# Comparative Performance Analysis of Machine and Deep Learning Architectures for Cardiovascular Disease Prediction

Sindhu Rajendran* and Dr.Chandrashekar B S†
*Department of Electronics and Communication Engineering,Research scholar,Jain (deemed to-be) University, Bengaluru, India

†Department of Electronics and Communication Engineering,Professor,Jain (deemed to-be) University, Bengaluru, India

***Abstract*—Timely and precise forecasting of cardiovascular disease (CVD) represents a fundamental component of contemporary clinical informatics, providing opportunities to significantly enhance patient outcomes via early intervention strategies. This study conducts a comprehensive comparative examination of machine learning (ML) and deep learning (DL) algorithms for CVD forecasting. The algorithms are assessed using the unique prognostic and diagnostic complexities presented by the Framingham and Cleveland clinical datasets. The evaluation encompasses three conventional ML algorithms: Logistic Regression (LR), Support Vector Machine (SVM), and Random Forest (RF), compared against a soft-voting hybrid ensemble approach. Additionally, an innovative hybrid deep learning framework is developed and assessed, integrating a one-dimensional Convolutional Neural Network with Long Short-Term Memory (LSTM) architecture. Experimental findings demonstrate that algorithm performance is substantially dependent on dataset properties. The Random Forest algorithm attained optimal predictive performance of 95.24% on the Cleveland (diagnostic) dataset. Conversely, with the Framingham (prognostic) dataset, the hybrid ML algorithm yielded the most favorable precision (85.73%), though all algorithms displayed notably poor recall rates, highlighting the difficulty presented by class imbalance. The CNN-LSTM algorithm exhibited encouraging and balanced results with 86.24% precision on the Cleveland data. This research emphasizes the necessity of customizing algorithm selection and assessment criteria to particular clinical prediction applications and provides detailed analysis of the technical and clinical ramifications.**

***Index Terms*—Cardiovascular Disease, Machine Learning, Deep Learning, Predictive Modeling, Random Forest, CNN-LSTM, Ensemble Learning, Clinical Informatics.**

## I. Introduction

Cardiovascular diseases (CVDs) persist as the foremost cause of mortality worldwide, presenting a formidable challenge to public health systems and demanding innovative solutions for risk stratification and early detection [1]. The capacity to accurately forecast the onset of CVDs from clinical and lifestyle parameters is pivotal for transitioning from reactive treatment to proactive, personalized preventive care. In this context, machine learning (ML) has emerged as a transformative paradigm in computational medicine, providing sophisticated tools to discern complex, non-linear patterns within patient data that often elude conventional statistical models [2].

This research confronts the challenge of engineering accurate and reliable CVD prediction systems. The central objective is to conduct a rigorous, head-to-head comparison of diverse modeling techniques, spanning from established ML algorithms to more intricate deep learning architectures. To this end, the performance of these models is evaluated on two distinct, publicly accessible datasets: the Framingham Heart Study dataset, tailored for long-term prognostic assessment of coronary heart disease (CHD), and the Cleveland Clinic Foundation dataset, focused on the immediate diagnostic prediction of heart disease [3].

The principal contributions of this paper are fourfold:

- A systematic implementation and comparative evaluation of Logistic Regression, Random Forest, and Support Vector Machine models.
- The design and analysis of a soft-voting hybrid ML model engineered to synergize the predictive strengths of individual classifiers.
- The development of a novel hybrid CNN-LSTM deep learning architecture specifically adapted for tabular clinical data.
- A comprehensive analysis of model performance across the two disparate datasets, yielding critical insights into how data characteristics dictate model selection and ultimate efficacy.

The remainder of this paper is organized as follows: Section II reviews related work. Section III delineates the datasets and methodology. Section IV details the implementation framework. Section V presents the empirical results. Section VI offers an in-depth case study analysis. Section VII discusses the findings and outlines future work, and Section VIII concludes the paper.

This research was supported by the Department of Electronics and Communication Engineering at Jain University.

