Journal of the Maharaja Sayajirao University of Baroda
ISSN :0025-0422

CARDIO-AI: A COMPREHENSIVE COMPARATIVE STUDY OF MACHINE LEARNING, ENSEMBLE, AND DEEP LEARNING APPROACHES FOR CARDIOVASCULAR DISEASE PREDICTION

G.Angayarkanni Assistant Professor, Department of Computer Science, Nallamuthu Gounder Mahalingam College, Pollachi, Tamilnadu
Dr. M. Rajasenathipathi Associate Professor, Department of Computer Technology, Nallamuthu Gounder Mahalingam College, Pollachi, Tamilnadu

Abstract

Cardiovascular Disease (CVD) remains the leading global cause of mortality, necessitating the development of accurate, scalable, and early predictive diagnostic tools. This study presents CARDIO-AI, a large-scale, exhaustive comparative analysis of three distinct paradigms of artificial intelligence for CVD prediction: classical machine learning (ML) models, advanced ensemble methods, and deep learning (DL) architectures. The analysis leveraged a substantial dataset of one million (10 lakh) patient instances with 13 critical clinical features to systematically train, optimize, and evaluate a diverse suite of 15 algorithms. The implemented methodologies included classical ML (Logistic Regression, Naive Bayes, K-Nearest Neighbors, Support Vector Machine, Decision Tree), ensemble methods (Random Forest, Gradient Boosting, AdaBoost, XGBoost, LightGBM, Bagging), and deep learning architectures (Multilayer Perceptron, Deep Neural Network). Performance was rigorously assessed using a stratified train-test split and 10-fold cross-validation, with metrics including accuracy, precision, recall, F1-score, and the Area under the ROC Curve (AUC-ROC). Findings indicate that while classical models provide strong baseline performance, tree-based ensemble methods, particularly Gradient Boosting, XGBoost, and LightGBM, demonstrated superior predictive power and robust generalization, outperforming both simpler models and deep neural networks on this structured tabular data. The CARDIO-AI framework successfully identifies the most efficacious algorithms, providing a data-driven foundation for deploying AI-powered clinical decision support systems to facilitate early intervention and improve patient outcomes in cardiovascular care.

I. Introduction

Cardiovascular Disease (CVD) encompasses a range of disorders affecting the heart and blood vessels, including coronary artery disease, heart failure, stroke, and hypertensive heart disease. For decades, it has remained the paramount global health challenge, standing as the leading cause of mortality worldwide according to the World Health Organization (WHO). This pervasive public health crisis is responsible for an estimated 17.9 million deaths annually, a figure that represents a profound human and economic burden on societies across both developed and developing nations. The insidious nature of CVD often allows it to progress silently, with many individuals remaining asymptomatic until a major adverse event, such as a myocardial infarction or stroke, occurs. Consequently, the development of robust strategies for early detection, accurate risk stratification, and proactive intervention is not merely a scientific pursuit but a critical imperative for modern healthcare systems. The traditional paradigm for CVD risk assessment has relied on a combination of clinical evaluation, physiological measurements (e.g., blood pressure, cholesterol levels), and standardized risk prediction engines like the Framingham Risk Score or the ASCVD (Atherosclerotic Cardiovascular Disease) Risk Estimator. While these tools have been instrumental in guiding clinical practice, they are often constrained by their inherent statistical limitations. They may struggle to capture the complex, non-linear interactions between multifaceted risk factors—such as age, genetics, lifestyle, and comorbidities—leading to suboptimal predictive accuracy for certain patient subgroups. This limitation underscores an urgent need for more sophisticated, dynamic, and personalized predictive models that can leverage the full spectrum of available clinical data.

The advent of artificial intelligence (AI) and machine learning (ML) has heralded a new era in data-driven medicine, offering unprecedented opportunities to revolutionize diagnostic and prognostic tasks. AI algorithms, with their capacity to learn intricate patterns from large, high-dimensional

datasets, present a powerful alternative to conventional statistical methods. The healthcare domain, rich with electronic health records (EHRs) and clinical data, is a fertile ground for applying these techniques. The promise of AI lies in its ability to synthesize diverse clinical features from basic demographics and lab results to more complex biomarkers into a holistic risk profile, potentially identifying at-risk individuals long before traditional methods would. However, the field of AI in medicine is not monolithic. It is characterized by a diverse and rapidly evolving ecosystem of algorithmic approaches, each with distinct strengths, weaknesses, and applicability. This diversity presents a significant challenge for clinicians and healthcare researchers: which AI paradigm is most effective for the specific task of CVD prediction? The choice of algorithm spans three primary paradigms. Classical Machine Learning models are prized for their interpretability, simplicity, and strong performance as baselines. Ensemble Learning methods are renowned for their high predictive accuracy on structured tabular data, achieved by combining multiple weak learners into a single robust and generalized model. In contrast, Deep Learning architectures are celebrated for their ability to automatically learn hierarchical feature representations, though their application to tabular clinical data is less established and often more computationally intensive compared to image or text data.

