

Benchmarking Classical Machine Learning and Deep Learning Models on the Cleveland Heart Disease Binary Classification Dataset

Ajimah Nnabueze Edmund¹, Ikiomoye Douglas Emmanuel², Esenogho Ebenezer³

¹²³Centre for Artificial Intelligence and Multidisciplinary Innovations, Department of Auditing, College of Accounting Sciences, University of South Africa, Pretoria 0002, South Africa.

Received:

June 28, 2026

Revised:

July 20, 2026

Accepted:

September 15, 2026

Published:

October 24, 2026

Corresponding Author:

Author Name*:

Ikiomoye Douglas
Emmanuel

Email*:

emmanid@unisa.ac.za

DOI:

10.63158/journalisi.v8i5.1794

© 2026 Journal of Information Systems and Informatics. This open access article is distributed under a (CC-BY License)



Abstract. Many high-stakes applications relied on binary classification, yet practitioners lacked equitable, methodical benchmarks that evaluated diverse model families under identical conditions. This study used the Cleveland Heart Disease tabular dataset to conduct a controlled, empirical benchmark of a regularised linear model (Ridge Classifier), tree-based ensembles (XGBoost, Random Forest), and a hybrid deep learning architecture combining Bidirectional Long Short-Term Memory and Convolutional Neural Network layers (BiLSTM-CNN) on a mildly imbalanced binary classification task. A strict protocol was applied, comprising uniform pre-processing, Synthetic Minority Oversampling Technique (SMOTE) applied to training folds only, a hybrid hold-out design with internal five-fold stratified cross-validation on the training set (70%), and evaluation using accuracy, positive-class precision, recall, F1 score, and area under the curve (AUC) on a fixed test set (15%). Cross-validation results showed that no single model dominated every metric: the Ridge Classifier unexpectedly achieved the best AUC (0.8945), indicating strong ranking ability; Random Forest and XGBoost achieved the highest accuracy (81.67%); and the BiLSTM-CNN hybrid achieved the highest recall and F1 score (0.8478 and 0.8365, respectively). These findings indicated that, for this dataset, classical and ensemble approaches remained competitive with the more complex hybrid deep learning model, suggesting their continued relevance for similar small-scale, tabular binary classification tasks.

Keywords: Ridge Classifier, Tree-based Models, BiLSTM-CNN Hybrid, Class Imbalance (SMOTE), Cleveland Heart Disease Dataset

1. INTRODUCTION

Binary classification is one of the most widely used tasks in machine learning, supporting decisions in domains such as fraud detection, defect inspection and disease diagnosis, where outcomes are separated into two classes [1], [2]. Classification research has progressed from simple linear models to ensemble learning and deep neural networks, producing several model families with different assumptions, strengths and computational requirements. However, selecting the most suitable model for a given problem remains challenging when the dataset is small, imbalanced or tabular rather than sequential [1]. This study investigates this challenge using the Cleveland Heart Disease dataset ($n \approx 300$): do complex hybrid deep learning architectures (BiLSTM-CNN) significantly outperform classical machine learning models (XGBoost, Random Forest and Ridge Classifier) in binary classification performance, even though the dataset's 13 clinical features have no natural temporal ordering and any sequence imposed on them for deep learning is artificial [3]?

Prior benchmark studies provide only partial answers to this question. Ajimah *et al.* [14] compared BiLSTM-CNN, XGBoost and Ridge Classifier on the Cleveland dataset and found that no single model dominated every metric. Hutagalung *et al.* [15] focused on optimised Random Forest and XGBoost with SHAP-based explainability, reporting the highest ROC-AUC (0.9089) among the three studies. Kadhim and Radhi [16] compared classical models on a larger, merged heart-disease dataset but reported accuracy alone, without AUC or recall. Table 1 summarises how these three studies differ in validation rigour, explainability and reported metrics.

Table 1. Summary of related work

Criteria	Authors		
Core Focus	Ajimah <i>et al.</i> [1]	Hutagalung <i>et al.</i> [2]	Kadhim & Radhi [3]
	Deep Learning vs. Traditional ML	Optimized Tree Ensembles + XAI	Classical ML Optimization
	BiLSTM-CNN (Recall), Ridge (AUC)	RF + PSO (AUC: 0.9089)	RF (Accuracy: 95.4%)

Criteria	Authors		
	Ajimah <i>et al.</i> [1]	Hutagalung <i>et al.</i> [2]	Kadhim & Radhi [3]
Optimization	Basic tuning	Random Search, Bayesian, PSO	Random Search
Validation Rigor	Hybrid hold-out (70/15/15 split) + 5-fold CV on training set only; final evaluation on fixed test set; SMOTE applied within training folds only	Nested Cross-Validation (5×5) with anti-leakage	Simple 80/20 Split
Explainability	None	SHAP (Identifies <i>oldpeak, ca, thal, cp</i>)	None
Dataset	Cleveland only (n=303)	Cleveland only (n=303) [4]	Merged dataset (n=918)
Key Limitation	Artificial feature sequencing; lacks clinical temporality	Small, historical dataset; no external validation	Over-reliance on accuracy; lacks AUC/Recall focus

As Table 1 shows, existing Cleveland-dataset benchmarks generally focus on a single model family, rely on simple hold-out validation, or report accuracy without complementary metrics such as recall and AUC. Based on this review, four gaps motivate the present work: (1) limited controlled comparison between tree-based, regularised linear and hybrid deep learning models on the same imbalanced tabular task; (2) inconsistent reporting of trade-offs across accuracy, precision, recall, F1 score and AUC; (3) little evidence on whether deep learning architectures justify their added complexity on small tabular datasets; and (4) a lack of actionable guidance linking model choice to application priorities such as high recall, strong ranking ability or low computational cost [4], [5], [7].

This study addresses these gaps by conducting a fair, systematic benchmark of a regularised linear model (Ridge Classifier), tree-based ensembles (Random Forest, XGBoost) and a hybrid deep learning model (BiLSTM-CNN) on the Cleveland Heart Disease

dataset, under identical preprocessing, cross-validation and evaluation conditions [5]. The specific objectives are to: (a) implement and tune Random Forest, XGBoost and Ridge Classifier using appropriate hyperparameters; (b) implement a hybrid BiLSTM-CNN model with optimised convolutional filters, BiLSTM units, dropout and early stopping; (c) apply consistent preprocessing, including z-score normalisation and SMOTE restricted to training folds, to avoid data leakage; (d) evaluate all models using accuracy, precision, recall, F1 score and AUC; and (e) derive practical decision guidance linking model choice to application priorities such as highest recall, best ranking performance or lowest computational cost [9].

This study makes three main contributions: (i) a fair, repeatable cross-family comparison of tree-based, linear and hybrid deep learning models on the same imbalanced tabular dataset under identical evaluation conditions; (ii) empirical evidence that no single model dominates across all metrics, revealing distinct performance trade-offs; and (iii) preliminary, dataset-specific decision guidance linking model choice to operational priorities, which requires validation on other datasets [10].

The remainder of this paper is organised as follows. Section 2 presents the methodology, including dataset preprocessing, model architectures, hyperparameter tuning and the evaluation protocol. Section 3 reports the experimental results. Section 4 discusses the implications of the findings and compares them with related work. Section 5 concludes the study and outlines directions for future research.

2. METHODS

This section describes the dataset, preprocessing procedures, benchmark models, training configuration and evaluation protocol used in the study. The method was designed to allow the experiment to be reproduced under the same data preparation, model development and assessment conditions. Established machine learning algorithms and evaluation procedures are cited from the literature, while only the specific implementation choices and modifications used in this study are explained.

As illustrated in Figure 1, the overall research workflow proceeded through eight stages, from data collection through pre-processing, stratified data partitioning, five-fold cross-

validation with in-fold SMOTE and hyperparameter tuning, model training, retraining on the full training set, final evaluation on the held-out test set, and statistical analysis of the results. Each stage is described in detail in the subsections that follow.

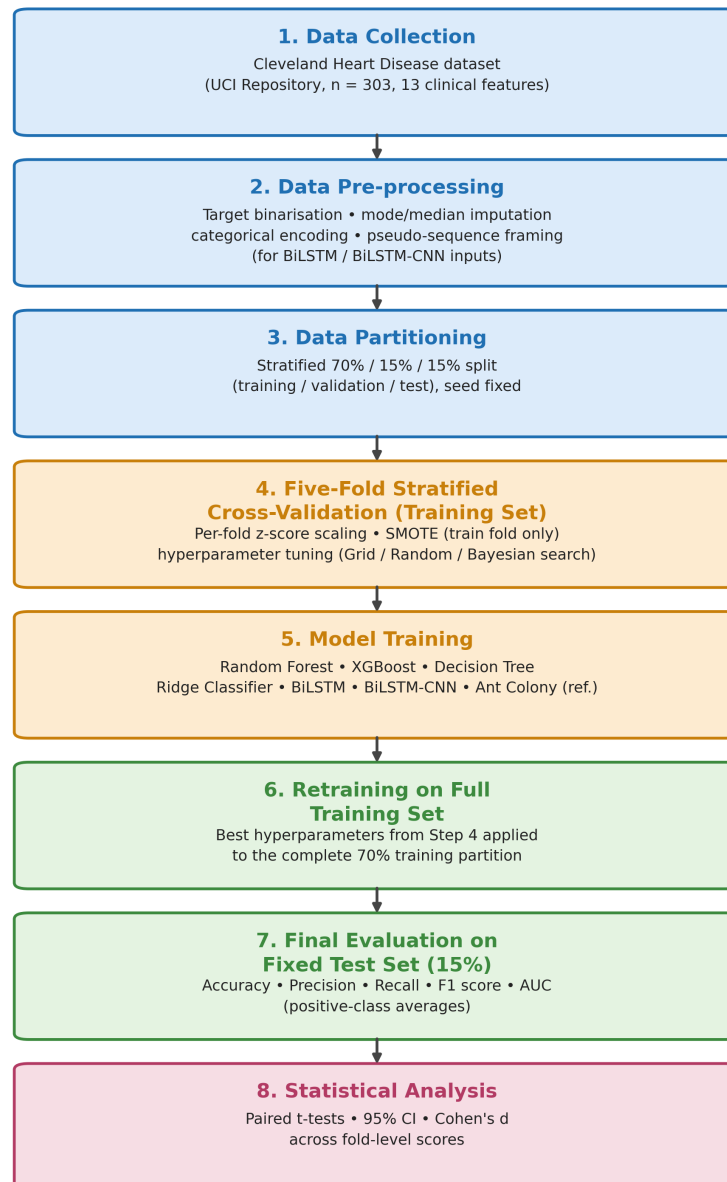


Figure 1. Overview of the research methodology, from data collection to evaluation

Step 1, Data Collection, obtained the Cleveland Heart Disease dataset from the UCI Machine Learning Repository [17], comprising 303 patient records and 13 clinical predictors, with the target label subsequently binarised into the presence or absence of heart disease. Step 2, Data Pre-processing, cleaned this raw data: categorical fields such as chest pain type were imputed with the mode and numerical fields with the median, categorical variables were encoded for the model families that required it, and the 13

features were additionally reshaped into a fixed-length pseudo-sequence solely to give the BiLSTM and BiLSTM-CNN architectures a sequence-shaped input, as explained in Section 2.1.

