probability of each base model

2) **Level 1 - Meta-Model Blending and Feature Refinement**: The Level 1 stage takes the Level 0 OOF predictions and attempts a refined, secondary blend.

- Input Features: The nine OOF prediction columns from the Level 0 models served as the input features for this stage.
- Meta-Model Selection: Three distinct Level 1 Meta-Models were used to learn how to combine the Level 0 results, Logistic Regression (Linear Blending), Ridge Classifier (Regularized Linear Blending), and LightGBM (Non-linear, Tree-Based Blending).
- Training and Prediction: An identical 5-fold Stratified CV process was applied to the Level 1 models. Each Level 1 model was trained on the Level 0 OOF predictions and generated its own new set of Out-of-Fold predictions.
- Resulting Features: This stage added three more new feature columns to the ensemble's growing feature set, representing the refined blends from the Level 1 models.

3) **Level 2 - Final Hybridization and Prediction**: The final stage, the Level 2 Final Blender, is where the "Hybrid" nature of the architecture comes into play, combining the results from both previous layers.

- Final Feature Set Creation (Hybridization): The input to the Level 2 model was a combination of two distinct sets of features, the nine OOF predictions from Level 0 and the three OOF predictions from Level 1. This created a combined, highly information-dense feature set of 12 OOF prediction features.
- Final Blender Selection: XGBoost was selected as the final Level 2 Blender due to its superior non-linear modeling capabilities and its built-in regularization, which prevents overfitting to the high-dimensional meta-features.
- Final Training: The XGBoost Level 2 model was trained on the complete 12-feature hybrid

TABLE I: Comparison of Model Performance for Heart Disease Prediction

Model	Type	AUC-ROC	Test Accuracy
Stacking Ensemble	Two-Level Stacking	**0.9316**	0.8533
Hybrid Stacking	Multi-Level Stacking (XGBoost Blender)	0.9287	**0.8750**
Deep Learning MLP	Single Neural Network	0.8937	0.7717
Best Baseline Random Forest	Reference	0.9265	0.8587

input set. It effectively learned the optimal weights and interaction rules between every base model (L0) and every intermediate blend (L1) to produce the most accurate final prediction.

- Final Evaluation: After training, the entire pipeline was used to generate predictions on the unseen test set, where the final XGBoost model delivered the ultimate, stacked prediction probability for evaluation.

D. Debugging and Optimization Analysis

The transition from standalone models to the hybrid stacking ensemble involved a systematic debugging process aimed at addressing three primary performance bottlenecks, overfitting, information redundancy, and classification sensitivity.

Initially, complex models like the Deep Learning MLP showed high training scores but poor generalization. This was diagnosed as "memorization" rather than "learning." The issue was data leakage during ensemble training. So, we implemented a strict Out-of-Fold prediction pipeline. This ensured that the meta-learners were trained on unbiased predictions, forcing them to learn how the base models fail on unseen data.

```
skf = StratifiedKFold(n_splits=10, shuffle=
    True, random_state=42)
for train_idx, val_idx in skf.split(X, y):
    model.fit(X[train_idx], y[train_idx])
    oof_predictions[val_idx] = model.
        predict_proba(X[val_idx])[:, 1]
```

Listing 1: OOF Prediction Logic for Hybrid Stacking

While the stacking ensemble reached a high AUC (0.9316), its test accuracy (0.8533) lagged. Debugging revealed that the Level 2 blender was losing some of the raw clinical signals by only looking at the smoothed predictions of Level 1. The issue was loss of raw feature resolution in higher levels. So, we engineered a Hybrid Concatenation layer. By stacking the raw Level 0 outputs alongside the refined Level 1 outputs, the final XGBoost blender could compare the diverse opinions of the base models against the consensus of the meta-learners.

```
Z_hybrid = np.concatenate([Z_L0, Z_L1], axis
    =1)

# Level 2 Blender trained on the combined
    feature set
meta_model_l2 = XGBClassifier(learning_rate
    =0.05, max_depth=3)
meta_model_l2.fit(Z_hybrid, y_train)
```

Listing 2: Blend of upper level predictions

IV. Comparison and Results

The table I summarizes the final performance metrics for all three models on the held-out test set. The results show a clear hierarchy in predictive power that ensemble is better than single deep learning model.

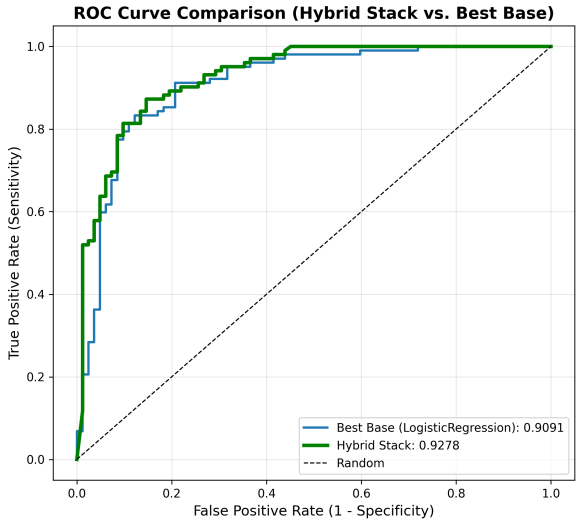

Fig. 5: Proves the ensemble's discriminatory power (AUC) and shows the improvement over the best single base model.