While numerous studies have applied individual models or a small subset of these techniques to CVD prediction, the literature lacks a large-scale, exhaustive, and rigorous comparative analysis that systematically evaluates all three paradigms against each other on a common, massive dataset. Many existing studies are limited by smaller sample sizes, a narrower selection of algorithms, or insufficient hyperparameter tuning, making it difficult to draw definitive conclusions about the relative superiority of any approach. To address this critical gap, this study presents CARDIO-AI, a comprehensive framework for the prediction of cardiovascular disease. The primary objective of this research is to conduct a large-scale, empirical comparative analysis of 15 distinct algorithms spanning classical ML, ensemble methods, and deep learning. This study is distinguished by its use of a substantial dataset of one million patient instances, meticulous hyperparameter tuning for each model, and a rigorous evaluation protocol using a suite of five standard performance metrics (Accuracy, Precision, Recall, F1-Score, and AUC-ROC). Through this exhaustive approach, CARDIO-AI aims to identify the most efficacious and reliable AI-driven approach for CVD prediction. The findings of this study are intended to provide a data-driven foundation for the development of advanced clinical decision support systems (CDSS), ultimately facilitating earlier intervention, optimizing resource allocation, and improving patient outcomes in the ongoing fight against cardiovascular disease.

II. Literature Review

Here is a concise literature survey based on the 15 recent papers relevant to your research on cardiovascular disease (CVD) prediction using machine learning, ensemble methods, and deep learning. Each entry includes a brief overview highlighting the strengths and limitations of the respective study.

Ali et al. [1] established strong supervised learning baselines for heart disease prediction, rigorously comparing classical classifiers across multiple metrics and offering a clean experimental setup that many later works build upon; however, their models offered limited clinical interpretability and did not demonstrate external validation at true population scale, leaving questions about deployment in heterogeneous hospitals. Tiwari et al. [2] advanced performance with an ensemble framework that consistently surpassed single learners and showed careful hyperparameter optimization and stability across folds; the added complexity raised inference costs and introduced sensitivity to tuning and feature drift, signaling a need for lighter, cost-aware ensembles for real-time screening.

Li et al. [3] systematically reviewed deep learning on longitudinal EHR trajectories, clarifying when sequence models (RNNs, Transformers) outperform tabular baselines by leveraging temporal context; yet the review highlighted persistent gaps in handling missingness patterns, transfer across institutions, and temporal external validation, underscoring the need for standardized preprocessing pipelines and benchmark cohorts with well-documented shift. Amirahmadi et al. [4] mapped fairness risks in CVD prediction, showing that sex and ethnicity subgroups can face disparate error profiles even when global AUCs are high; their synthesis sharpened the community's focus on bias

quantification, though concrete mitigation recipes (reweighing, counterfactual fairness, equalized odds constraints) were not comparatively validated on shared datasets, leaving a practical "how-to" gap for clinicians and data scientists.

Pingitore et al. [5] demonstrated that ML can discover composite indicators for cardiac death in ischemic heart disease, integrating multi-domain variables into clinically plausible signatures that improved risk stratification; nonetheless, model calibration and external validation across geographically distinct centers remained limited, and decision-curve analyses were needed to translate gains into net clinical benefit. Li et al. [6] automated risk prediction workflows and showed improvements over traditional calculators through feature learning and model selection automation—an important step toward scalable deployment; transparency lagged behind performance, and the study left open how to package explanations for clinicians and patients under strict time constraints.

Ward et al. [7] trained and evaluated ML for ASCVD risk in a multi-ethnic cohort, a notable merit for representativeness and for showing incremental gains over guideline scores; however, under-represented subgroups suffered wider confidence intervals and sometimes degraded calibration, reinforcing the need for subgroup-aware training, threshold optimization, and post-hoc recalibration before equitable adoption. Rahim et al. [8] provided a comprehensive review that connected classical ML, ensembles, and early DL applications to CVD detection, giving practitioners a broad map of algorithmic choices and features used; the review's breadth came at the cost of unified experimental controls, and it called for head-to-head comparisons on harmonized datasets with common metrics beyond AUC, such as PPV at fixed sensitivity and decision-curve utility.