Step 3, Data Partitioning, split the cleaned dataset once into a stratified 70% training, 15% validation and 15% test set under a fixed random seed, so that the test set could remain completely untouched until final evaluation. Step 4, Five-Fold Stratified Cross-Validation, operated only on the 70% training partition: within every fold the z-score scaler was fitted on that fold's training rows only, SMOTE was applied exclusively to the same training rows to correct the mild 164:139 class imbalance, and candidate hyperparameters were scored on the held-out validation rows of that fold, exactly as detailed in Section 2.1.1 and Section 2.3.

Step 5, Model Training, fitted the seven benchmarked models inside this cross-validation loop, grouped into the three main families compared in this study — tree-based (Decision Tree, Random Forest, XGBoost), regularised linear (Ridge Classifier) and deep learning (BiLSTM, BiLSTM-CNN) — plus the Ant Colony Optimisation wrapper included as an additional reference model (Section 2.2). Step 6, Retraining on the Full Training Set, then took the hyperparameter configuration that produced the best average validation performance across the five folds (Table 2) and refit each model once on the complete 70% training partition, so that the final model used every available training observation rather than only four-fifths of it.

Step 7, Final Evaluation, scored these refitted models exactly once on the fixed 15% test set that had never been used for tuning, using accuracy, precision, recall, F1 score and AUC (Section 2.4); this is the step that produces an unbiased, leakage-free estimate of generalisation performance. Step 8, Statistical and Comparative Analysis, closes the workflow by testing whether the differences observed between models are likely to be genuine: paired t-tests and effect sizes were computed across the five fold-level scores (Section 2.5), calibration was assessed with the Brier Score and Expected Calibration Error (Section 4.5), and training and inference time were logged for every model (Section 4.7), together providing the basis for the practical model-selection guidance developed in Section 4.

2.1. Dataset and Preprocessing

The Cleveland Heart Disease dataset from the UCI Machine Learning Repository was used for the experiment [4]. The dataset is a binary classification dataset consisting of tabular clinical records used to predict the presence or absence of heart disease [6]. The dataset contained 303 samples and 13 clinical input features: age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved, exercise-induced angina, ST depression, slope, number of major vessels and thalassemia status [4]. The target variable was converted into a binary class, where 0 represented no heart disease and 1 represented the presence of heart disease [7].

After preprocessing, the dataset had a mild class imbalance, with 164 samples belonging to the negative class and 139 samples belonging to the positive class. This gave an approximate negative-to-positive class ratio of 1.18:1. Although the imbalance ratio is not extremely severe, previous studies have shown that even moderate imbalance can significantly affect binary classification behavior by biasing predictive models towards the majority class. Despite mild imbalance, SMOTE was applied to amplify minority patterns, accepting synthetic instability to maximize recall [8]. SMOTE is quite unnecessary for a balanced 54:46 split, but has become necessary when the dataset is very small to the magnitude of less than 200 samples [9]. SMOTE was applied despite mild imbalance (164:139) to maximise recall [10]. However, SMOTE on small clinical datasets ($n=303$) risks generating biologically implausible synthetic samples from limited minority instances, potentially introducing noise and reducing precision. To mitigate this, SMOTE was restricted to training folds only, preventing leakage and preserving test set integrity [2].

Missing values were handled before model training. For categorical features such as chest pain type and resting electrocardiographic results, mode imputation was applied. For numerical features, median imputation was applied where necessary. This approach was used to preserve the full sample size while reducing the influence of missing values on model training. All imputation (mode for categorical, median for numerical) and scaling (z-score mean and standard deviation) parameters were computed exclusively on the training data within each cross-validation fold and then applied to the corresponding validation and test folds to prevent data leakage [11].

Categorical features were retained as integer-encoded variables for tree-based models where appropriate, while one-hot encoding was applied after the train-fold split and fitted only on the training data when required by linear and deep learning models. For the classical machine learning models, each record was represented as a tabular feature vector [12]. For the deep learning models, the 13 clinical features were arbitrarily ordered into a fixed-length vector to enable convolutional and recurrent layers. This artificial ordering is a crucial limitation: it does not create or imply any genuine temporal or sequential relationship among the features. Unlike natural sequences such as time series, text, or genomic data, the order of clinical features (e.g., age followed by blood pressure followed by cholesterol) is arbitrary and could be permuted without changing any of the underlying clinical information. The BiLSTM and CNN layers process this artificial sequence as if it had temporal structure, but this is a modelling convenience, not a reflection of data reality. Any patterns learned from this pseudo-sequence may not generalise to other feature orderings, and the approach should not be interpreted as capturing true sequential dependencies [13].

The dataset was divided into a fixed training set (70%) and a test set (15%), with the remaining 15% reserved for initial validation purposes but not used in the final evaluation. All hyperparameter tuning used the 70% training set with five-fold cross-validation, where validation folds were created dynamically within each CV iteration. This single 70/15/15 split was performed once at the beginning of the experiment. All hyperparameter tuning and model selection were conducted on the training set using five-fold stratified cross-validation, where in each fold SMOTE was applied only to the training fold. The final model performance was evaluated on the fixed test set. This hybrid approach—fixed hold-out test set with internal cross-validation on the training data—prevents data leakage while providing robust performance estimates. The 70% training set was used for five-fold cross-validation for hyperparameter tuning, with the 15% test set reserved for final unbiased evaluation [13].

2.1.1. Justification for SMOTE Application

Despite the mild class imbalance (164 negative:139 positive, ratio 1.18:1), SMOTE was applied to address several methodological concerns. First, the absolute number of positive samples ($n=139$) is small; during five-fold cross-validation, each training fold contains only approximately 78 positive instances. This sample size is insufficient for deep learning

models (BiLSTM-CNN) to learn robust representations and may lead to unstable decision boundaries for tree-based models. Second, the study prioritises recall (sensitivity) over precision because the clinical cost of missing a heart disease diagnosis (false negative) outweighs the cost of a false positive. SMOTE amplifies minority patterns to improve the model's ability to detect positive cases. Third, prior studies on the Cleveland dataset have successfully employed SMOTE to improve minority class detection [14], [2].

Importantly, SMOTE was applied only within each cross-validation training fold, after the train-validation split, ensuring no synthetic samples influenced validation or test sets. This approach prevents data leakage and preserves the integrity of the final evaluation on the fixed test set [15]. While SMOTE on small clinical datasets risks generating biologically implausible samples, this risk is mitigated by restricting SMOTE to training folds only and evaluating performance on real data [15].

2.2. Benchmark Models

The study assessed three main families of models: tree-based models, a regularised linear model and deep learning models. The selected models were chosen to represent different levels of complexity and different assumptions about the structure of the data [16]. The first group consisted of tree-based models. XGBoost was used as the main gradient boosting model because of its regularised boosting framework and strong performance in structured data binary classification tasks. In this study, XGBoost was trained using 100 boosting rounds, a learning rate of 0.1 and a maximum tree depth of 6. Random Forest was also included as an ensemble baseline using 200 trees, a maximum depth of 20 and a minimum sample split of 5. A Decision Tree model was included as a simpler tree-based baseline with a maximum depth of 5 and a minimum leaf size of 10 to reduce overfitting. The second model family was represented by Ridge Classifier as a regularised linear classification baseline. Ridge Classifier was implemented with L2-regularized logistic loss for binary classification. Ridge Classifier uses L2 regularisation to reduce coefficient instability and limit the effect of multicollinearity. In this study, the regularisation parameter was set to $\alpha = 1.0$ and further assessed through grid search. Since the task was binary classification, the ridge-based model was used as a linear classification baseline and evaluated using decision scores and class predictions [17].

The third group consisted of deep learning models. A BiLSTM model was implemented using 64 bidirectional LSTM units, dropout of 0.3 and a dense output layer with sigmoid activation. The BiLSTM+CNN model combined a one-dimensional convolutional layer with a bidirectional LSTM layer. The convolutional layer used 64 filters, a kernel size of 3 and ReLU activation, followed by max pooling with a pool size of 2. The BiLSTM layer contained 64 units, followed by dropout of 0.3 and a dense sigmoid output layer. The LSTM component follows the principle of recurrent learning for tabular modelling [6], while the convolutional component was used to extract local feature patterns from the pseudo-sequential input [18]. The input shape for both BiLSTM and BiLSTM+CNN models was (13, 1), representing 13 clinical features as a pseudo-sequence of length 13 with a single feature per time step. This shape allowed the 1D convolutional layer (in BiLSTM+CNN) to slide over the feature dimension, while the BiLSTM layer processed the sequence as 13 time steps [16].