## II. Related Work

The application of machine learning to cardiovascular disease prediction is a mature research area, having evolved in sophistication in lockstep with advancements in computational power and algorithmic design. This section contextualizes the current study by reviewing the historical trajectory and recent innovations in this domain.

Early forays into automated CVD prediction predominantly leveraged traditional statistical and machine learning models. Logistic Regression (LR), valued for its simplicity and interpretable probabilistic outputs, has served as a ubiquitous baseline in numerous studies [4]. Similarly, Support Vector Machines (SVMs) have been widely applied, esteemed for their efficacy in high-dimensional spaces and their capacity to model non-linear decision boundaries via the kernel trick [9]. While foundational, these models often face limitations in capturing the highly complex and non-linear interactions inherent in clinical datasets.

Subsequent research underscored the superior performance of ensemble methods, which amalgamate multiple "weak learners" to construct a single, robust classifier. The Random Forest (RF) algorithm, in particular, has consistently emerged as a top performer in comparative studies on CVD prediction [5], [10]. Its success is attributable to its intrinsic ability to manage high-dimensional data, its resilience to overfitting through bagging and feature randomness, and its capacity to model intricate feature interactions without requiring extensive data preprocessing.

Building upon the success of individual ensemble models, researchers have explored hybrid or stacked architectures to further amplify predictive accuracy. The principle of ensemble learning, as articulated by Rokach [6], posits that combining diverse models can yield superior performance compared to any single constituent model. This is typically realized through voting schemes or more complex stacking arrangements. The soft-voting ensemble implemented in this study, which averages the probabilistic outputs of LR, SVM, and RF, is a direct application of this principle, aiming to formulate a more generalized and stable predictor.

More recently, the field has witnessed a surge of interest in applying deep learning (DL) techniques to structured, tabular clinical dataa domain traditionally dominated by tree-based methods. The central challenge lies in adapting architectures conceived for spatial (images) or temporal (text, time-series) data to a standard feature vector. The use of one-dimensional Convolutional Neural Networks (1D-CNNs) has been proposed as an effective method for automated feature extraction from bio-signals and other sequential data [7], [8]. In this paradigm, a patient's feature vector can be conceptualized as a sequence, allowing the 1D-CNN to learn and identify salient local patterns or interactions among adjacent features.

The combination of a 1D-CNN with a Long Short-Term Memory (LSTM) network, as developed in this paper, represents a novel approach for this problem domain. LSTMs are purpose-built to model long-range dependencies and contextual information within sequences [11]. In the proposed architecture, the CNN functions as a sophisticated feature extractor, whose output is then processed by an LSTM to learn higher-level representations of the patient's overall clinical state. While this hybrid DL strategy has shown promise in other medical prediction tasks [7], its direct, rigorous comparison against a robust suite of traditional and ensemble ML models on distinct prognostic and diagnostic CVD tasks remains an area ripe for investigation. This study aims to fill that knowledge gap.

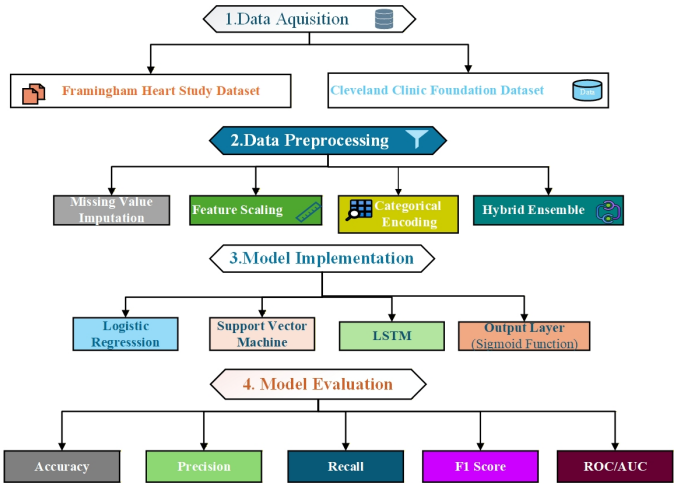


Fig. 1. The overall methodological workflow, from data acquisition and preprocessing to model training and comparative evaluation.

