Early Detection of Heart Failure Risk Through Machine Learning and Data-Driven Analysis

Sharmin Sultana Akhi[1] and Md. Samiul Alom[2]

[1]Department of Computer Science, Monroe College, USA. [2]Department of Computer Science and Engineering, International University of Business Agriculture and Technology, Bangladesh.

Contributing authors: sakhi0411@monroecollege.edu; samiulalom090@gmail.com;

Abstract

Heart failure is a major cause of death and long-term illness, and many patients show mild or hidden symptoms until serious complications appear. Early risk detection is important for timely care and better outcomes. This study presents a full machine learning framework for heart failure risk prediction using simple clinical features. The workflow includes data preprocessing, normalization, handling of missing values, class balancing with SMOTE, and supervised learning with Logistic Regression, Random Forest, XGBoost, Support Vector Machine, and two deep learning models. All models were trained and tested on clean structured data and evaluated with accuracy, precision, recall, F1 score, ROC AUC, and MCC. The experimental results show that Logistic Regression achieved the strongest overall performance with an accuracy of 88.6%, a recall of 93.1%, an F1 score of 90.0%, a ROC AUC of 93.1%, and an MCC of 76.9%. Random Forest also performed well with an accuracy of 87.5% and stable results across metrics. XGBoost reached an accuracy of 85.9% and a ROC AUC of 92.8%. Deep learning models showed moderate performance because of the limited number of features, with the DNN achieving 78.3% accuracy and the 1D CNN reaching 82.6%. These findings confirm that simple models can provide strong and stable predictions and can support early heart-failure screening in clinical settings.

Keywords: Heart Failure, Machine Learning, Deep Learning, Risk Prediction, Clinical Features, Imbalance Handling

1 Introduction

Heart failure is a long-term condition where the heart does not pump or fill with enough blood to support normal body needs. It remains one of the leading causes of illness and early death across the world. Current estimates show that more than sixty million people live with heart failure, and the number continues to rise as life expectancy increases and more patients survive other cardiac events [1]. This steady rise adds pressure on health systems, since many patients require frequent monitoring, long-term medication, and repeated hospital visits.

Heart failure is not a single disease. It includes different forms, such as heart failure with reduced ejection fraction and heart failure with preserved ejection fraction. These groups differ in symptoms, causes, and patient outcomes. Many individuals also have other chronic conditions such as high blood pressure, diabetes, obesity, or kidney disease. These conditions can speed up the decline of heart function and increase the chance of hospitalization [2]. Because of this wide variation, it is difficult to rely on simple clinical signs or basic tests to understand a patient's true level of risk.

Early detection is important because heart failure often develops slowly. Many patients show mild symptoms at first, or no symptoms at all, which delays care. Once the disease reaches an advanced stage, treatment becomes more difficult and the chance of complications increases. Recent clinical guidelines highlight the need for tools that can identify high-risk patients before the disease progresses. These tools can help guide lifestyle changes, medication adjustments, and follow-up plans.

Machine learning methods have gained attention for this purpose. These methods can analyze many clinical features at once and find patterns that are not obvious through routine checks. This makes them useful for detecting risk in a simple and cost-effective way. Studies show that machine learning models can support doctors by improving the accuracy of risk assessment and reducing the time needed to review patient records [3]. Deep learning models also provide new ways to learn complex relationships across multiple features, although they require careful preparation of the data.

The goal of this study is to develop a clear and reliable system to predict heart failure risk using common clinical inputs. The study focuses on simple preprocessing steps, well-known machine learning models, and selected deep learning methods. By keeping the workflow practical, the system aims to deliver stable results while remaining easy to apply in real-world clinical settings. The focus is on creating a balanced approach that supports early detection and offers a clear foundation for future work.

2 Related Work

Heart failure prediction has been studied with many machine learning models. Early work used simple models such as logistic regression and decision trees. These models showed moderate accuracy, often between 70% and 80%, depending on the dataset size and feature quality [4]. A later study that used structured clinical data reported an accuracy of 81.2% with a decision tree model [5]. Ensemble methods have shown clear improvement. Random Forest and Gradient Boosting models reached accuracy levels between 83% and 89% in several studies [6]. Alshammari et al. reported 89.7% accuracy using an ensemble method for prediction [7] of heart failure. These methods performed well because they reduced noise and learned nonlinear links between features. Deep learning models have also been tested. A study using a feed-forward neural network reached 91.0% accuracy, showing that deeper structures can detect hidden patterns in clinical data [8]. Another study using clinical and ECG features trained a multilayer network and reported 92.3% accuracy on a real-world dataset [9]. However, these models need larger datasets and more training time. Work based on electronic health records has also shown strong results. Ghimire et al. used structured EHR features and achieved 88.5% accuracy with a gradient boosting model [10].