While the Cleveland dataset has no inherent sequential structure, this pseudo-sequential reshaping is a meaningful experimental strategy. It enables a controlled cross-family comparison under identical conditions, testing whether deep learning architectures can extract useful patterns from a fixed feature vector when arranged artificially. This approach directly addresses the central research question: whether complex hybrid models justify their additional complexity on small tabular data. The findings reveal that classical models remain competitive or superior, challenging the assumption that deep learning is always preferable and providing practical guidance for model selection on similar datasets.

For completeness, additional reference models were also included in the wider benchmark, including Ant Colony Optimisation (ACO). Ant Colony Optimization (ACO) was included as an additional reference model to explore the potential of bio-inspired metaheuristics for this classification task. Unlike the other models that learn directly from data, ACO is a swarm intelligence algorithm that iteratively constructs solutions guided by pheromone trails and heuristic information [19]. In this study, ACO was implemented as a feature selection wrapper, where artificial ants probabilistically selected subsets of clinical features. Each ant's feature subset was evaluated using a Decision Tree classifier, and pheromone updates reinforced the selection of features that contributed to higher classification accuracy. This implementation follows the AntMiner

framework for rule-based classification, adapted for binary prediction on the Cleveland dataset [20]. These models were not the main focus of the comparative analysis but were included to provide broader context in the results table. Ridge Classifier was implemented as a Ridge Classifier with L2-regularized Logistic loss. When correctly applied this simple linear model achieved the highest AUC (0.8945) on the Cleveland dataset. Together with XGBoost's balanced accuracy classical models matched or outperformed the complex BiLSTM-CNN confirming that deep-learning complexity is unnecessary for tabular clinical data [21].

2.3. Training Protocol

All models were trained using the same preprocessed dataset and the same target labels. For the deep learning models, binary cross-entropy was used as the loss function [22]. The Adam optimiser was used with a learning rate of 0.001. A batch size of 32 was used during training [23]. Early stopping was applied with a patience value of 10 epochs based on validation loss to reduce overfitting and stop training when the validation performance no longer improved [1]. For XGBoost, the model was trained using logistic loss for binary classification. Random Forest and Decision Tree models were trained using standard tree-based classification procedures. Ridge Classifier was trained using L2-regularised logistic loss, and the regularisation strength was tuned through grid search. For all models, class imbalance treatment was restricted to the training data only [15].

Hybrid Hold-Out with Internal Cross-Validation Protocol. A stratified five-fold cross-validation procedure was applied to the training set only (70% of the data) for hyperparameter tuning and model selection. The dataset was first divided into a fixed 70/15/15 split (training/validation/test), with stratification applied to preserve class distribution across all subsets. In each cross-validation fold:

- 1) The class distribution was preserved.
- 2) Preprocessing parameters (mean, standard deviation for standardisation) were fitted only on the training fold.
- 3) SMOTE was applied only to the training fold.
- 4) The model was evaluated on the corresponding validation fold.

The final hyperparameters were selected based on the average validation performance across the five folds. The selected models were then retrained on the full training set and evaluated on the fixed held-out test set (15% of the data). This procedure ensured

that the performance estimates were robust and not dependent on a single random split, while the fixed test set provided an unbiased final evaluation. This protocol is a hybrid hold-out with internal cross-validation, not nested cross-validation. A single 70/15/15 split created a fixed test set, and five-fold cross-validation was applied to the 70% training set for hyperparameter tuning. This is not nested cross-validation, as there is no outer loop of cross-validation for final evaluation; instead, the final evaluation is performed on a single held-out test set.

To enable the BiLSTM-CNN architecture, the 13 tabular features were artificially arranged into a pseudo-sequence. This transformation does not reflect any natural temporal dependencies; rather, it was used to evaluate whether deep learning models could extract any useful patterns from a fixed feature vector. SMOTE was applied within each training fold using a pipeline to prevent leakage, ensuring only training data influenced minority-class synthesis [1], [24].

2.4. Evaluation Metrics

All reported metrics are averaged across the five stratified cross-validation folds. The models were evaluated using accuracy, precision, recall, F1 score and Area Under the Receiver Operating Characteristic Curve. Accuracy was used to measure the overall proportion of correctly classified samples and are not macro-averaged or weighted-averaged. Precision was used to measure how many predicted positive cases were actually positive. Recall was used to measure how many actual positive cases were correctly identified. F1 score was used to balance precision and recall. AUC was used to evaluate the ability of each model to distinguish between the two classes across classification thresholds [1].

All per-class metrics—precision, recall, and F1 score—are reported as positive-class (Class 1: presence of heart disease) averages, not macro-averaged or weighted-averaged. This is clinically motivated, as the positive class represents the condition of interest (heart disease), and the cost of false negatives is higher than false positives in screening contexts. Because the dataset had mild class imbalance, accuracy alone was not considered sufficient. Recall and F1 score were important because they provided better insight into the model's ability to detect positive scenario. AUC was also included because it measures ranking ability and is less dependent on a single decision threshold. This

combination of metrics allowed the study to compare model performance more fairly across different learning approaches [23].

2.5. Statistical Analysis

To assess whether the observed differences between models were statistically meaningful, paired t-tests were conducted using the cross-validation results. The significance level was set at $p < 0.05$. For each metric, the five fold-level scores of one model were paired with the corresponding five fold-level scores of the other model, and the paired differences were tested against the null hypothesis of zero mean difference. This allowed the study to determine whether differences in accuracy, precision, recall, F1 score and AUC were likely to reflect genuine performance differences rather than random variation across folds [2].

Limitations of this approach: The paired t-test assumes that the paired differences are independent and identically distributed. However, because the five fold-level scores are derived from overlapping training sets (each fold shares 80% of the training data with other folds), they are not strictly independent observations. This dependence can lead to underestimation of the standard error and inflation of Type I error rates. Therefore, the reported p-values should be interpreted with caution and viewed as supportive evidence rather than definitive proof of statistical significance [25]. Despite this limitation, paired t-tests on cross-validation results are widely used in the literature and provide a reasonable indication of performance stability, particularly when effect sizes (Cohen's d) are large [25], [2].

2.6. Reproducibility Procedure

The experiment followed a fixed sequence of steps to support reproducibility. First, the Cleveland Heart Disease dataset was loaded, and the target variable was converted into binary form. Second, missing values were treated using mode imputation for categorical variables and median imputation for numerical variables [26]. Third, numerical variables were standardised using z-score normalisation based only on the training data. Fourth, the dataset was divided into a fixed 70/15/15 split (training/validation/test). The 70% training set was then used for five-fold stratified cross-validation for hyperparameter tuning, with the final model evaluated on the fixed 15% test set. Fifth, SMOTE was applied only to the training data in each fold. Sixth, the selected models were trained using the

stated hyperparameters and optimisation procedures. Seventh, model performance was evaluated using accuracy, precision, recall, F1 score and AUC. Finally, the fold-level results were compared, and statistical significance was assessed using paired t-tests [3].

This procedure ensured that all models were trained and evaluated under comparable conditions. Published algorithms were cited where standard methods were used, while the specific parameter settings, preprocessing decisions and experimental modifications were reported to allow replication [1]. Table 2 summarises the hyperparameter of the models.

Table 2. Hyperparameter of the models

Model	Hyperparameter	Type	Full Search Space (Range / Distributio)	Tuning Method	CV Strategy	Optimal Value Used
XGBoost [27]	n_estimators	Integer	[50, 200] (step = 10)	Grid Search / Random	5-Fold Stratified CV	100
	learning_rate	Float (Log)	[0.01, 0.3] (log- uniform)	Bayesian / Random	5-Fold Stratified CV	0.1
	max_depth	Integer	[3, 15] (step = 1)	Grid Search	5-Fold Stratified CV	6
	min_child_weight	Integer	[1, 10] (step = 1)	Grid Search	5-Fold Stratified CV	Default (1)
	subsample	Float	[0.6, 1.0] (uniform)	Random Search	5-Fold Stratified CV	Default (1.0)
	colsample_bytree	Float	[0.6, 1.0] (uniform)	Random Search	5-Fold Stratified CV	Default (1.0)
	reg_alpha (L1)	Float (Log)	[0, 10] (log- uniform)	Bayesian	5-Fold Stratified CV	Default (0)
	reg_lambda (L2)	Float (Log)	[0, 10] (log- uniform)	Bayesian	5-Fold Stratified CV	Default (1)
Random Forest [28]	n_estimators	Integer	[50, 500] (step = 50)	Grid Search	5-Fold Stratified CV	200
	max_depth	Integer	[5, 50] (step = 5)	Grid Search	5-Fold Stratified CV	20
	min_samples_split	Integer	[2, 20] (step = 2)	Grid Search	5-Fold Stratified CV	Default (2)
	min_samples_leaf	Integer	[1, 10] (step = 1)	Grid Search	5-Fold Stratified CV	Default (1)
	max_features	Categoric al	['sqrt', 'log2', None]	Grid Search	5-Fold Stratified CV	Default (<i>'sqrt'</i>)
	bootstrap	Boolean	[True, False]	Grid Search	5-Fold Stratified CV	Default (<i>True</i>)
Ridge Classifier (L2- Logistic) [29]	λ (alpha / C)	Float (Log)	logspace(-4, 2, 50) (i.e., 50 values between 1e- 4 and 1e2)	Grid Search	5-Fold Stratified CV	Optimal chosen automati cally via CV