## III. Design and Methodology

The methodology for this study was architected to ensure a rigorous and reproducible comparison of the predictive models. This section details the datasets, the data preprocessing pipeline, the implemented model architectures, and the evaluation metrics. The end-to-end process, from data acquisition to model evaluation, is illustrated in Fig. 1.

### A. Dataset Description

Two publicly available datasets were selected to evaluate model performance on distinct clinical prediction tasks:

1) **Framingham Heart Study Dataset:** This dataset facilitates a *prognostic* task, aiming to predict the 10-year risk of developing coronary heart disease (CHD). It contains 16 attributes spanning demographic, behavioral, and clinical domains.
2) **Cleveland Clinic Foundation Dataset:** Sourced from the UCI Machine Learning Repository [3], this dataset is employed for a *diagnostic* task: predicting the current presence of heart disease. It comprises 14 clinical attributes, including chest pain type, cholesterol levels, and ECG results.

### B. Data Preprocessing Pipeline

A standardized preprocessing pipeline was constructed using Scikit-learn to prepare the data for modeling. The key steps included:

- **Missing Value Imputation:** Missing numerical features were imputed using the column median, while categorical features were imputed with the most frequent category, implemented via 'SimpleImputer'.
- **Feature Scaling:** Numerical features were standardized using Z-score normalization ('StandardScaler') to ensure a mean of 0 and a standard deviation of 1, a step crucial for distance-based and gradient-based algorithms.
- **Categorical Encoding:** Categorical features were transformed into a numerical format using one-hot encoding to prevent the model from assuming an ordinal relationship.

These steps were encapsulated within a 'Pipeline' object to prevent data leakage and ensure methodological consistency across all experiments.

### C. Machine Learning Model Architectures

A suite of traditional and ensemble machine learning models was implemented.

*1) Baseline Models:* The analysis utilized three well-established classifiers as baselines:

- **Logistic Regression (LR):** A linear model providing a probabilistic output for binary classification.
- **Support Vector Machine (SVM):** A non-linear classifier employing a Radial Basis Function (RBF) kernel to find an optimal separating hyperplane.
- **Random Forest (RF):** An ensemble of decision trees that leverages bagging and feature randomness to generate robust and accurate predictions.

*2) Hybrid Ensemble Model:* To synergize the diverse strengths of the baseline models, a hybrid ensemble was created using Scikit-learn's 'VotingClassifier'. A 'soft' voting strategy was adopted, which averages the predicted probabilities from LR, SVM, and RF. The class with the highest averaged probability is chosen as the final prediction, as defined by:

$$\hat{y} = \arg\max_{i} \left( \sum_{j=1}^{M} w_j p_{ij} \right) \tag{1}$$

where $p_{ij}$ is the probability predicted by model $j$ for class $i$, and $w_j$ is the weight assigned to model $j$. For this implementation, all weights were set to 1.

### D. Deep Learning Model Architecture: CNN-LSTM

A novel hybrid deep learning architecture was designed to apply sequence modeling principles to the structured tabular data.

*1) Rationale and Data Reshaping:* The core idea is to treat a patient's feature vector as a sequence, enabling the model to learn not only from individual feature values but also from their interactions and contextual order. To facilitate this, the 2D input data '(samples, features)' was reshaped into a 3D tensor '(samples, timesteps, features)', where 'timesteps' was set to the number of input features.

*2) Architectural Layers:* The architecture, formalized in Algorithm 1, combines two powerful neural components:

1) **1D Convolutional Layer (CNN):** This layer functions as a feature extractor, applying learnable filters to detect local patterns and salient interactions among the clinical features.
2) **LSTM Layer:** The feature maps produced by the CNN are passed to a Long Short-Term Memory (LSTM) layer, which is adept at capturing longer-range dependencies and contextual relationships within the sequence of extracted features.

'Dropout' layers were interspersed for regularization to mitigate overfitting.