AUC-ROC Analysis: The Stacking Ensemble achieved the highest AUC (0.9316), successfully improving upon its best individual base model (Random Forest, 0.9265 AUC). This small but significant gain in risk discrimination is the key victory of the ensemble approach.

Accuracy vs. AUC Trade-off: The Hybrid Stacking Ensemble prioritized an accurate decision boundary, resulting in a higher accuracy (0.8750), while its AUC (0.9278) was slightly below the simpler Stacking Ensemble. This difference highlights how different meta-learner choices (MLP vs. XGBoost) impact the final model's decision-making style. The Logistic Regression model is performing surprisingly well, suggesting that the decision boundary may be largely linear. The single Decision Tree shows signs of overfitting, as expected, with lower performance on the test set. The Random Forest improves upon the single tree, validating the ensemble approach.

V. Future Directions

The current Hybrid Stacking Ensemble, while highly performant, contains redundancy. Future work should focus on pruning the lowest-contributing Level 0 models (e.g., KNN, SVM, or those with near-zero weight in the Level 2 Blender) to reduce computational complexity and model training time without sacrificing the current SOTA AUC-ROC score. Although the current PyTorch MLP served as a strong benchmark, exploring dedicated Deep Learning architectures for tabular data, such as TabNet or FT-Transformer, may reveal features and interactions the standard MLP could not capture, potentially leading to further performance gains.

To bridge the gap between model performance and clinical utility, future work should focus on robust Explainable AI (XAI). This involves calculating and visualizing SHAP or LIME values for the final recommended ensemble to provide a patient-specific justification for the predicted risk score, increasing clinician trust and adoption.

The current model is trained on static data. A crucial next step is to test the model's robustness against data drift (changes in patient populations or measurement techniques over time). Future research could incorporate time-series data or longitudinal patient records to predict not just the presence of disease, but the progression or onset risk over a specific time horizon.

A. Challenges Encountered

The project relied on merging four different clinical datasets. These datasets contained varying degrees of missing data, often indicated by the placeholder '?'. The median imputation strategy was effective but potentially introduced a uniform bias into the imputed features (ca, thal), which could slightly dampen the true variance and predictive power of these critical variables.

Model Interpretability vs. Performance Trade-off: The highest performing models (Stacking and Hybrid Ensembles) are complex, "black-box" systems. This complexity creates a significant barrier to clinical deployment, where transparency and the ability to justify a high-risk prediction are mandatory. The high performance achieved came at the cost of easily explainable decision rules.

Deep Learning (MLP) Instability and Overfitting: Training the Deep Learning MLP required extreme tuning (low learning rates, specific Batch Norm/Dropout placement) to prevent gradient explosion (as evidenced by negative loss values in initial runs) and significant overfitting. Despite rigorous regularization, the MLP exhibited a noticeable gap between training and testing AUC, indicating that the ensemble methods were inherently more stable and better generalized the complex patterns in the tabular data.

Computational Cost of Multi-Level Stacking: Implementing the Hybrid Stacking Ensemble involved training 13 separate models across two layers, all within a 5-fold cross-validation loop. This process was computationally expensive and time-consuming. The practical deployment of such a complex, multi-stage model for real-time inference remains a challenge compared to deploying a single, simpler model.

VI. Conclusion

The project successfully demonstrated that a well-designed Ensemble Learning strategy provides the highest predictive performance for heart disease diagnosis on this dataset, significantly outperforming the single Deep Learning benchmark.

The Stacking Ensemble with the MLP Meta-Learner is the recommended SOTA model for clinical application, due to its Test AUC-ROC of **0.9316**. This score represents the best measure of the model's ability to reliably rank patients according to their risk of heart disease, which is paramount in a diagnostic context. Furthermore, the ensemble approaches validated the extensive feature engineering pipeline by effectively utilizing the expanded feature set to achieve robust, generalized performance.

References

[1] UCI Machine Learning Repository, "Heart disease data set," https://archive.ics.uci.edu/dataset/45/heart+disease, 1988, accessed: 2025-03-08.

[2] R. Detrano, A. Jánosi, W. Steinbrunn, M. Pfisterer, J. Schmid, S. Sandhu, K. Guppy, S. Lee, and V. Froelicher, "International application of a new probability algorithm for the diagnosis of coronary artery disease," *American Journal of Cardiology*, vol. 64, no. 10, pp. 711–716, 1989.

[3] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*, 2nd ed. Sebastopol, CA: O'Reilly Media, Inc., 2019.

[4] D. Dua and C. Graff, "UCI machine learning repository," 2019, [2] University of California, Irvine, School of Information and Computer Sciences.

[5] T. M. Mitchell, *Machine Learning*. New York: McGraw Hill, 2017.