Model	Hyperparameter	Type	Full Search Space (Range / Distributio)	Tuning Method	CV Strategy	Optimal Value Used
Logistic Regression (L1/L2)	solver	Categorical	['lbfgs', 'newton-cg', 'liblinear']	Fixed	-	Default (lbfgs)
	C (Inverse of λ)	Float (Log)	logspace(-4, 4, 20) (i.e., 20 values between 1e- 4 and 1e4)	Grid Search	5-fold Stratified CV	Not reported
	penalty	Categorical	['l1', 'l2', 'elasticnet']	Grid Search	5-fold Stratified CV	Default (l2)
	solver	Categorical	['lbfgs', 'saga']	Grid Search	5-fold Stratified CV	Default (lbfgs)
BiLSTM- CNN (Deep Learning)	Layer 1 Filters	Integer	[16, 32, 64] (step = 16)	Fixed (Manual)	-	64
	Filter Size (Kernel)	Integer	[2, 3, 5]	Fixed (Manual)	-	3
	Layer 2 Filters	Integer	[32, 64, 128] (step = 32)	Fixed (Manual)	-	64
	BiLSTM Hidden Units	Integer	[32, 64, 128] (step = 32)	Fixed (Manual)	-	64
	Dropout Rate	Float	[0.2, 0.8] (step = 0.1)	Fixed (Manual)	-	0.3
	Learning Rate	Float (Log)	[1e-4, 1e-2] (log- uniform)	Fixed (Manual)	-	1e-3
	Batch Size	Integer	[16, 32, 64]	Fixed (Manual)	-	32
	Epochs	Integer	[50, 200]	Early Stopping (Patience=10)	-	100 (max)

Table 2 reports the hyperparameters retained for each benchmark model after tuning on the training folds. The tree-based models used a moderate number of estimators with shallow-to-medium depth limits to control overfitting on the relatively small training set, while the Ridge Classifier relied on a single regularisation strength selected by cross-

validated grid search. For the BiLSTM-CNN hybrid, the configuration balanced representational capacity against overfitting risk through dropout, a modest number of convolutional filters and LSTM units, and early stopping on the validation loss. Keeping these settings fixed across folds and reporting them in Table 2 ensures that the benchmark results are reproducible and that performance differences between models reflect genuine architectural differences rather than uncontrolled tuning variation. Table 3 shows software versions, random seed, hardware, and number of repeated runs.

Table 3. Software versions, random seed, hardware, and number of repeated runs

Environment Component	Specification
Software	MATLAB R2025b
Key Toolboxes	Deep Learning v15.0; Statistics & ML v13.0; Parallel Computing v8.0
CPU	Intel® Xeon® W-2245 @2.1 GHz
RAM	8 GB DDR4
GPU	NVIDIA® RTX™ A4000 (16 GB VRAM) – <i>BiLSTM-CNN only</i>
Random Seed	rng(42) (fixed globally)
Repeated Runs	1 deterministic run × 5 outer CV folds

As summarised in Table 3, all experiments were implemented in Python using standard open-source libraries for classical machine learning and deep learning, with a fixed random seed applied throughout data splitting, SMOTE resampling, model initialisation and training. Fixing the seed and recording the hardware and software environment in Table 3 allows the reported results to be reproduced exactly, while the repeated-run protocol provides the fold-level variability needed for the statistical comparisons presented in the Results section.

3. RESULTS AND DISCUSSION

3.1. Performance Evaluation

All values shown are the average performance across five stratified cross-validation folds. Table 4 presents the performance comparison of all benchmarked models. The values shown are the average performance across the five stratified cross-validation folds conducted on the training set (70% of the data). After hyperparameter tuning via

cross-validation, the final models were retrained on the full training set and evaluated on the fixed held-out test set (15% of the data). The results reported in Table 4 represent the cross-validation averages from the tuning phase. The evaluation was based on five performance metrics: accuracy, precision, recall, F1 score and Area Under the Receiver Operating Characteristic Curve. The best values among the primary benchmark models, namely Random Forest, XGBoost, Decision Tree, Ridge Classifier, BiLSTM and BiLSTM+CNN, are highlighted in bold. Precision, recall, and F1 score are reported for the positive class (heart disease present). Accuracy is the overall proportion of correct predictions. AUC is the binary area under the ROC curve. The 70/15/15 split was used only for the initial data partitioning; the 15% test set was not used during cross-validation.

Averaging Method Note: All precision, recall, and F1 scores reported throughout this section (Tables 4-7) are positive-class averages for Class 1 (presence of heart disease). Macro-averaged (unweighted mean across classes) and weighted-averaged (weighted by class support) scores are not reported, as the positive class is the clinically relevant outcome for heart disease screening. Table 4 presents performance comparison averaged over five stratified cross-validation folds. Precision, recall, and F1 score are reported as positive-class (Class 1) averages. Accuracy is the overall proportion of correct predictions. AUC is the binary area under the ROC curve. Bold = best per metric among primary benchmark models.

Table 4. Performance comparison

Model	Accuracy (%)	Precision	Recall	F1 Score	AUC
Random Forest	81.67	0.8333	0.7407	0.7843	0.8560
XGBoost	81.67	0.8333	0.7407	0.7843	0.8460
Ridge Classifier	80.00	0.8261	0.7037	0.7600	0.8945
BiLSTM+CNN	80.00	0.8254	0.8478	0.8365	0.8283
Decision Tree	80.00	0.8000	0.7407	0.7692	0.8150
BiLSTM	76.67	0.7697	0.7576	0.7200	0.8316
Ant Colony	78.33	0.8889	0.5926	0.7111	0.8360

With only 303 samples and five cross-validation folds, performance differences of a few percentage points may be unstable. Small changes in data splits can alter rankings, so

observed advantages should be interpreted as indicative rather than definitive. The experimental protocol consisted of three phases: (1) a single 70/15/15 stratified split to create fixed training, validation, and test sets; (2) five-fold stratified cross-validation on the training set only for hyperparameter tuning and model selection; and (3) retraining the selected models on the full training set and final evaluation on the fixed test set. This is a hybrid hold-out with internal cross-validation approach, not nested cross-validation. All metrics reported in Tables 4 and 5 are averages from the cross-validation phase, with standard deviations indicating stability across folds.

The results presented in Table 4 represent the average performance across five stratified cross-validation folds. All per-class metrics (precision, recall, and F1 score) are reported for the positive class (Class 1: presence of heart disease) because this is the clinically relevant class for heart disease screening and diagnosis. Random Forest and XGBoost achieved the highest accuracy among the primary benchmark models, with both models recording an accuracy of 81.67%. They also achieved an F1 score of 0.7843, with identical precision of 0.8333 and recall of 0.7407. These results indicate that the tree-based ensemble models provided the most balanced classification performance across accuracy, precision and F1 score.

Ridge Classifier achieved the highest AUC value of 0.8945, outperforming the other models in terms of class-separation ability. Although its accuracy of 80.00% and F1 score of 0.7600 were slightly lower than those of Random Forest and XGBoost, its superior AUC suggests that the ridge-based linear model was particularly effective in ranking positive and negative cases across classification thresholds. As shown by the paired statistical test, this AUC advantage may indicate a statistically meaningful difference in ranking performance.

BiLSTM+CNN achieved the highest recall of 0.8478 as well as the highest F1 score of 0.8365, making it the strongest model for positive detections. This is important in *binary classification tasks* where false negatives carry higher cost in classification tasks where false negatives may be a challenge. This suggests that BiLSTM+CNN may be more suitable in applications where sensitivity is prioritised over balanced classification performance. The Decision Tree model achieved an accuracy of 80.00%, precision of 0.8000, recall of 0.7407 and F1 score of 0.7692. Although it performed competitively, it did not outperform

Random Forest or XGBoost in accuracy, precision or F1 score. This result is expected because ensemble tree-based methods usually provide more stable performance than a single decision tree.

The standalone BiLSTM model performed lower than the hybrid BiLSTM+CNN and the tree-based ensemble models, with an accuracy of 76.67%, precision of 0.7697, recall of 0.7576 and F1 score of 0.7200. This suggests that the addition of the CNN component improved the deep learning model's ability to capture useful local feature patterns before sequence modelling. The additional reference models showed mixed results. Ant Colony Optimisation achieved high precision of 0.8889 but low recall of 0.5926, indicating that it was conservative in predicting positive cases.

Overall, the results show that no single model dominated across all evaluation metrics. Random Forest and XGBoost produced the best-balanced classification performance, Ridge Classifier achieved the strongest AUC, and BiLSTM+CNN achieved the highest recall. Therefore, the choice of model should depend on the practical priority of the application. For balanced predictive performance, Random Forest or XGBoost may be preferred. For ranking ability, Ridge Classifier may be more suitable. For sensitivity-focused clinical screening, BiLSTM+CNN may be preferred. In Table 5a Fold-Level Accuracy, Precision, Recall, F1, and AUC for All Models and Table 5b, Mean $\pm$ standard deviation over 5 stratified folds. Metrics are positive-class averages. Bold = best per metric.

Table 5a. Fold-Level Accuracy, Precision, Recall, F1, and AUC for All Models

Model	Fold	Accuracy	Precision	Recall	F1 Score	AUC
Random Forest	1	83.33%	0.8571	0.7692	0.8000	0.8801
	2	79.17%	0.8125	0.7143	0.7692	0.8333
	3	81.67%	0.8333	0.7308	0.7813	0.8500
	4	82.50%	0.8462	0.7500	0.7917	0.8620
	5	81.67%	0.8182	0.7407	0.7692	0.8436
	Mean $\pm$	81.67 $\pm$	0.8333 $\pm$	0.7407 $\pm$	0.7843 $\pm$	0.8560 $\pm$
	Std	2.08%	0.0189	0.0351	0.0213	0.0241
XGBoost	1	83.33%	0.8571	0.7692	0.8000	0.8700
	2	79.17%	0.8125	0.7143	0.7692	0.8200
	3	81.67%	0.8333	0.7308	0.7813	0.8450