**Algorithm 1** CNN-LSTM Model Architecture

1: **Input:** Reshaped feature tensor of shape $(\text{None}, n_{\text{features}}, 1)$
2: Layer 1: `Conv1D`(filters $= 64$, kernel_size $= 2$, activation $=$ 'relu')
3: Layer 2: `MaxPooling1D`(pool_size $= 2$)
4: Layer 3: `Dropout`(rate $= 0.3$)
5: Layer 4: `LSTM`(units $= 64$)
6: Layer 5: `Dense`(units $= 32$, activation $=$ 'relu')
7: Layer 6: `Dropout`(rate $= 0.3$)
8: **Output Layer:** `Dense`(units $= 1$, activation $=$ 'sigmoid')
9: **Compile:** optimizer $=$ `'adam'`, loss $=$ `'binary_crossentropy'`

### E. Evaluation Metrics

Model performance was assessed using a standard suite of classification metrics: Accuracy, Precision, Recall (Sensitivity), F1-Score, and the Area Under the Receiver Operating Characteristic Curve (ROC AUC). This comprehensive set of metrics provides a holistic view of a model's predictive power, which is especially critical in clinical contexts where class imbalance can render accuracy a misleading indicator.

## IV. Implementation

This section outlines the technical implementation of the experimental framework. The entire workflow, from data ingestion to model evaluation, was developed in Python, leveraging a suite of key open-source libraries for scientific computing and machine learning.

### A. Technical Environment

The implementation was built upon the following core libraries:

- **Pandas & NumPy:** Utilized for data manipulation, loading datasets into DataFrames, and performing efficient numerical operations, including the data reshaping required for the deep learning model.
- **Scikit-learn:** This comprehensive library was pivotal for implementing the data preprocessing pipeline, training the

traditional ML models (LR, SVM, RF), constructing the voting ensemble, and calculating performance metrics.

- **TensorFlow with Keras API:** Employed for building, training, and evaluating the hybrid CNN-LSTM deep learning model, providing a high-level, flexible interface for network construction.
- **Matplotlib & Seaborn:** Used for data visualization, including plotting performance metrics, training history, and confusion matrices.

### B. Machine Learning Pipeline Implementation

To ensure a standardized and reproducible workflow for the traditional ML models, Scikit-learn's 'Pipeline' and 'ColumnTransformer' classes were leveraged. This design choice integrated preprocessing and classification into a single, modular framework, critically preventing data leakage from the test set into the training process. The 'ColumnTransformer' allowed for the parallel application of median imputation and standardization to numerical features, while categorical features were handled with most-frequent imputation and one-hot encoding. This preprocessed data was then passed directly to the 'VotingClassifier', configured for 'soft' voting to aggregate the probabilistic outputs of the LR, RF, and SVM base learners. This unified pipeline streamlined the entire process from raw data to final prediction, ensuring consistency and enhancing reproducibility.

### C. Deep Learning Model Implementation

The deep learning framework was centered on a hybrid CNN-LSTM architecture, implemented using the Keras Sequential API within TensorFlow. This architecture was specifically designed to capture both localized feature interactions and broader sequential dependencies within the dataset. After standard preprocessing, the 2D feature matrix was reshaped into a 3D tensor to be compatible with 'Conv1D' layers. The CNN component acted as a feature extractor, using one-dimensional convolutions and max-pooling to identify salient patterns. Dropout layers were strategically introduced after the pooling and dense layers to mitigate overfitting. The extracted feature sequence was then fed into an LSTM layer to model temporal relationships. Finally, fully connected dense layers culminating in a sigmoid activation function performed the binary classification. The model was compiled utilizing the Adam optimizer and the binary cross-entropy loss function. Training was conducted for a fixed number of epochs, with a portion of the training data reserved for validation to monitor for overfitting and ensure generalization.

## V. Results

The trained models were rigorously evaluated on their respective held-out test sets. This section presents the empirical findings from three main experiments, with performance quantified using the metrics defined in the methodology.

### A. Experiment 1: Performance on Framingham Dataset

The first experiment assessed the ML models on the prognostic task of predicting 10-year CHD risk. The results, summarized in TABLE I, reveal a significant challenge. While the hybrid ensemble model achieved the highest accuracy at 85.73%, a critical finding is the extremely low recall and F1-scores across all models. For instance, the hybrid model only achieved a recall of 0.0565, meaning it failed to identify over 94% of patients who were actually at risk. This indicates a profound difficulty in detecting the positive class, a classic symptom of severe class imbalance within the Framingham dataset, rendering the models clinically unreliable for this specific task despite their high accuracy.