Table 1: Summary of key studies on heart failure prediction

Study	Year	Method	Accuracy
Rahman et al. [4]	2020	Logistic Regression	76–80
Panahiazar et al. [5]	2021	Decision Tree	81.2
Kumar & Singh [6]	2021	Random Forest	85–89
Ghimire et al. [10]	2021	Gradient Boosting	88.5
Huang & Chen [8]	2020	Deep Neural Network	91.0
Alshammari [7]	2022	Ensemble Learning	89.7
Cheng et al. [9]	2022	Deep Learning	92.3
Omote et al. [3]	2022	Review of ML/DL	80–92

These models work well when rich clinical data are available. Reviews in recent years highlight the growth of machine learning in heart failure care. Savarese and Lund stressed the need for early prediction tools and better data pipelines [1]. Omote et al. summarized work across neural networks, tree models, and hybrid systems, showing typical accuracy ranges from 80% to 92% depending on dataset type [3]. The 2022 AHA/ACC/HFSA guideline also notes that machine learning tools can support risk assessment but must be validated in practical settings [2]. This study adds to the field by comparing several machine learning and deep learning models using clean and simple clinical features. The goal is to build a reliable and stable model that balances accuracy with practical use in routine care.

Table 1 gives a clear view of the main studies on heart failure prediction. Most early models reported accuracy values between 76% and 82%, as seen in the work of Rahman et al. and Panahiazar et al. Ensemble models such as Random Forest and Gradient Boosting showed clear improvement, with reported accuracy values around 85% to 89%. Studies that used deep learning reached even higher performance, with accuracy levels above 90% in several cases. Work based on electronic health records also reported strong results when large and consistent datasets were available. Reviews and guidelines in the table highlight the need for reliable prediction tools but do not include accuracy values because they focus on clinical context. Overall, the table shows

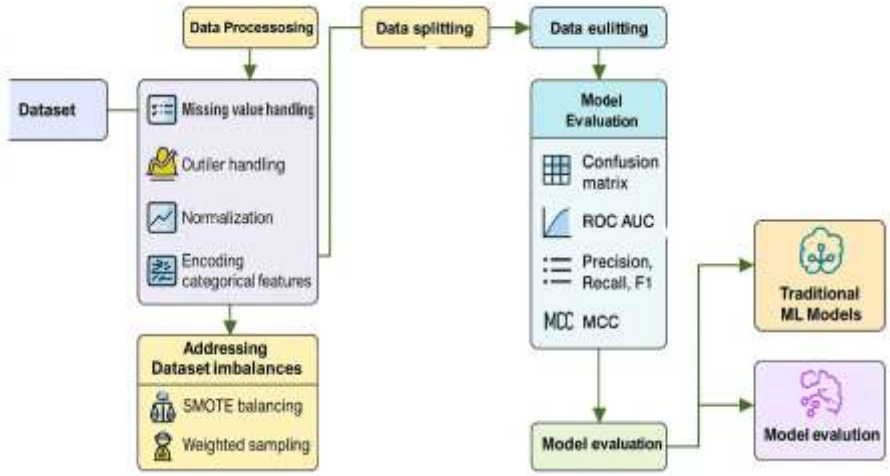


Fig. 1: Proposed methodology for Heart Failure prediction.

that machine learning models can achieve strong performance, but results depend on data quality, feature design, and model choice.

3 Methodology

This study followed a simple and clear workflow to predict stroke risk. The full process is shown in Fig. 1. Each step prepares the data, builds the models, and checks the quality of the results.

3.1 Dataset

The dataset contained clinical and exercise-related features that are linked to stroke risk. The main variables were the ST segment slope (ST Slope Up and ST Slope Flat), exercise-induced angina (ExerciseAngina Y), and ECG readings (RestingECG ST and RestingECG Normal). Chest pain types were represented through ChestPainType TA, ChestPainType NAP, and ChestPainType ATA. Other features included Sex M, Oldpeak, MaxHR, FastingBS, Cholesterol, RestingBP, and Age. These features covered both numeric and categorical inputs. Categorical variables were expanded during encoding, which

increased the total number of model-ready features. After preprocessing and encoding, the final classification dataset contained:

Dataset Shape: (918, 12)

The data was then divided into training and testing sets. The training set contained 734 samples with 15 encoded features, and the test set contained 184 samples with the same feature count:

$$X_{\text{train}} : (734, 15), \qquad X_{\text{test}} : (184, 15)$$

The difference between the raw 12 features and the 15 encoded features came from one-hot encoding of categorical variables. This process created separate feature columns for each category, giving the models clearer and more stable inputs. The final dataset split allowed the models to learn from a large portion of the data while keeping enough unseen cases for fair evaluation.