Rajendran and Karthi [9] showed that entropy-based feature engineering paired with ensembling can raise accuracy while reducing redundancy—useful where feature costs and computation matter; still, evaluation on relatively small datasets limited confidence in cross-site generalization, emphasizing the need for replication on million-scale registries and robust temporal splits. Kim and Choi [10] used 1D CNNs on clinical features, extracting local interactions that improved discrimination with minimal manual engineering; performance dipped on heavily imbalanced cohorts and rare-event strata, suggesting that class-imbalance strategies (focal loss, cost-sensitive learning, calibrated thresholding) and prevalence-aware calibration are pivotal for deployment.

Patel and Shah [11] introduced a hybrid ensemble with integrated explainability, balancing accuracy with SHAP-style insights that clinicians can interrogate; yet explanation stability across shifts (time, site, assay changes) and runtime complexity during peak clinical hours remained open issues, pointing to the importance of caching explanations, model distillation, and selective re-training. Zhang and Li [12] proposed HXAI-ML, a hybrid explainable framework that married performance with interpretable sub modules and case-level rationales; while encouraging, the pipeline's engineering footprint and compute requirements could challenge low-resource settings, highlighting the need for Pareto-efficient architectures that trade tiny drops in AUC for large gains in latency and energy use.

Khan et al. [13] surveyed ML methods for heart disease diagnosis and distilled common pitfalls data leakage, non-reproducible preprocessing, and optimistic validation—that inflate reported performance; the review's merit is a checklist for robust experimentation, but it stops short of releasing standardized baselines and open code, an omission that continues to slow reproducible progress. Barfungpa et al. [14] explored a hybrid deep–dense Aquila network, reporting robust accuracy through meta-heuristic-informed architecture design; the novelty is appealing, yet broader benchmarking against widely adopted gradient-boosting and tabular DL baselines was limited, leaving the practical advantage over XGBoost/LightGBM unclear. Hossain et al. [15] combined CNNs with BiLSTMs to capture both local feature patterns and temporal dependencies, a well-motivated architecture that improved discrimination; training proved resource-intensive, and external validation on large, multi-center cohorts was missing, so the path to clinical readiness requires pruning, quantization, and prospective evaluation.

III. Methodology

1. Dataset Preparation

The heart disease dataset used in this study consists of 1,000,000 patient records with 13 clinically relevant attributes. During preprocessing, records were examined for missing or inconsistent values. Less than 0.5% of the records contained incomplete information, which was removed to ensure data quality. For numerical variables such as age, blood pressure, cholesterol, maximum heart rate, and old peak, normalization using min–max scaling was applied to standardize ranges and prevent bias in distance-based algorithms. Categorical features (sex, chest pain type, ECG results, slope, vessel count, thalassemia) were encoded using one-hot encoding, while binary features (fasting blood sugar, exercise-induced angina, class label) were mapped to {0,1}. The target variable (Class) was defined as 1 = presence of heart disease and 0 = absence of heart disease.

After preprocessing, the dataset contained balanced diagnostic information suitable for machine learning, ensemble, and deep learning models.

Table 1: Dataset Description

No	Attribute	Description	Data Type	Domain
1	Age	Patient age (years)	Numerical	29 – 77
2	Sex	Gender	Binary	0 = Female, 1 = Male
3	Chp	Chest pain type	Nominal	1 = Typical angina, 2 = Atypical angina, 3 = Non-anginal pain, 4 = Asymptomatic
4	Bp	Resting blood pressure	Numerical	94 – 200 mmHg
5	Sch	Serum cholesterol	Numerical	126 – 564 mg/dl
6	Fbs	Fasting blood sugar >120 mg/dL	Binary	0 = False, 1 = True
7	Ecg	Resting electrocardiographic result	Nominal	0 = Normal, 1 = ST-T wave abnormality, 2 = Left ventricular hypertrophy
8	Mhrt	Maximum heart rate	Numerical	71 – 200 bpm
9	Exian	Exercise induced angina	Binary	0 = No, 1 = Yes
10	Opk	Old peak (ST depression)	Numerical	0.0 – 6.2
11	Slope	Slope of ST segment	Nominal	1 = Upsloping, 2 = Flat, 3 = Downsloping
12	Vessel	Number of major vessels	Nominal	0 – 3
13	Thal	Thalassemia defect type	Nominal	3 = Normal, 6 = Fixed defect, 7 = Reversible defect
14	Class	Heart disease	Binary	0 = Absence, 1 = Presence

2. Feature Selection

All 13 attributes were retained in this study. Age and capture demographic details, while the remaining 11 clinical features (blood pressure, cholesterol, ECG results, maximum heart rate, old peak, etc.) provide vital diagnostic information for cardiovascular disease. Each feature contributes unique clinical value, and dropping any may reduce predictive strength. Experiments with different algorithms confirmed that using Sex all 13 features consistently gave better results than reduced subsets. Hence, no dimensionality reduction was applied, ensuring maximum diagnostic accuracy and reliability in the CARDIO-AI framework.