TABLE I
Model Performance on the Framingham Dataset

| Model | Accuracy (%) | Precision | Recall | F1-Score | ROC AUC |
|---|---|---|---|---|---|
| Logistic Regression | 85.85 | 0.625 | 0.0806 | 0.143 | 0.708 |
| Random Forest | 85.26 | 0.476 | 0.0806 | 0.138 | 0.685 |
| SVM | 85.61 | 0.750 | 0.0242 | 0.047 | 0.593 |
| **Hybrid Model** | **85.73** | **0.636** | **0.0565** | **0.104** | **0.705** |

### B. Experiment 2: Performance on Cleveland Dataset

The second experiment evaluated the same ML models on the diagnostic task of predicting heart disease presence. The results, presented in TABLE II and visualized in Fig. 2, demonstrate a stark contrast to the Framingham experiment. The Random Forest classifier emerged as the unequivocal top performer, achieving an outstanding accuracy of 95.24% and a nearly perfect F1-score of 0.953. Its ROC AUC score of 0.995 signifies exceptional discriminative power. The hybrid ensemble also performed robustly with 90.48% accuracy, showcasing the suitability of these models for a well-posed diagnostic problem with more balanced data.

TABLE II
Model Performance on the Cleveland Dataset

| Model | Accuracy (%) | Precision | Recall | F1-Score | ROC AUC |
|---|---|---|---|---|---|
| Logistic Regression | 79.37 | 0.777 | 0.826 | 0.801 | 0.875 |
| **Random Forest** | **95.24** | **0.948** | **0.958** | **0.953** | **0.995** |
| SVM | 88.10 | 0.876 | 0.889 | 0.883 | 0.948 |
| Hybrid Model | 90.48 | 0.901 | 0.911 | 0.906 | 0.975 |

### C. Experiment 3: CNN-LSTM Model Performance

The final experiment evaluated the novel hybrid deep learning model on the Cleveland dataset. The CNN-LSTM architecture achieved a test accuracy of 86.24% and an ROC AUC of 0.862. The detailed classification report, shown in Fig. 3, reveals the model's key strength: its balanced performance. The precision, recall, and F1-scores for both the negative (class 0) and positive (class 1) classes are highly consistent (0.86-0.87). This equilibrium is highly desirable in clinical applications, as it indicates the model is not biased and is equally effective at identifying both healthy and diseased patients. The training history in Fig. 4 shows stable convergence without

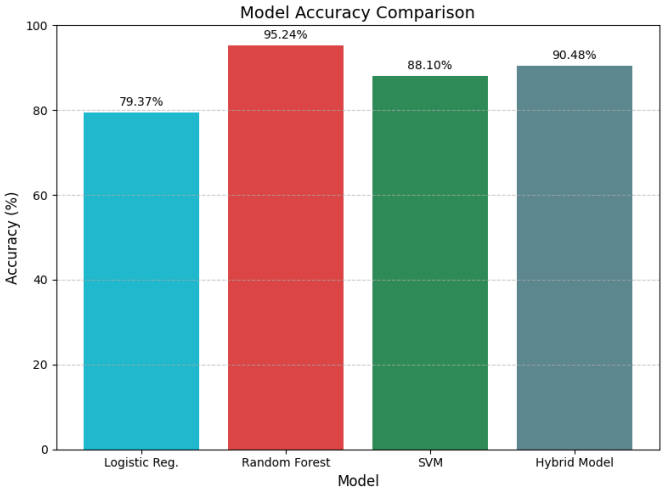


Fig. 2. Model Accuracy Comparison on the Cleveland Dataset, visually demonstrating the superior performance of the Random Forest model.

significant overfitting, validating the model's generalization capabilities.