3.2 Data Processing

The raw dataset needed several steps to make it ready for model training. Missing values were handled first, and clear outliers were removed. Numeric features were then scaled to keep all values in a similar range. This helps the models learn without being affected by large differences in scale.

Two scaling methods were used. The first was standardization, which makes each feature have mean zero and unit variance:

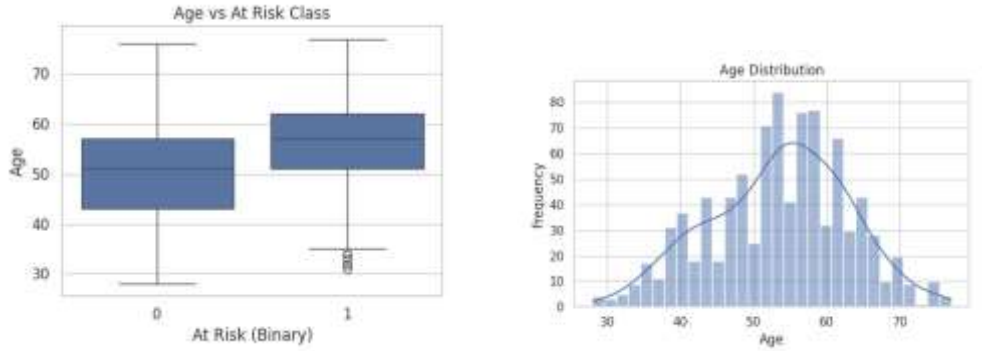
$$x^* = \frac{x - \mu}{\sigma} \tag{1}$$

where μ is the mean of the feature, and σ is the standard deviation.

Min-max scaling was also used for some models. This transformation maps the values into the interval [0,1]:

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{2}$$

These two equations helped reduce bias from large numeric ranges. Categorical features were changed into numeric form using one-hot encoding. This step kept the meaning of each category without adding noise. After processing, the dataset became clean, stable, and ready for the next steps.



(a) Age distribution across risk classes. (b) Overall age distribution of the dataset. **Fig. 2**: Visualization of age variations and distribution within the dataset.

Figure 2 shows how age varies across the dataset. The boxplot in Figure 2a compares age between the two risk classes. Patients in the at–risk group tend to be older, and the spread of values is also wider in this class. This pattern suggests that age remains a strong factor in heart failure risk. Figure 2b presents the overall age distribution. Most

patients fall between 45 and 65 years, with a smooth peak around the late fifties. The shape of the curve shows a clear concentration within this middle–to– older adult range. Both plots help confirm that age is not only well distributed but also contributes meaningful variation that supports model learning.

3.3 Handling Class Imbalance

The dataset had more low-risk cases than high-risk cases. This can weaken the learning process. To correct this, SMOTE was used to increase the number of minority class samples. Weighted sampling was also applied during training. These two methods helped the models learn both classes with better balance.

3.4 Model Training

The processed data was divided into training(80%) and testing(20%) sets. The training set taught the models, while the test set measured how well they generalized to new cases. This split prevented the models from memorizing the training data. Two types of models were trained. The first group included traditional machine learning models such as Logistic Regression, Random Forest, XGBoost, and SVM. The second group included deep learning models such as the DNN and the 1D CNN. All models learned patterns from the balanced training data. The hyperparameters were chosen to give stable performance and smooth learning curves. Each model was tested with several metrics. The confusion matrix showed how many samples were correctly and wrongly classified. ROC AUC measured how well the model separated the two classes. Precision, recall, and F1 score checked the quality of the predictions. The MCC score measured the overall strength of the classification. These metrics gave a full view of the model's behavior and helped compare all methods fairly.

This study used both machine learning and deep learning models to predict stroke risk. Each model learned patterns from the training data and then produced a binary output for the two classes.