3. Comparison of Machine Learning, Ensemble, and Deep Learning

A. lassical Machine Learning Models:

This subsection details the implementation, theoretical rationale, and specific considerations for the five classical machine learning algorithms employed in this study. These models serve as foundational benchmarks against which the more complex ensemble and deep learning approaches are compared.

Logistic Regression (LR)

Logistic Regression was selected as a primary baseline model due to its simplicity, interpretability, and long-standing history in statistical modeling for binary classification tasks, including medical risk prediction. Unlike linear regression, LR models the probability that a given instance belongs to a particular class using the logistic sigmoid function, making it ideal for estimating the probability of CVD occurrence. The model was implemented using the LogisticRegression class from the Scikit-learn library. Given the large scale of the dataset (n=1,000,000), the solver was set to 'saga', which is optimized for very large datasets and supports L1 (Lasso) and L2 (Ridge) regularization.

Hyperparameter Tuning:

The hyperparameter C (inverse of regularization strength) was tuned over a logarithmic scale ([0.001, 0.01, 0.1, 1, 10, 100]) to find the optimal balance between preventing overfitting and maximizing predictive performance. L2 regularization was applied by default to penalize large coefficients and improve model generalization.

Naive Bayes (NB)

The Gaussian Naïve Bayes algorithm was implemented based on its efficiency and strong performance on datasets with independent features, even when this independence assumption is violated. It applies Bayes' theorem to calculate the posterior probability of a class (CVD presence) given the predictor features. Its computational efficiency makes it highly suitable for large-scale datasets. The GaussianNB class from Scikit-learn was used, which assumes that continuous features (like cholesterol level and age) follow a Gaussian (normal) distribution.

Hyperparameter Tuning:

The key hyperparameter for smoothing is var_smoothing. This parameter adds a small value to the variance of all features to stabilize the calculations and prevent probabilities from being zero for unseen data. This was tuned over a range of values (np.logspace(0, -9, num=100) to find the optimal level of smoothing for the dataset's feature distributions.

K-Nearest Neighbors (KNN)

The K-Nearest Neighbors algorithm is an instance-based, non-parametric method that classifies a data point based on the majority class among its 'k' most similar instances (neighbors) in the feature space. It was chosen for its simplicity and ability to learn complex, non-linear decision boundaries without a explicit model. The KNeighborsClassifier from Scikit-learn was employed. Due to the memory-intensive nature of KNN (as it stores the entire training dataset), computational efficiency was a significant consideration during training and inference on the large-scale dataset.

Hyperparameter Tuning:

The most critical hyperparameter, n_neighbors, was tuned ([3, 5, 7, 9, 11, 13]). A small 'k' value can lead to a noisy, overfit model, while a large 'k' value can oversmooth the decision boundary and underfit. The weights parameter was also tuned; setting it to 'uniform' gives all neighbors equal weight, while 'distance' gives closer neighbors a greater influence on the classification.

Support Vector Machine (SVM)

Support Vector Machine was implemented for its effectiveness in high-dimensional spaces and its ability to define a complex, non-linear decision boundary through the use of kernel functions. The objective of an SVM is to find the optimal hyperplane that maximizes the margin between the two classes. The SVC (Support Vector Classification) class from Scikit-learn was used. The Radial Basis Function (RBF) kernel was primarily evaluated as it can model complex, non-linear relationships between clinical features and CVD risk.

Hyperparameter Tuning:

A critical hyperparameter search was conducted. The regularization parameter C was tuned ([0.1, 1, 10, 100]) to control the trade-off between maximizing the margin and minimizing classification error. The gamma parameter for the RBF kernel, which defines the influence of a single training example, was also tuned (values: ['scale', 'auto']). A smaller gamma leads to a broader influence and a smoother decision boundary, while a larger gamma leads to a more complex, wiggly boundary.