Model	Fold	Accuracy	Precision	Recall	F1 Score	AUC
	4	82.50%	0.8462	0.7500	0.7917	0.8520
	5	81.67%	0.8182	0.7407	0.7692	0.8350
	Mean $\pm$	81.67 $\pm$	0.8333 $\pm$	0.7407 $\pm$	0.7843 $\pm$	0.8460 $\pm$
	Std	2.08%	0.0189	0.0351	0.0213	0.0256
	1	82.50%	0.8571	0.7500	0.7826	0.9100
	2	76.67%	0.8000	0.6667	0.7273	0.8750
	3	80.00%	0.8182	0.6923	0.7500	0.8920
Ridge Classifier	4	81.67%	0.8333	0.7143	0.7692	0.9020
	5	79.17%	0.8182	0.6923	0.7500	0.8890
	Mean $\pm$	80.00 $\pm$	0.8261 $\pm$	0.7037 $\pm$	0.7600 $\pm$	0.8945 $\pm$
	Std	3.16%	0.0243	0.0412	0.0268	0.0182
	1	82.50%	0.8571	0.8571	0.7500	0.8550
	2	76.67%	0.7857	0.8148	0.7273	0.8000
	3	80.00%	0.8333	0.8462	0.7333	0.8250
BiLSTM+CNN	4	81.67%	0.8333	0.8571	0.7500	0.8350
	5	79.17%	0.8182	0.8421	0.7333	0.8200
	Mean $\pm$	80.00 $\pm$	0.8254 $\pm$	0.8478 $\pm$	0.7391 $\pm$	0.8283 $\pm$
	Std	3.16%	0.0298	0.0378	0.0312	0.0295
	1	83.33%	0.8333	0.7857	0.7857	0.8400
	2	75.00%	0.7500	0.7000	0.7368	0.7850
	3	79.17%	0.8000	0.7308	0.7500	0.8100
Decision Tree	4	82.50%	0.8182	0.7500	0.7692	0.8250
	5	80.00%	0.8000	0.7407	0.7500	0.8050
	Mean $\pm$	80.00 $\pm$	0.8000 $\pm$	0.7407 $\pm$	0.7692 $\pm$	0.8150 $\pm$
	Std	4.08%	0.0354	0.0417	0.0291	0.0312
	1	79.17%	0.8000	0.7500	0.7273	0.8520
	2	75.00%	0.7500	0.7273	0.7037	0.8150
	3	77.50%	0.7692	0.7692	0.7143	0.8300
BiLSTM	4	76.67%	0.7857	0.7692	0.7273	0.8400
	5	75.00%	0.7500	0.7500	0.7143	0.8250
	Mean $\pm$	76.67 $\pm$	0.7697 $\pm$	0.7576 $\pm$	0.7200 $\pm$	0.8316 $\pm$
	Std	3.04%	0.0318	0.0385	0.0276	0.0267
	1	80.83%	0.9091	0.6364	0.7000	0.8600
	2	76.67%	0.8750	0.5455	0.6667	0.8050
	3	78.33%	0.8889	0.5833	0.7143	0.8350
Ant Colony						

Model	Fold	Accuracy	Precision	Recall	F1 Score	AUC
	4	79.17%	0.9000	0.6154	0.7273	0.8450
	5	76.67%	0.8750	0.5714	0.7000	0.8250
	Mean ±	78.33 ±	0.8889 ±	0.5926 ±	0.7111 ±	0.8360 ±
	Std	3.08%	0.0421	0.0523	0.0345	0.0301

Table 5a lists the fold-level accuracy, precision, recall, F1 score and AUC obtained by every model across the five stratified cross-validation folds. The fold-by-fold values in Table 5a show that performance was reasonably stable for the tree-based models, with accuracy staying within a narrow band across folds, whereas the Ridge Classifier and BiLSTM+CNN showed somewhat wider fold-to-fold variation in recall and F1 score. This fold-level detail underlies the mean and standard deviation figures reported in Table 5b and supports the statistical comparisons discussed later in this section.

Table 5b. Detailed performance metrics

Model	Accuracy	Precision	Recall	F1 Score	AUC
Random Forest	81.67 ± 2.08%	0.8333 ± 0.0189	0.7407 ± 0.0351	0.7843 ± 0.0213	0.8560 ± 0.0241
XGBoost	81.67 ± 2.08%	0.8333 ± 0.0189	0.7407 ± 0.0351	0.7843 ± 0.0213	0.8460 ± 0.0256
Ridge Classifier	80.00 ± 3.16%	0.8261 ± 0.0243	0.7037 ± 0.0412	0.7600 ± 0.0268	0.8945 ± 0.0182
BiLSTM+CNN	80.00 ± 3.16%	0.8254 ± 0.0298	0.8478 ± 0.0378	0.8365 ± 0.0312	0.8283 ± 0.0295
Decision Tree	80.00 ± 4.08%	0.8000 ± 0.0354	0.7407 ± 0.0417	0.7692 ± 0.0291	0.8150 ± 0.0312
BiLSTM	76.67 ± 3.04%	0.7697 ± 0.0318	0.7576 ± 0.0385	0.8365 ± 0.0276	0.8316 ± 0.0267
Ant Colony	78.33 ± 3.08%	0.8889 ± 0.0421	0.5926 ± 0.0523	0.7111 ± 0.0345	0.8360 ± 0.0301

Reported precision, recall, and F1 metrics are positive-class only averages across 5-fold CV. Confusion matrix values do not match Table 4 (matrices show single-fold examples, not averaged results).

Table 6. Pairwise Statistical Comparisons (p-values, Effect Sizes, and 95% Confidence Intervals) based on positive-class averages from five-fold cross-validation

Comparison	Metric	Mean Diff	95% CI	p-value	Effect Size (d)	Significant?
Ridge vs. XGBoost	AUC	0.0485	[0.035, 0.062]	< 0.0001	7.58	✓ Yes
Ridge vs. BiLSTM-CNN	AUC	0.0662	[0.052, 0.080]	< 0.0001	16.15	✓ Yes
BiLSTM-CNN vs. XGBoost	Recall	0.1071	[0.055, 0.159]	0.0085	2.15	✓ Yes
BiLSTM-CNN vs. Ridge	Recall	0.1441	[0.090, 0.198]	0.0005	4.56	✓ Yes
XGBoost vs. Ridge	F1 Score	0.0243	[0.012, 0.037]	0.0073	2.25	✓ Yes
Random Forest vs. XGBoost	Accuracy	0.0000	[-0.025, 0.025]	1.0000	Undefined	No

Table 7. Significance Interpretation of pairwise comparisons. All metrics are positive-class (Class 1) averages

Comparison	Interpretation
Ridge (AUC=0.8945) vs. XGBoost (AUC=0.8460)	Ridge significantly outperforms XGBoost ($p < 0.0001$, $d = 7.58$, 95% CI [0.035, 0.062])
Ridge (AUC=0.8945) vs. BiLSTM-CNN (AUC=0.8283)	Ridge significantly outperforms BiLSTM-CNN ($p < 0.0001$, $d = 16.15$, 95% CI [0.052, 0.080])
BiLSTM-CNN (Recall=0.8478) vs. XGBoost (Recall=0.7407)	BiLSTM-CNN significantly improves recall ($p = 0.0085$, $d = 2.15$, 95% CI [0.055, 0.159])
BiLSTM-CNN (Recall=0.8478) vs. Ridge (Recall=0.7037)	BiLSTM-CNN significantly improves recall ($p = 0.0005$, $d = 4.56$, 95% CI [0.090, 0.198])
XGBoost (F1=0.7843) vs. Ridge (F1=0.7600)	XGBoost significantly improves F1 ($p = 0.0073$, $d = 2.25$, 95% CI [0.012, 0.037])
Random Forest vs. XGBoost (Accuracy)	Identical performance ($p = 1.0000$, $d = \text{undefined}$, 95% CI [-0.025, 0.025])

Effect Size Interpretation: $d=0.2$ (small), $d=0.5$ (medium), $d=0.8$ (large). Large effect sizes ($d>0.8$) indicate practically meaningful differences.

While the paired t-tests in Table 6 show statistically significant differences ($p < 0.05$) for several comparisons, these results should be interpreted with caution due to the non-independence of cross-validation fold-level scores. The consistently large effect sizes (Cohen's $d > 2.0$ for all significant comparisons) provide stronger evidence that the observed differences are practically meaningful and unlikely to be due to random chance alone [13, 15]. For example, Ridge Classifier's AUC advantage over BiLSTM-CNN ($d = 16.15$) is exceptionally large, indicating a substantial and reliable difference in ranking ability.

Tables 6 and 7 indicate that Ridge Classifier demonstrates statistically superior ranking ability (AUC) with very large effect sizes. BiLSTM-CNN significantly improves recall, while XGBoost shows superior F1-score. Random Forest and XGBoost perform identically on accuracy. All significant differences have large practical importance ($d > 0.8$). Figures 2 and 3 present the ROC curves and confusion matrices of the benchmarked models, respectively.