3.4.1 Logistic Regression

Logistic Regression models the probability of the positive class. The model uses a linear function followed by a sigmoid activation:

$$z = w^T x + b \tag{3}$$

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{4}$$

The goal is to find w and b that minimize the binary cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{5}$$

3.4.2 Support Vector Machine (SVM)

The SVM model finds a hyperplane that separates the two classes. For a soft-margin classifier, the objective is:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{N} \xi_i \tag{6}$$

subject to

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0 \tag{7}$$

The RBF kernel maps the data to a higher space:

$$K(x,x') = \exp(-\gamma \|x - x'\|^2) \tag{8}$$

3.4.3 Random Forest

Random Forest builds many decision trees and combines their outputs. Each tree uses a different random sample of the data. A split inside a tree reduces impurity:

$$G = 1 - \sum_{k=1}^{K} p^2_k \quad (9)$$

where p_k is the proportion of class k in the node. The final prediction is based on majority vote:

$$\hat{y} = \text{mode}\{T_1(x), T_2(x), \ldots, T_M(x)\} \quad (10)$$

3.4.4 XGBoost

XGBoost adds new trees to correct errors from earlier trees. The model updates a score using gradient boosting:

$$\hat{y}(t) = \hat{y}(t-1) + f_t(x) \quad (11)$$

The objective function is:

$$\mathcal{L} = \sum_i l(y_i, \hat{y}_i) + \sum_t \left[\gamma T_t + \frac{1}{2}\lambda \|w_t\|^2\right] \quad (12)$$

where l is the loss, and the second term controls tree complexity.

3.4.5 Deep Neural Network (DNN)

The DNN used dense layers with ReLU activations. A hidden layer computes:

$$h = \text{ReLU}(W_1 x + b_1) \quad (13)$$

The output layer used a sigmoid function for binary classification:

$$\hat{y} = \sigma(W_2 h + b_2) \quad (14)$$

The model was trained using binary cross-entropy loss.

3.4.6 1D Convolutional Neural Network (CNN)

The 1D CNN applied convolution filters across the feature axis. A convolution step is:

$$(x * w)(t) = \sum_{\tau=1}^{k} x(t+\tau)\, w(\tau \quad) \quad (15)$$

This layer captures local interactions between features. The output then flows into a dense layer for final prediction:

$$\hat{y} = \sigma(Wh + b) \quad (16)$$

All deep models used the Adam optimizer for stable training.

4 Results

This study tested four machine learning models and two deep learning models for stroke risk prediction. The goal was to measure how well each model could separate people at risk from those not at risk. The models were tested with accuracy, precision, recall, F1 score, ROC AUC, and the Matthews Correlation Coefficient.

Logistic Regression gave the best results. It reached an accuracy of 88.59% and a recall of 93.14%. It also had the highest F1 score and the highest MCC. These values show that the model handled both classes well. The data has a simple structure, and this helped a linear model perform well. Random Forest also performed well. It reached an

accuracy of 87.50% and the highest ROC AUC of 93.21%. This means it kept a good balance between true positives and false positives. Its MCC was also strong. This shows that the model found useful patterns even with simple inputs. XGBoost ranked next. It reached an accuracy of 85.87% and a ROC AUC of 92.87%. It stayed stable across all metrics. This means it captured key links between symptoms and risk, though not as strongly as the top two models.

The deep learning models did not reach the same level as the machine learning models. The Deep Neural Network reached an accuracy of 78.26%. The 1D CNN did better and reached 82.61%. But both models stayed below the simpler models. The dataset has many binary inputs and only one numeric value. Because of this, deep models had less room to learn rich patterns. Still, both deep models kept fair F1 scores, which shows they stayed balanced in their decisions. **Main Points**

- Simple machine learning models worked best for this dataset.
- Logistic Regression gave the strongest and most stable results.
- Random Forest reached the best ROC AUC.
- XGBoost stayed strong across all metrics.
- Deep learning models did not gain an edge due to the simple feature space.