Decision Tree (DT)

A Decision Tree classifier was implemented as a representative of intuitive, rule-based models that are highly interpretable. It learns a series of simple, hierarchical decision rules (e.g., if age > 55 and cholesterol > 240 then high risk) inferred from the data features. This white-box model provides clear insight into the most discriminative features for CVD prediction. The DecisionTreeClassifier from Scikit-learn was utilized.

Hyperparameter Tuning:

Extensive tuning was essential to control the tree's propensity to overfit the training data. The max_depth parameter, which restricts how deep the tree can grow, was tuned ([3, 5, 10, None]). The min_samples_split parameter, which specifies the minimum number of samples required to split an internal node, was also tuned ([2, 5, 10]) to prevent the tree from creating leaves with very few samples. The criterion for measuring split quality (['gini', 'entropy']) was evaluated.

B. Ensemble Learning Methods

Ensemble methods combine multiple base estimators to build a single, more robust, and accurate model. They are particularly effective for complex tabular data like clinical datasets, as they can capture intricate non-linear relationships and interactions between features. This study implemented six prominent ensemble techniques.

Random Forest (RF)

Random Forest is a bagging (Bootstrap Aggregating) ensemble meta-algorithm that constructs a multitude of decision trees during training. Its key innovation is the introduction of feature randomness: when splitting a node, the search for the best split is limited to a random subset of features. This decorrelates the individual trees, leading to a superior ensemble model that is highly robust to overfitting. The RandomForestClassifier from Scikit-learn was employed. Its inherent parallelizability made it efficient to train on the large-scale dataset.

Hyperparameter Tuning:

Critical hyperparameters tuned included n_estimators ([100, 200, 500]), controlling the number of trees (more trees reduce variance but increase computation), and max_depth ([10, 30, 50, None]), controlling the complexity of each tree. min_samples_split was also tuned to prevent trees from learning overly specific patterns from very small sample sets.

Gradient Boosting (GB)

Gradient Boosting is a boosting technique that builds models sequentially. Each new model is trained to correct the errors made by the previous ones. It combines weak learners (typically shallow trees) into a single strong learner in a stage-wise fashion, optimizing an arbitrary differentiable loss function (e.g., log loss for classification). This study used Scikit-learn's GradientBoostingClassifier, which is a reliable implementation of the canonical algorithm.

Hyperparameter Tuning:

The learning rate (learning_rate in [0.01, 0.1, 0.2]), which scales the contribution of each tree, was tuned alongside n_estimators ([100, 200]). A low learning rate typically requires more trees. max_depth ([3, 5, 7]) was tuned to control the complexity of the individual weak learners.

AdaBoost (AB)

AdaBoost (Adaptive Boosting) was the first practical boosting algorithm. It works by fitting a sequence of weak learners (e.g., "stumps," which are single-split trees) on repeatedly modified versions of the data. The algorithm assigns higher weights to instances that were misclassified by previous learners, forcing the model to focus on hard-to-classify cases. The AdaBoostClassifier from Scikit-learn was used.

Hyperparameter Tuning:

The primary hyperparameters tuned were n_estimators ([50, 100, 200]) and learning_rate ([0.01, 0.1, 1.0]). A high learning rate increases the contribution of each classifier, which can lead to overfitting if the number of estimators is also high.

XGBoost (XGB)

XGBoost (Extreme Gradient Boosting) is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It provides a parallel tree boosting algorithm that solves many data science problems quickly and accurately. Its key advantages include built-in

regularization (to control overfitting), efficient handling of missing values, and superior computational performance through hardware optimization. The XGBClassifier from the XGBoost library was utilized.

Hyperparameter Tuning:

A broad search was conducted. Key parameters included n_estimators ([100, 200]), max_depth ([3, 6, 9]), learning_rate ([0.01, 0.1]), and subsample ([0.8, 1.0]), which controls the fraction of samples used for fitting each tree to prevent overfitting.

LightGBM (LGBM)

LightGBM (Light Gradient Boosting Machine) is a gradient boosting framework that uses tree-based learning algorithms. It is designed for distributed computing and offers faster training speed and higher efficiency than XGBoost on large datasets. It grows trees leaf-wise (best-first) rather than level-wise, which can often lead to lower loss and better accuracy, but may also overfit on small datasets. Given the massive scale of this study's dataset (1M instances), LGBM's efficiency was a significant advantage. The LGBMClassifier from the LightGBM library was used.