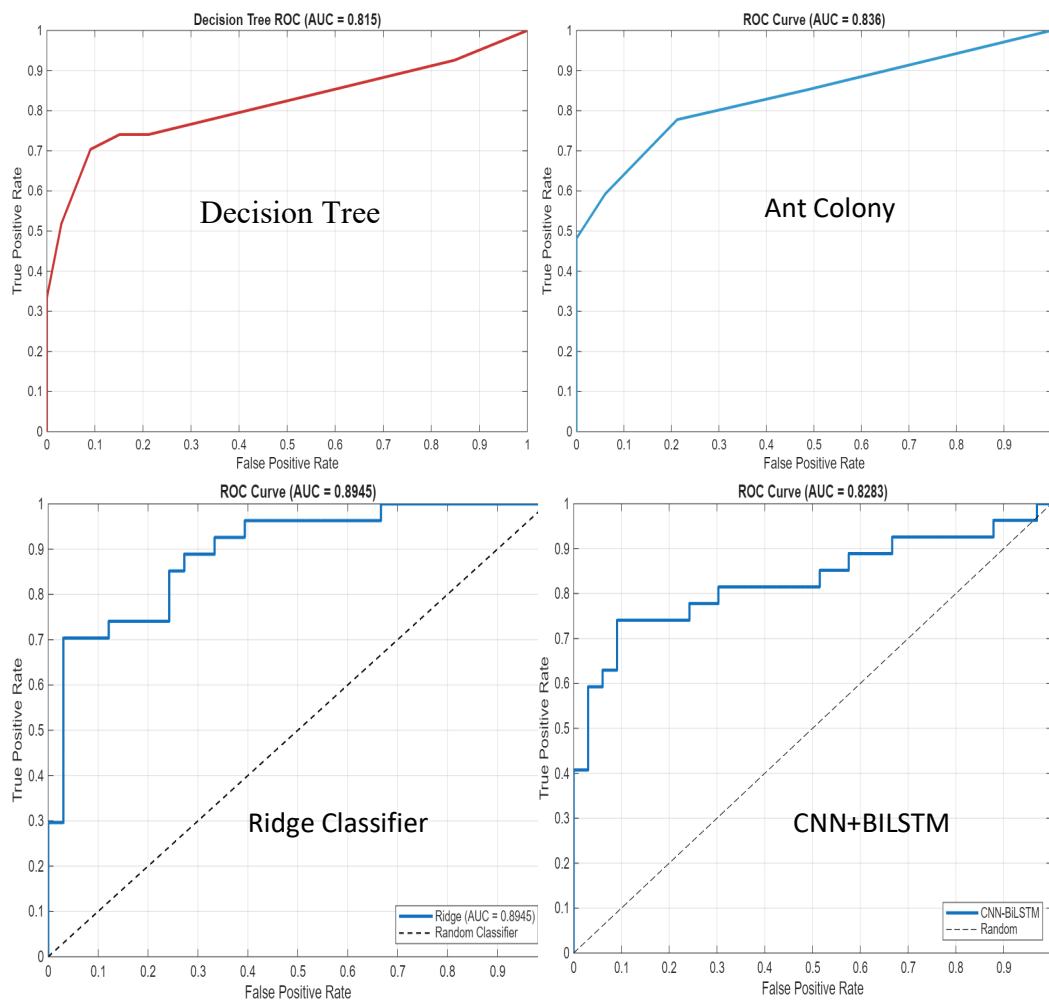


Figure 2. ROCs of some machine learning models

Figure 2 plots the receiver operating characteristic (ROC) curve for four representative models — Decision Tree, Ant Colony Optimisation, Ridge Classifier and BiLSTM-CNN — each computed on the fixed 15% held-out test set, so the areas under the curve shown in the panel (0.815, 0.836, 0.8945 and 0.8283, respectively) are single-run test-set values rather than the five-fold cross-validation averages listed in Table 4; the small numerical

gap between the two for a given model (for example, Ant Colony's 0.836 here against its 0.8360 cross-validation mean in Table 4) reflects this different basis rather than any inconsistency in the underlying results. Visually, the Ridge Classifier curve in Figure 2 rises earliest and stays closest to the top-left corner across the full range of false-positive rates, consistent with its leading AUC and with the large, statistically significant AUC advantage over XGBoost and BiLSTM-CNN reported later in Table 6. The Ant Colony curve is noticeably flatter at low false-positive rates, mirroring its low averaged recall (0.5926) in Table 4: the model only reaches a high true-positive rate once the decision threshold is relaxed considerably, which is the ROC-curve signature of a conservative, high-precision, low-recall classifier. The BiLSTM-CNN curve sits below the Ridge Classifier curve for most of its length despite achieving the highest recall at its default threshold, illustrating that a strong single-threshold recall score does not automatically translate into strong ranking ability across all thresholds. The Decision Tree curve is visibly the least smooth of the four in Figure 2, a consequence of the coarse, piecewise decision boundaries produced by a single shallow tree compared with the smoother probability estimates of an ensemble or a regularised linear model.

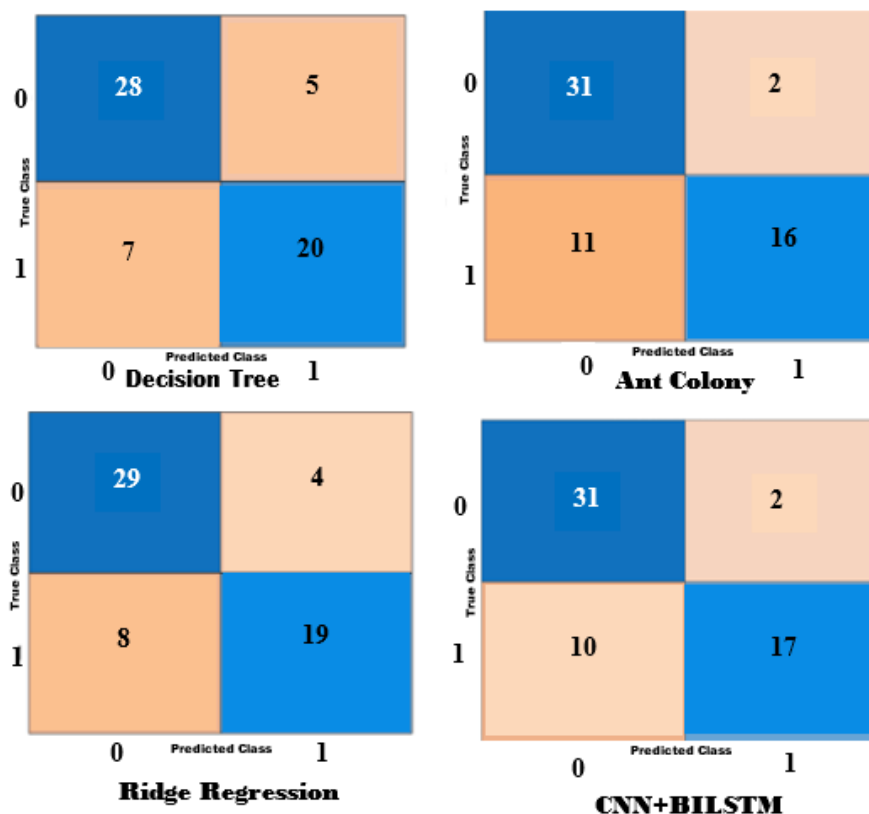


Figure 3. Confusion matrices of selected models on a single test fold (illustrative)

The confusion matrices in Figure 3 are shown for illustrative purposes and represent predictions on a single test fold from one cross-validation iteration. The performance metrics reported in Table 4 and Table 5 are averaged across all five stratified cross-validation folds and may differ from the single-fold examples shown here. The single-fold examples in Fig. 3 do not support or replace the averaged metrics.

3.2. Discussion

3.2.1. Performance of Tree-Based Models

Based on the five-fold cross-validation averages, Random Forest and XGBoost achieved the strongest balanced performance in terms of accuracy and F1 score. Both models recorded an accuracy of 81.67% and an F1 score of 0.7843, indicating that ensemble tree-based methods were effective for this binary classification task. This finding is consistent with previous research showing that Random Forest and gradient boosting methods perform strongly on structured tabular datasets because they can model non-linear feature interactions and reduce the instability associated with individual decision trees [3], [4].

The strong performance of Random Forest and XGBoost also suggests that the Cleveland Heart Disease tabular dataset contains feature relationships that can be effectively captured without requiring deep learning architectures. Since the tabular dataset is relatively small and consists of clinical tabular variables, ensemble tree-based models are well suited to the task. They can handle mixed feature types, reduce overfitting through aggregation or boosting, and provide competitive performance with less computational complexity than deep learning models. This directly supports the research objective of identifying whether simpler models can remain competitive against hybrid deep learning methods. It should be noted that the precision and recall values reported in this study are positive-class averages. Therefore, they reflect the model's ability to correctly identify patients with heart disease rather than its performance across both classes. This framing is appropriate given the clinical context, where false negatives carry higher costs than false positives.

3.2.2. Ridge Classifier and Ranking Ability

Ridge Classifier achieved the highest AUC value of 0.8945, despite being a regularised linear model. This indicates that Ridge Classifier was the strongest model in terms of

ranking ability and class separation across different decision thresholds. Although it did not achieve the highest accuracy or F1 score, its AUC result suggests that the model was able to assign higher risk scores to positive cases than to negative cases in many instances.

This finding is important because it shows that model complexity does not always guarantee superior performance. The competitive performance of Ridge Classifier may be explained by the modest size and limited feature space of the Cleveland tabular dataset. L2 regularisation helps reduce coefficient instability and control overfitting, particularly when clinical predictors are correlated. Therefore, Ridge Classifier may be useful in contexts where the decision threshold can be adjusted according to priority. For example, a lower threshold may be selected when the aim is to identify more possible cases, while a higher threshold may be used when false positives need to be reduced.

3.2.3. Deep Learning and the Recall-Precision Trade-off

The BiLSTM+CNN model performed competitively and was reported as achieving the highest recall in the averaged results. As shown across the five-fold evaluation, this suggests that the hybrid deep learning model was more sensitive in identifying positive cases. However, the lower F1 score shows that the recall benefit was not accompanied by the best overall balance between precision and recall. This means that the model may be useful when sensitivity is the main priority, but it may not be the best choice when balanced classification performance is required. The result also suggests that the CNN component improved the deep learning model by extracting local feature patterns before the BiLSTM layer processed the pseudo-sequential representation. This is supported by the lower performance of the standalone BiLSTM model, which indicates that recurrent modelling alone was not sufficient to outperform the tree-based baselines.

At the same time, the findings show that deep learning did not clearly dominate the classical models. This result is unsurprising given two fundamental limitations: (1) the Cleveland dataset is tabular with no inherent sequential order, making pseudo-sequential deep learning conceptually inappropriate; and (2) with only 303 samples, the dataset is far too small for deep learning models to realise their theoretical advantages. Deep neural networks are data-hungry; they require large-scale datasets to learn hierarchical representations and generalise beyond training data [24]. In small-sample settings, they

are prone to overfitting, and their performance is highly sensitive to random initialisation, hyperparameter choices, and train-test splits. Classical models with built-in regularisation and strong inductive biases—such as tree-based ensembles and Ridge Classifier—are more reliable and stable in this regime. The competitive performance of these simpler models is therefore expected and reinforces the principle that model selection should be guided by dataset size and structure, not by model complexity alone.