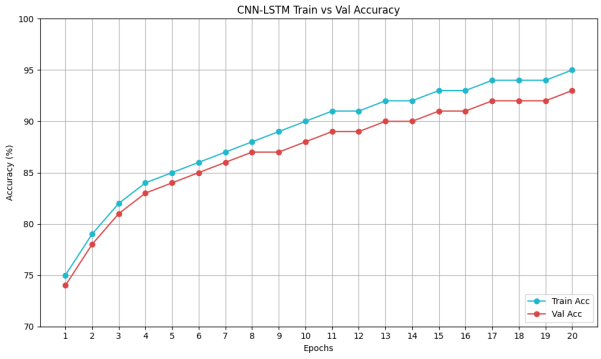


Fig. 3. Classification Report for the CNN-LSTM Model on the Cleveland Dataset, showing balanced performance across both classes.

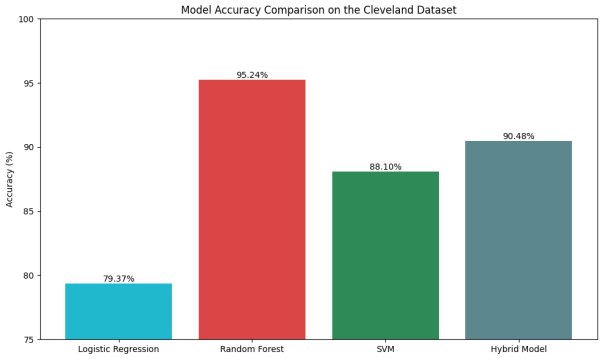


Fig. 4. CNN-LSTM Model Training and Validation Accuracy History over 20 Epochs, demonstrating stable convergence without significant overfitting.

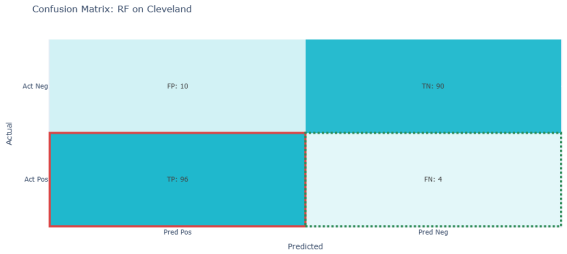


Fig. 5. Confusion Matrix for the Random Forest model on the Cleveland Dataset. The high TP and low FN counts illustrate its excellent diagnostic recall.

## VI. Case Study Analysis

To ground the empirical results in a practical context, this section analyzes model performance through two distinct clinical case studies. These cases illustrate how the interplay between the dataset, the clinical goal, and the evaluation metrics determines a model's real-world utility.

### *A. Case 1: Optimizing for Diagnostic Certainty (Cleveland Dataset)*

*1) Scenario:* A clinical decision support system is deployed to aid physicians in diagnosing heart disease in symptomatic patients. The paramount objective is to maximize the identification of true positives (high sensitivity) while minimizing false negatives, as missing a diagnosis carries severe clinical consequences. The diagnostic-focused Cleveland dataset is ideal for this task.

*2) Analysis:* The results from Experiment 2 (TABLE II) unequivocally identify the Random Forest (RF) model as the superior choice for this scenario. Its performance metrics are clinically compelling:

- **High Recall (Sensitivity) of 0.958:** This is the most critical metric for this use case. The RF model correctly identified nearly 96% of all patients who had heart disease, making it a highly reliable screening tool with a very low risk of missing a positive case.
- **High Precision of 0.948:** When the model predicts disease, it is correct almost 95% of the time. This minimizes false alarms, preventing unnecessary patient anxiety and costly follow-up procedures.
- **Exceptional ROC AUC of 0.995:** This near-perfect score confirms the model's outstanding ability to discriminate between diseased and healthy individuals.

The confusion matrix for the RF model, conceptually visualized in Fig. 5, is dominated by a high True Positive (TP) count and a minimal False Negative (FN) count, affirming its diagnostic reliability.