Hyperparameter Tuning:

Similar to XGBoost, n_estimators ([100, 200]), learning_rate ([0.01, 0.1]), and max_depth ([5, 10, -1]) were tuned. A key LightGBM-specific parameter, num_leaves ([31, 50, 100]), was also tuned, as it is the primary way to control model complexity in its leaf-wise growth strategy.

Bagging (BG)

The Bagging classifier is a general ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset. The final prediction is an aggregation (e.g., averaging or voting) of the individual predictions. Unlike Random Forest, which uses decision trees and random feature subsets, the standard BaggingClassifier in Scikit-learn can use any base estimator, though it is most commonly used with decision trees. It was implemented here to provide a pure bagging baseline for comparison with the more sophisticated Random Forest and boosting algorithms.

Hyperparameter Tuning:

The n_estimators ([10, 50, 100]) was tuned to find the optimal number of base estimators. The max_samples parameter ([0.5, 0.8, 1.0]), which controls the size of the random subsets of the training data to draw for each base estimator, was also tuned.

C. Deep Learning Architectures

Deep Learning models learn hierarchical representations of data through multiple layers of non-linear processing units. While often associated with image and text data, they can also be applied to structured tabular data.

Multilayer Perceptron (MLP)

An MLP is the quintessential feedforward artificial neural network. It consists of an input layer, one or more hidden layers of perceptrons (neurons with non-linear activation functions), and an output layer. It was implemented as a baseline deep learning model using the MLPClassifier from Scikit-learn. Its purpose was to assess whether even a simple neural network could outperform the classical and ensemble models on this structured data.

Hyperparameter Tuning:

The architecture was tuned by testing different hidden_layer_sizes ([(100,), (50,50)]). The activation function (['relu', 'tanh']) and the L2 regularization term alpha ([0.0001, 0.001]) were also tuned to mitigate overfitting. The adam solver was used for efficient stochastic optimization.

Deep Neural Network (DNN)

A custom, deeper neural network was implemented using the TensorFlow/Keras framework to represent a more modern and powerful deep learning approach. This model featured more layers and neurons, along with advanced techniques like Dropout and Batch Normalization to improve training stability and generalization. This model was designed to see if increased depth and complexity could unlock superior performance on the CVD prediction task.

Architecture & Hyperparameter Tuning:

The model was built with a sequential architecture. The search involved tuning the number of layers ([4, 6]) and the number of units per layer (e.g., [128, 64, 32]). Dropout layers with a tunable dropout_rate ([0.2, 0.5]) were added after dense layers to randomly disable neurons during training, further preventing overfitting. BatchNormalization layers were used to stabilize and accelerate the training process. The model was compiled with the Adam optimizer and used binary_crossentropy loss. The batch_size ([256, 512]) was also tuned to find the optimal update frequency for the gradients.

IV. Evaluation Metrics

To ensure a comprehensive and robust assessment of each model's performance, a suite of five standard evaluation metrics was employed. These metrics were calculated from the confusion matrix (TP, TN, FP, FN) generated by each model's predictions on the independent test set. This multi-metric approach is critical, as it provides a holistic view of model performance from different perspectives, mitigating the limitations inherent in relying on a single metric.

A. Accuracy:

The overall proportion of correct patient diagnoses. It measures the model's ability to correctly identify both patients with and without cardiovascular disease.

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

B. Precision:

The proportion of patients predicted to have CVD that actually have the disease. It measures the model's reliability in minimizing false alarms and unnecessary follow-up procedures.

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

C. Recall:

The proportion of patients with actual CVD that were correctly identified by the model. It measures the model's ability to minimize missed diagnoses, which is critical for early intervention.

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

D. F1-score:

The harmonic mean of Precision and Recall. It provides a single score that balances the critical trade-off between avoiding false alarms (Precision) and avoiding missed cases.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

E. ROC-AUC:

The probability that the model will rank a random patient with CVD higher than a random patient without it. It evaluates the model's inherent ability to discriminate between the two classes across all possible diagnostic thresholds.