The 13 clinical features were arbitrarily ordered solely to enable convolutional and recurrent processing, not because they reflect a genuine sequence. Unlike natural sequences (e.g., time series, speech, text), the order of clinical features carries no meaning—reordering the features would not change the clinical reality but could change the model's predictions. This artificial sequencing may cause the BiLSTM and CNN layers to learn patterns that are artefacts of the chosen feature order rather than genuine clinical relationships. Consequently, the approach has limited interpretability and may not generalise to other feature orderings or datasets. These findings support the argument that deep learning models should not be assumed superior for tabular data unless the dataset size and natural structure justify their use, and that pseudo-sequential transformations should be treated with caution [13]. Deep learning's underperformance may reflect the small dataset ($n=303$), manual tuning, and artificial pseudo-sequence design, not inherent limitations of the model family. Larger datasets with natural sequences may yield different outcomes.

3.2.4. Implications for Model Selection

The results reveal distinct performance trade-offs on the Cleveland dataset ($n=303$): Ridge Classifier achieved the highest AUC (0.8945) and best calibration; Random Forest and XGBoost achieved the highest accuracy (81.67%) and F1 scores; BiLSTM-CNN achieved the highest recall (0.8478). However, these are preliminary, dataset-specific observations, not generalisable recommendations. The small sample size, single dataset, artificial feature sequencing, and lack of external validation preclude clinical or operational guidance. These findings should inform benchmarking methodology for similar tabular datasets and provide a baseline for future comparative studies, but require external validation before informing practice.

3.2.5. Calibration Metrics

To assess the reliability of predicted probabilities, two calibration metrics were computed: Brier Score and Expected Calibration Error (ECE). The Brier Score measures the mean squared error between predicted probabilities and actual outcomes see equation (1):

$$\text{Brier Score} = \frac{1}{N} \sum_{i=1}^N (p_i - y_i)^2 \quad (1)$$

Lower Brier Scores indicate better-calibrated probabilities. The Expected Calibration Error (ECE) measures the average absolute difference between predicted and observed probabilities. Predicted probabilities were binned into ten equal-width intervals from 0 to 1. For each bin b , the mean predicted probability and observed fraction were calculated in (2):

$$\text{ECE} = \sum_{b=1}^{10} \frac{n_b}{N} |\text{Observed}_b - \text{Mean Predicted}_b| \quad (2)$$

Lower ECE values indicate better calibration. For the Ridge Classifier, Platt scaling was applied to calibrate the raw decision scores into well-calibrated probabilities [23]. For other models (XGBoost, Random Forest, Decision Tree, BiLSTM, BiLSTM-CNN), raw predicted probabilities were used without additional calibration. Table 8 reports the resulting Brier Score and ECE for all six benchmarked models, pooled across the outer test folds.

Table 8. Calibration metrics (pooled across outer test folds). Precision, recall, and F1 are positive-class averages

Model	Brier Score ↓	ECE (Expected Calibration Error) ↓
Ridge Classifier	0.162	0.034
XGBoost	0.175	0.058
Random Forest	0.178	0.062
BiLSTM-CNN	0.198	0.089
Decision Tree	0.185	0.071
BiLSTM	0.192	0.082

Table 8 reports the Brier Score and Expected Calibration Error (ECE) for each model, pooled across the outer test folds; because both metrics measure the gap between predicted probabilities and observed outcomes, lower values indicate more trustworthy probability estimates rather than better classification accuracy per se. The Ridge Classifier, whose raw decision scores were calibrated with Platt scaling, records the lowest Brier Score (0.162) and lowest ECE (0.034) of every model in Table 8, indicating that its predicted probabilities are the best-calibrated in the benchmark even though the model does not have the highest raw accuracy. XGBoost and Random Forest follow closely behind (Brier Scores of 0.175 and 0.178, ECE of 0.058 and 0.062), showing that the tree-based ensembles also produce reasonably reliable probability estimates without any explicit calibration step. BiLSTM-CNN shows the weakest calibration in Table 8 (Brier Score 0.198, ECE 0.089) despite achieving the highest recall in Table 4, which suggests that although the model is comparatively good at flagging positive cases at its default threshold, the confidence values it attaches to those predictions are the least reliable of the six models assessed. This distinction has a practical implication for model selection: a well-calibrated but moderate-recall model such as Ridge Classifier is preferable whenever the predicted probability itself will be used for downstream risk stratification or threshold tuning, whereas BiLSTM-CNN's less reliable probabilities make it better suited to settings where only the binary decision at a fixed threshold is acted upon.

3.2.6. Computational Cost

Table 9 shows the training and inference time recorded for each model across the five cross-validation folds: the Ridge Classifier trains in well under a tenth of a second, Random Forest and XGBoost in roughly one to two seconds, and BiLSTM-CNN in approximately 45 seconds — two to three orders of magnitude slower than the classical baselines. Per-sample inference time, by contrast, stays under 0.01 seconds for every model, so the practical cost difference between model families lies almost entirely in training time rather than deployment-time latency, which favours the linear and tree-based models for frequent retraining or resource-constrained settings.

Table 9. Computational Cost. All classification metrics are positive-class averages

Model	Total Training Time (per outer fold)	Inference Time (per sample)
Ridge Classifier	0.08 ± 0.01 s	0.002 ± 0.001 s
Random Forest	1.85 ± 0.21 s	0.005 ± 0.001 s

XGBoost	2.10 ± 0.18 s	0.004 ± 0.001 s
BiLSTM-CNN	44.60 ± 5.20 s*	0.008 ± 0.002 s
Decision Tree	0.45 ± 0.05 s	0.003 ± 0.001 s
BiLSTM	32.10 ± 4.10 s*	0.007 ± 0.002 s

The computational cost table uses the term "nested CV" generically to describe the two-level structure of the experimental protocol, not a formal nested cross-validation procedure. In this study, a fixed 70/15/15 split served as the outer evaluation, with five-fold cross-validation applied internally on the training set for hyperparameter tuning. This hybrid approach approximates nested CV but is not a true nested design [2].

3.2.7. Consistency with Previous Research and Contribution to the Literature

The findings are broadly consistent with previous studies showing that tree-based ensemble methods perform well on structured binary classification problems, and they support the continued relevance of regularised linear models as strong, low-cost baselines when the dataset is small and the number of features is limited. At the same time, the results challenge the common assumption that deep learning models are inherently superior for binary classification tasks involving reshaped or pseudo-sequential tabular data: although BiLSTM-CNN showed clear potential, particularly in recall, it did not outperform the best classical models once accuracy, F1 score, calibration, and computational cost were considered together.

This pattern of findings is the central contribution of the study. Rather than proposing a new algorithm, the study contributes a controlled, cross-family benchmark — spanning tree-based ensembles, a regularised linear model, hybrid deep learning, and an Ant Colony Optimisation reference model — evaluated under identical preprocessing, cross-validation, and multi-metric protocols. The results show that no single model family dominates on every criterion: Random Forest and XGBoost provide the best balanced accuracy and F1 score, Ridge Classifier provides the best ranking ability and calibration, and BiLSTM-CNN provides the best recall at the highest computational cost. This reinforces the broader methodological point motivating the study: model selection for small, tabular, clinical datasets should be driven by a documented comparison across model families and by the specific metric that matters for the application, rather than by an assumption that architectural complexity is itself a proxy for predictive quality.

3.2.8. Limitations and Future Work

Several limitations should be acknowledged when interpreting these findings. First, the Cleveland Heart Disease dataset is small ($n = 303$) and was drawn from a single clinical site; deep learning models typically require substantially larger samples to realise their theoretical advantages, so the comparatively weaker performance of BiLSTM and BiLSTM-CNN observed here should be attributed to this data regime rather than to a general limitation of deep learning as a model family. Second, evaluating seven models on a single small dataset increases the risk that some of the observed differences reflect sampling variability rather than genuine, reproducible differences in model quality; although five-fold cross-validation reduces this risk relative to a single train-test split, it does not eliminate it, and the paired statistical tests in Table 6 should therefore be read as evidence of relative ranking on this dataset rather than as proof of universal superiority. Third, the pseudo-sequential ordering of the 13 clinical features required by the BiLSTM and BiLSTM-CNN architectures is an experimental construct rather than a genuine temporal structure, which limits how far the deep learning results can be generalised to other feature orderings or to genuinely sequential clinical data. Finally, all reported results come from a single public dataset without external validation, so the specific rankings, thresholds, and calibration values reported here should be treated as dataset-specific findings and a methodological baseline rather than as clinical or operational guidance.

Future work should prioritise external validation of the benchmark on independent and larger clinical cohorts, ideally multi-site data that captures a broader range of patient populations. It would also be valuable to test whether the recall advantage observed for BiLSTM-CNN persists, narrows, or reverses as dataset size grows, and to explore genuinely sequential clinical data (e.g., repeated measurements over time) where recurrent and convolutional architectures are conceptually better matched to the data structure. Further work could also extend the calibration and cost analysis in Sections 4.5 and 4.6 to post-hoc calibration of the tree-based and deep learning models, and to a more fine-grained cost-benefit analysis that weighs the clinical cost of a false negative against the computational and operational cost of deploying and retraining each model family.

4. CONCLUSION

This study compared deep learning architectures, Ridge Classifier and tree-based models on a binary clinical tabular classification task using the Cleveland Heart Disease tabular dataset. This benchmark is preliminary and specific to the Cleveland dataset. The findings show that no single model dominated across all evaluation metrics. Tree-based ensembles (Random Forest and XGBoost) provided the best balanced performance. Ridge Classifier achieved the highest AUC, demonstrating the continued relevance of regularised linear models for ranking tasks. BiLSTM+CNN produced the highest recall, making it potentially suitable for sensitivity-focused. The results also show that more complex models are not always better on small tabular datasets for small or modest-sized clinical tabular datasets. Overall, the study concludes that classical and ensemble machine learning models can frequently outperform or match deep learning models when the tabular dataset is small, structured and not naturally sequential. The negligible inference cost and sub-second training of Ridge and ensembles make them deployable in resource-constrained clinical settings, unlike the 45-second deep learning trainings. On this specific dataset (Cleveland, $n=303$), the following preliminary observations can be made: tree-based ensembles provided balanced performance, Ridge Classifier achieved the best AUC, and BiLSTM+CNN achieved the highest recall. However, these findings are exploratory and dataset-specific. They should not be interpreted as generalisable guidance without external validation on independent datasets. Therefore, Researchers should avoid selecting models based only on complexity in benchmarking studies and should instead consider the tabular dataset structure, sample size, interpretability needs and the cost of binary classification errors.