### *B. Case 2: The Pitfall of Accuracy in Prognosis (Framingham Dataset)*

*1) Scenario:* A public health program aims to identify individuals at long-term (10-year) risk of CHD for preventive

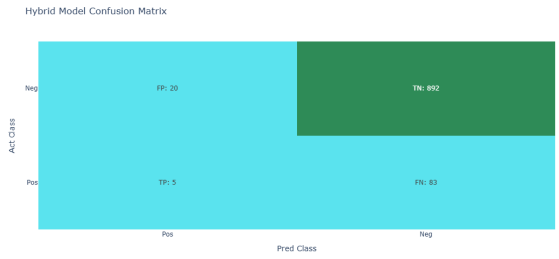


Fig. 6. Conceptual Confusion Matrix for the Hybrid Model on the Framingham Dataset. The large TN count drives high accuracy, while the high FN count reveals the model's clinical failure.

interventions. The prognostic Framingham dataset is used for this screening task.

*2) Analysis:* The results from Experiment 1 (TABLE I) serve as a crucial cautionary tale regarding the use of accuracy as a primary metric in the face of class imbalance.

- **Misleadingly High Accuracy:** All models reported accuracies above 85%, which, if viewed in isolation, would suggest strong performance.
- **Critically Low Recall:** In stark contrast, the recall scores were abysmal. The best model achieved a recall of only 0.0565, meaning it failed to identify over 94% of at-risk individuals.

This discrepancy arises because the models achieved high accuracy by simply predicting the majority class (no risk) for nearly every patient. In a clinical setting, such a model is not merely useless but actively harmful, providing a false sense of security to the very individuals who need intervention. The conceptual confusion matrix for this scenario (Fig. 6) would show a massive True Negative (TN) count driving the high accuracy, while the dangerously high False Negative (FN) count reveals its clinical failure. This case powerfully demonstrates that for imbalanced prognostic tasks, metrics like Recall, F1-Score, and ROC AUC are far more informative than raw accuracy.

## VII. Discussion and Future Work

The experimental results and case studies furnish several critical insights into the application of machine learning for CVD prediction. This section discusses these findings, analyzing the interplay between model architecture, dataset characteristics, and clinical utility, before outlining key directions for future research.

### A. The Critical Impact of Dataset Characteristics

The most salient finding is the dramatic performance disparity between the Framingham and Cleveland datasets. This underscores a core tenet of applied machine learning: no single model is universally superior. An algorithm's efficacy is inextricably linked to the data's nature and the specific problem it addresses. The Cleveland dataset, with its diagnostic focus and relatively balanced classes, presented a well-posed problem that was readily solved by standard classifiers. In contrast, the Framingham dataset represented a far more complex prognostic task compounded by severe class imbalance. This imbalance led the models to adopt a naive strategy of predicting the majority class, resulting in high accuracy but abysmal recall, rendering them clinically inept.

### B. Analysis of Model Performance

*1) The Dominance of Random Forest on Balanced Data:* The exceptional performance of the Random Forest on the Cleveland dataset (95.24% accuracy, 0.995 ROC AUC) stems from its architectural strengths. As a robust ensemble, it effectively models complex, non-linear feature relationships and is inherently resistant to overfitting, making it a powerful and reliable choice for well-posed diagnostic problems.

*2) The Promise of the CNN-LSTM Architecture:* While its raw accuracy on the Cleveland dataset was lower than RF's, the CNN-LSTM model's primary strength was its balanced performance. Its nearly identical precision and recall for both classes (Fig. 3) is a highly desirable property in clinical settings, indicating an unbiased model. The novel approach of treating patient features as a sequence for combined CNN-LSTM processing is a promising direction for tabular clinical data.