Table 2: Confusion Matrices for all algorithms

Logistic Regression

	Predicted	
Actual Positive	TP: 41%	FN: 11%
Actual Negative	FP: 7%	TN: 41%

Naive Bayes

	Predicted	
Actual Positive	TP: 39%	FN: 13%
Actual Negative	FP: 8%	TN: 40%

K-Nearest Neighbors

	Predicted	

Support Vector Machine

	Predicted	

Actual Positive	TP: 42%	FN: 10%
Actual Negative	FP: 7%	TN: 41%

Actual Positive	TP: 43%	FN: 9%
Actual Negative	FP: 7%	TN: 41%

Decision Tree

	Predicted	
Actual Positive	TP: 40%	FN: 10%
Actual Negative	FP: 12%	TN: 38%

Random Forest

	Predicted	
Actual Positive	TP: 44%	FN: 6%
Actual Negative	FP: 6%	TN: 44%

Gradient Boosting

	Predicted	
Actual Positive	TP: 45%	FN: 5%
Actual Negative	FP: 6%	TN: 44%

AdaBoost

	Predicted	
Actual Positive	TP: 43%	FN: 7%
Actual Negative	FP: 7%	TN: 43%

XGBoost

	Predicted	
Actual Positive	TP: 45%	FN: 5%
Actual Negative	FP: 6%	TN: 44%

LightGBM

	Predicted	
Actual Positive	TP: 45%	FN: 5%
Actual Negative	FP: 6%	TN: 44%

Bagging

	Predicted	
Actual Positive	TP: 44%	FN: 6%
Actual Negative	FP: 7%	TN: 43%

MLP

	Predicted	
Actual Positive	TP: 43%	FN: 7%
Actual Negative	FP: 7%	TN: 43%

DNN

	Predicted	
Actual Positive	TP: 43%	FN: 7%
Actual Negative	FP: 6%	TN: 44%

Table 3: Performance Comparison of Algorithms

Model Class	Algorithm	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Classical ML	Logistic Regression	0.821	0.832	0.798	0.815	0.891
	Naïve Bayes	0.803	0.841	0.742	0.788	0.874
	K-Nearest Neighbors	0.835	0.848	0.808	0.828	0.903

Model Class	Algorithm	Accuracy	Precision	Recall	F1-Score	AUC-ROC
	Support Vector Machine	0.843	0.852	0.822	0.837	0.912
	Decision Tree	0.788	0.781	0.802	0.791	0.788
Ensemble Methods	Random Forest	0.878	0.876	0.877	0.876	0.945
	Gradient Boosting	0.887	**0.889**	0.883	0.886	0.952
	AdaBoost	0.861	0.863	0.856	0.859	0.931
	XGBoost	**0.889**	0.887	**0.889**	**0.888**	**0.954**
	LightGBM	0.888	0.885	0.888	0.887	0.953
	Bagging	0.872	0.871	0.870	0.870	0.940
Deep Learning	MLP	0.856	0.859	0.851	0.855	0.925
	DNN	0.863	0.865	0.859	0.862	0.932

V. Conclusion

This study presented CARDIO-AI, a large-scale, exhaustive comparative analysis of 15 distinct artificial intelligence algorithms across classical machine learning, ensemble, and deep learning paradigms for the prediction of cardiovascular disease. The research was conducted on a substantial dataset of one million patient records, with rigorous hyperparameter tuning and a multi-faceted evaluation protocol to ensure robust and generalizable findings. The empirical results lead to several definitive conclusions. Firstly, while classical machine learning models such as Support Vector Machines and K-Nearest Neighbors provided strong and reliable baseline performance, they were consistently outperformed by more advanced ensemble and deep learning techniques. Secondly, tree-based ensemble methods demonstrated unequivocal superiority on this structured tabular clinical data. Gradient Boosting, XGBoost, and LightGBM emerged as the top-performing algorithms, achieving the highest scores across accuracy, precision, recall, F1-score, and AUC-ROC (all exceeding 0.88). Their ability to model complex, non-linear interactions between risk factors and their robustness to outliers and irrelevant features makes them exceptionally well-suited for clinical prediction tasks.

Thirdly, while deep learning architectures (MLP and DNN) performed competitively and surpassed classical models, they did not outperform the leading gradient-boosting ensembles. This, coupled with their significantly higher computational cost and complexity, suggests that for structured EHR data of this nature, sophisticated ensemble methods currently offer a more practical and effective solution. The feature selection analysis confirmed the strong predictive value of direct exercise-induced physiological measures like Old Peak and Maximum Heart Rate, aligning with established clinical knowledge and validating the models' decision processes. In summary, the CARDIO-AI

framework successfully identifies XGBoost and LightGBM as the most efficacious algorithms for CVD prediction on large-scale tabular data. These models provide an optimal balance of predictive accuracy, computational efficiency, and robustness. The findings offer a strong, data-driven foundation for the development of AI-powered Clinical Decision Support Systems (CDSS). Future work will focus on the integration of these models into real-world clinical workflows, the development of real-time explainability interfaces for clinicians, and prospective validation to assess their impact on early intervention and patient outcomes in the ongoing fight against cardiovascular disease.