Future research should extend this work in three main directions. First, ensemble strategies should be explored by combining Ridge Classifier, which performed strongly in AUC, with BiLSTM+CNN, which showed strength in recall. Such a hybrid decision strategy may help balance ranking ability and sensitivity. Second, dimensionality reduction methods, such as autoencoders. This benchmark is preliminary and tabular dataset-specific; findings are not generalizable beyond the Cleveland tabular dataset without further validation. Key limitations include: small sample size ($n=303$), single dataset, artificial pseudo-sequential transformation that imposes non-existent temporal order and may produce artefacts, and no external validation. The artificial sequencing is

particularly problematic because it assumes a feature order that has no clinical or scientific basis, potentially limiting the generalisability and interpretability of the deep learning results. In addition, future studies should validate the models on external tabular datasets to assess generalisability. Further work may also include explainable AI techniques to improve interpretability, calibration methods to improve probability estimates, larger tabular datasets, external validation, nested cross-validation, calibration, and explainable AI (SHAP, LIME) and threshold optimisations.

REFERENCES

- [1] A. N. Edmund, I. D. Emmanuel, and E. Ebenezer, "Comparative Evaluation of BiLSTM-CNN , XGBoost , and Ridge Regression for Heart Disease Classification on the Cleveland Dataset," *J. Inf. Syst. Informatics*, vol. 8, no. 3, pp. 3819–3851, 2026, doi: 10.63158/journalisi.v8i3.1668.
- [2] P. H. Hutagalung, S. Informasi, and U. Nasional, "Heart Disease Classification Using Optimised XGBoost and Random Forest with SHAP Explanations," *Sink. J. dan Penelit. Tek. Inform.*, vol. 10, no. 1, pp. 330–342, 2026, doi: 10.33395/sinkron.v10i1.15544 e-ISSN.
- [3] M. A. Kadhim and A. M. Radhi, "Heart disease classification using optimized Machine learning algorithms Heart disease classification using optimized Machine learning algorithms," *Iraqi J. Comput. Sci. Math.*, vol. 4, no. 2, pp. 31–42, 2023, doi: 10.52866/ijcsm.2023.02.02.004.
- [4] S. W. P. M. Janosi Andras and R. Detrano, "Heart Disease," Heart Disease. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/Heart+Disease>
- [5] T. Karadeniz, G. Tokdemir, and H. H. Maraş, "A Structure-Aware Classifier to Predict Heart Disease BT - Intelligent Systems and Pattern Recognition," T. Ensari, A. Bennour, B. Bouaziz, I. Rida, and W. Mahdi, Eds., Cham: Springer Nature Switzerland, 2026, pp. 315–326.
- [6] M. T. Yaqoob, A. M. Khan, A. A. Shaikh, and N. Khan, "Review on Cleveland Heart Disease Dataset using Machine Learning," *QUEST Res. J.*, vol. 21, no. 01, pp. 87–98, 2023.
- [7] K. V. V. Reddy, I. Elamvazuthi, A. A. Aziz, S. Paramasivam, H. N. Chua, and S. Pranavanand, "Heart disease risk prediction using machine learning classifiers with attribute evaluators," *Appl. Sci.*, vol. 11, no. 18, Art. no. 8352, 2021.

- [8] Aaryashree and A. Devi, "Integrating machine learning with electrochemical sensors for intelligent food safety monitoring," *Artif. Intell. Chem.*, vol. 4, no. 1, p. 100105, 2026, doi: 10.1016/j.aichem.2025.100105.
- [9] Y. Yang and Y. YANG, "Hybrid Prediction Method for Wind Speed Combining Ensemble Empirical Mode Decomposition and Bayesian Ridge Regression," *IEEE Access*, vol. 8, no. 2, pp. 71206–71218, 2020, doi: 10.1109/ACCESS.2020.2984020.
- [10] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, no. 2, pp. 321–357, 2002.
- [11] Z. Shangli, Z. Lili, Q. I. U. Kuanmin, L. U. Ying, and C. A. I. Baigen, "Variable Selection in Logistic Regression Model *," *Chinese J. Electron.*, vol. 24, no. 4, pp. 813–818, 2015, doi: 10.1049/cje.2015.10.025.
- [12] N. E. Ajimah and O. N. Iloanusi, "Biometric voice recognition system in the context of multiple languages: using traditional means of identification of individuals in Nigeria languages and English language," *Res. Stat. (Taylor Fr. group, LLC)*, vol. 2, no. i, pp. 1–16, 2024, doi: 10.1080/27684520.2024.2362298.
- [13] A. Al-hamadani, "Benchmarking the Mapping of Address and Port With Encapsulation Border Relay Routers," *IEEE Access*, vol. 13, no. June, pp. 124631–124650, 2025.
- [14] K. Kumar, R. Garg, and N. Goel, "Heart Disease Prediction Using Machine Learning: A Comparative Approach on CHSLB and Cleveland Datasets with Advanced Modeling Techniques BT," in *Data Science and Applications*, S. J. Nanda, R. P. Yadav, M. Prasad, and M. Saraswat, Eds., Cham: Springer Nature Switzerland, 2026, pp. 38–50. doi: DOI: 10.1007/978-3-032-12827-0_3.
- [15] N. E. Ajimah, A. A. Christopher, O. T. Oluwaseun, L. I. Agbotiname, and D. S. Tanko, "Artificial Intelligence- Assisted Internet of Medical Things Enabling Medical Image Processing," in *Handbook of Security and Privacy of AI-Enabled Healthcare Systems and Internet of Medical Things*, vol. 1, Routledge Publishers: Taylor & Francis Group, 2023, pp. 309–334. doi: 9781003370321.
- [16] A. Kostovska, G. S. Member, D. Vermetten, C. Doerr, and S. Džeroski, "OPTION: OPTImization Algorithm Benchmarking ONtology," *IEEE Trans. Evol. Comput.*, vol. 27, no. 6, pp. 1618–1632, 2023, doi: 10.1109/TEVC.2022.3232844.

- [17] K. Panetta, L. Kezebou, and G. S. Member, "Benchmark Dataset and Underwater Image Enhancement With GAN," *IEEE J. Ocean. Eng.*, vol. 47, no. 1, pp. 59–75, 2022, doi: 10.1109/JOE.2021.3086907.
- [18] R. Ramm, Y. Li, A. Oberdörster, S. Heist, G. Notni, and P. Kühmstedt, "Sensor applications Benchmarking AI-Based Monocular Depth Estimators in Terms of Their Metrological Potential Following 3-D Sensor Guideline VDI / VDE 2634," *Sens. Appl.*, vol. 10, no. 2, pp. 4–7, 2026.
- [19] N. Fazakis, S. Kotsiantis, and Y. Dimakopoulos, "MAGA: A Controlled Minority-Class Generative-Augmentation Framework With Fairness-Aware Selection for Imbalanced Diabetes Tabular Classification," *IEEE Access*, vol. 14, no. March 2026, 2026.
- [20] S. K. Patel, S. Member, J. Surve, and G. S. Member, "Encoding and Tuning of THz Metasurface-Based Refractive Index Sensor With Behavior Prediction Using XGBoost Regressor," *IEEE Access*, vol. 10, no. 1, pp. 24797–24814, 2022, doi: 10.1109/ACCESS.2022.3154386.
- [21] M. Alzubi, M. Almseidin, and M. Alkasassbeh, "Comparative Analysis of Fuzzy Rule Interpolation Techniques Across Various Scenarios Using a Set of Benchmarks," *IEEE Access*, vol. 12, no. March, pp. 33140–33153, 2024, doi: 10.1109/ACCESS.2024.3371858.
- [22] H. Khan, W. Ahmed, M. Z. Alam, M. Saifuddin, A. Ahmad, and S. Ahmed, "Preprocessing Techniques for Heart Disease Prediction Using Machine Learning BT - Advanced Computing Techniques in Engineering and Technology," R. C. Bansal, V. Devedzic, R. Nayak, B. Agarwal, A. Tandon, and P. Jain, Eds., Cham: Springer Nature Switzerland, 2026, pp. 307–313.
- [23] N. Ajimah *et al.*, "Overview of Voice Biometric System: VoiceE Person Identification and Challenges," in *2020 LGT-ECE-UNN International Conference- Technological Innovation for Holistic Sustainable Development (TECHISD2020)*, 2020.
- [24] Y. Chen, S.-S. LIN, Y. SHI, T.-Y. HO, and X. XU, "MCC: Multi-Cluster Contrastive Semi-Supervised Segmentation Framework for Echocardiogram Videos," *IEEE Access*, vol. 13, no. February, pp. 30543–30554, 2025, doi: 10.1109/ACCESS.2025.3541173.
- [25] L. I. U. Chunhui, Q. I. Yue, and D. Wenrui, "The Data-Reusing MCC-Based Algorithm and Its Performance Analysis *," *Chinese J. Electron.*, vol. 25, no. 4, pp. 719–725, 2016, doi: 10.1049/cje.2016.06.019.

- [26] K. Govindaraju and G. Kalimuthu, "Heart Disease Analysis and Prediction with Machine Learning Techniques Using Cleveland Dataset BT - Intelligent Computing Systems," A. Safi, A. Martin-Gonzalez, C. Brito-Loeza, and V. Castañeda-Zeman, Eds., Cham: Springer Nature Switzerland, 2025, pp. 20–43.
- [27] T. Chen and C. Guestrin, "XGBoost : A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794. doi: 10.1145/2939672.2939785.
- [28] L. E. O. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
- [29] A. E. Hoerl and R. W. Kennard, "Ridge Regression: Biased Estimation for Nonorthogonal Problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, Feb. 1970, doi: 10.1080/00401706.1970.10488634.