### C. Limitations and Future Work

This study, while comprehensive, has limitations that pave the way for future research:

- **Hyperparameter Optimization:** The models were trained with default or basic hyperparameters. Future work should implement systematic tuning using methods like *GridSearchCV* or *Bayesian Optimization* to unlock the full potential of each architecture.
- **Mitigating Class Imbalance:** The poor results on the Framingham dataset highlight the need for explicit strategies to handle imbalance. Future research should explore data-level techniques like *SMOTE* (Synthetic Minority Over-sampling Technique) and algorithm-level methods such as *class weighting* to force the model to focus on the minority class.
- **Enhancing Model Interpretability:** The "black-box" nature of Random Forest and deep learning models is a barrier to clinical adoption. Future work must integrate model-agnostic interpretability tools like *SHAP (SHapley Additive exPlanations)* and *LIME (Local Interpretable Model-agnostic Explanations)* to provide feature-level insights, thereby fostering clinical trust and transparency.
- **Investigating Advanced Architectures:** While the CNN-LSTM showed promise, newer architectures like *Transformers*, which have demonstrated remarkable success in other domains, could be adapted for structured clinical data. Future studies should explore these cutting-edge models to potentially push the boundaries of predictive accuracy.

## VIII. Conclusion

This investigation provided a comprehensive comparative examination of machine learning and deep learning approaches

for predicting cardiovascular disease, generating essential understanding regarding the relationship between model architecture, data properties, and practical clinical application. The primary discovery from this work demonstrates that although Random Forest demonstrates excellence in well-structured diagnostic problems with equilibrated datasets, attaining outstanding accuracy, the innovative CNN-LSTM framework represents an attractive option that delivers more equilibrated performance—a characteristic of critical significance in numerous clinical contexts. Additionally, this investigation presents an important warning, illustrated through the Framingham dataset, that elevated accuracy may serve as a deceptively misleading indicator when class imbalance exists. It emphasizes the imperative of focusing on measures such as recall and F1-score for developing clinically meaningful applications. In summary, this work establishes a comprehensive methodology for model selection and assessment in cardiovascular predictive analytics and argues that subsequent research must transition toward a multidimensional optimization objective that incorporates not only accuracy, but also stability, equity, and clinical explainability.

## REFERENCES


[1] World Health Organization, "Cardiovascular diseases (CVDs)," Jun. 2021. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[2] A. L. Beam and I. S. Kohane, "Big Data and Machine Learning in Health Care," *JAMA*, vol. 319, no. 13, pp. 1317–1318, Apr. 2018.

[3] D. Janosi, A. Andras, and R. Detrano, "UCI Machine Learning Repository: Heart Disease Data Set," 1988. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/heart+disease

[4] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013.

[5] M. A. M. Hasan, M. M. A. B. Nasser, B. Pal, and S. Ahmad, "A comparison of machine learning algorithms for predicting heart disease," in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2019, pp. 1-6.

[6] L. Rokach, "Ensemble-based classifiers," *Artificial intelligence review*, vol. 33, no. 1-2, pp. 1-39, 2010.

[7] J. Lee, J. Yoon, and S. Kim, "Bio-Signal-Based Emotion Recognition Using 1D-CNN and LSTM," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, 2018, pp. 1323-1325.

[8] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-time patient-specific ECG classification by 1-D convolutional neural networks," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 3, pp. 664-675, 2015.

[9] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273-297, 1995.

[10] A. H. Uddin, M. A. Moni, and M. A. H. Khan, "A comparative study of machine learning algorithms for predicting heart disease," in *2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, 2019, pp. 553-558.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[12] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5-32, 2001.

[13] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861-874, 2006.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321-357, 2002.

[15] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in neural information processing systems*, 2017, pp. 4765-4774.

[16] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should I trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135-1144.

[17] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *Journal of machine learning research*, vol. 12, pp. 2825-2830, 2011.

[18] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[19] F. Chollet et al., "Keras," 2015. [Online]. Available: https://keras.io

[20] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, 2010, vol. 445, pp. 51-56.

[21] P. K. Kannel, W. B. Kannel, R. S. Paffenbarger, and T. R. Dawber, "Heart rate and cardiovascular mortality: the Framingham Study," *American heart journal*, vol. 113, no. 6, pp. 1489-1494, 1987.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2017.

[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[24] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281-305, 2012.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[26] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[27] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in science & engineering*, vol. 9, no. 3, pp. 90-95, 2007.

[28] M. L. Waskom, "Seaborn: statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021.

[29] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc., 2019.

[30] R. A. Johnson and D. W. Wichern, *Applied multivariate statistical analysis*. Pearson, 2014.