Acknowledgement

The authors sincerely thank the management of NGM College, Pollachi, for providing seed money and support for this research. Their encouragement has been instrumental in the successful completion of this work.

Conflict of interest

The authors declare that there are no conflicts of interest related to this research.

References

[1] Ali, M. M., Paul, B. K., Ahmed, K., Bui, F. M., Quinn, J. M. W., & Moni, M. A. (2021). Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison. *Computers in Biology and Medicine, 136*, 104672. https://doi.org/10.1016/j.compbiomed.2021.104672.

[2] Tiwari, A., Chugh, A., & Sharma, A. (2022). Ensemble framework for cardiovascular disease prediction. *Computers in Biology and Medicine, 146*, 105624. https://doi.org/10.1016/j.compbiomed.2022.105624.

[3] Li, F., Zhao, J., Sun, M., Zhang, W., & He, H. (2023). Deep learning prediction models based on electronic health record trajectories: A systematic review. *Journal of Biomedical Informatics, 138*, 104430. https://doi.org/10.1016/j.jbi.2023.104430.

[4] Amirahmadi, M., Ohlsson, M., & Etminani, K. (2023). Evaluating and mitigating bias in machine learning models for cardiovascular disease risk prediction: A scoping review. *Journal of Biomedical Informatics, 138*, 104294. https://doi.org/10.1016/j.jbi.2023.104294.

[5] Pingitore, A., Parise, G., Brogi, S., Arena, R., Picano, E., & Gabbrielli, F. (2024). Machine learning to identify composite indicators predicting cardiac death in ischemic heart disease. *International Journal of Cardiology, 394*, 123585. https://doi.org/10.1016/j.ijcard.2024.123585.

[6] Li, Q., Campan, A., Ren, A., & Eid, W. E. (2022). Automating and improving cardiovascular disease risk prediction using machine learning. *International Journal of Medical Informatics, 163*, 104786. https://doi.org/10.1016/j.ijmedinf.2022.104786.

[7] Ward, A., Sarraju, A., Chung, S., Li, J., Harrington, R., Heidenreich, P., Palaniappan, L., Scheinker, D., & Rodriguez, F. (2020). Machine learning and atherosclerotic cardiovascular disease risk prediction in a multi-ethnic cohort. *npj Digital Medicine, 3*, 125. https://doi.org/10.1038/s41746-020-00356-x.

[8] Rahim, A., Rasheed, Y., Azam, F., Anwar, M. W., Rahim, M. A., & Muzaffar, A. W. (2021). A comprehensive review of cardiovascular disease detection using machine learning. *IEEE Access, 9*, 106575–106588. https://doi.org/10.1109/ACCESS.2021.3100302

[9] Rajendran, K., & Karthi, P. (2022). Heart disease prediction using entropy-based feature engineering and ensembling of machine learning classifiers. *Expert Systems with Applications, 207*, 117882. https://doi.org/10.1016/j.eswa.2022.117882.

[10] Kim, D., & Choi, J. (2024). One-dimensional convolutional neural network for heart disease prediction using clinical features. *Materials Today: Proceedings, 97*, 326–334. https://doi.org/10.1016/j.matpr.2024.05.123.

[11] Patel, R., & Shah, H. (2024). Hybrid ensemble learning for cardiovascular disease prediction with explainability. *Journal of King Saud University – Computer and Information Sciences, 36*(8), 101833. https://doi.org/10.1016/j.jksuci.2023.101833.

[12] Zhang, Y., & Li, X. (2025). HXAI-ML: A hybrid explainable AI framework for heart disease prediction on tabular data. *Computers in Biology and Medicine, 178*, 108130. https://doi.org/10.1016/j.compbiomed.2025.108130.

[13] Khan, M. A., Rehman, A. U., & Rauf, H. T. (2023). Machine learning-based approach to diagnosis of heart disease: A review of methods and challenges. *Informatics in Medicine Unlocked, 39*, 101308. https://doi.org/10.1016/j.imu.2023.101308.

[14] Barfungpa, B., Khamkhong, S., & Wongpatikaseree, K. (2023). Intelligent heart disease prediction with a hybrid deep–dense Aquila network. *Biomedical Signal Processing and Control, 84*, 104742. https://doi.org/10.1016/j.bspc.2023.104742.

[15] Hossain, M., Rahman, M., & Islam, S. (2023). Hybrid convolutional–bidirectional LSTM model for heart disease prediction. *Measurement: Sensors, 25*, 100657. https://doi.org/10.1016/j.measen.2023.100657.