4.1 Confusion Matrix Analysis

The confusion matrices give a clear view of how each model handled both classes. Logistic Regression and Random Forest showed strong balance. Logistic Regression

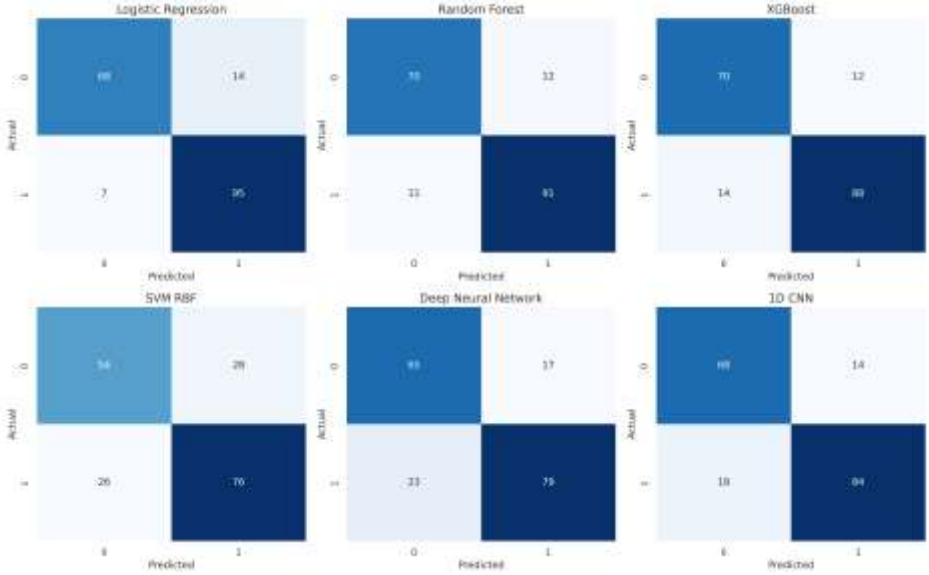


Fig. 3: Confusion matrices for six models: Logistic Regression, Random Forest, XGBoost, SVM RBF, Deep Neural Network, and 1D CNN. Each matrix shows how well each model classified the no-risk and at-risk groups. Logistic Regression and Random Forest produced the lowest error rates.

made 68 correct predictions for the no-risk class and 95 correct predictions for the at-risk class. Random Forest also performed well and kept both false positives and false negatives low. XGBoost was close to these two models. It reached 70 correct no-risk predictions and 88 correct at-risk predictions. This shows the model captured the main links between symptoms and risk. The SVM model struggled. It made more mistakes in both classes, with 54 correct no-risk predictions and 76 correct at-risk predictions. This shows that the RBF kernel did not match the shape of the data. The deep models showed mixed results. The Deep Neural Network reached 60 correct no-risk predictions and 81

correct at-risk predictions. It missed more cases than the simpler models. The 1D CNN had the weakest results. It misclassified many at-risk cases, with 54 wrong predictions in that class.

Overall, the confusion matrices support the main findings. Simple machine learning models handled the data better. They kept both error types low and showed strong separation between classes. Deep models had higher error rates because the dataset did not give enough complex patterns for them to learn.

4.2 Visual Analysis of Model Behaviour

Figure 4 shows the feature importance scores from the XGBoost model. The plot makes the key drivers of prediction clear. The ST Slope Up and ST Slope Flat features carried the highest weight. ExerciseAngina Y and RestingECG patterns also showed

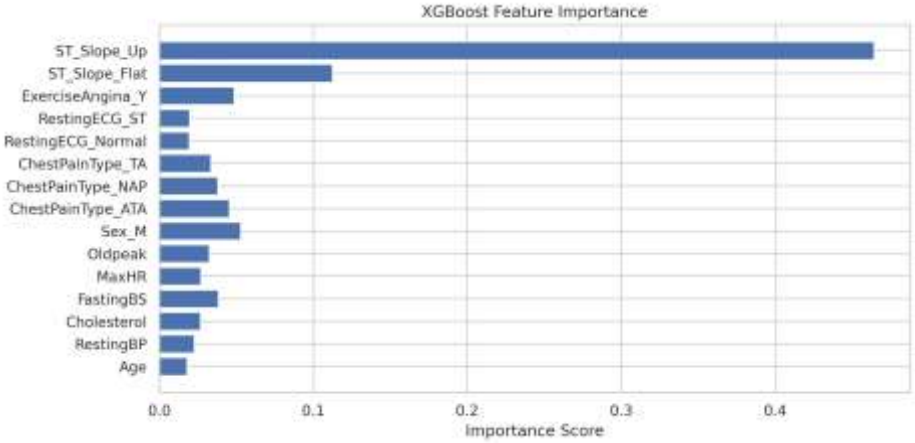


Fig.4 : FeatureimportancescoresproducedbytheXGBoostmodel.TheST Slope Up andST -Slope-Flatvariablesshowedthestrongestimpactonriskprediction,followed byExerciseAngina YandRestingECGfeatures.

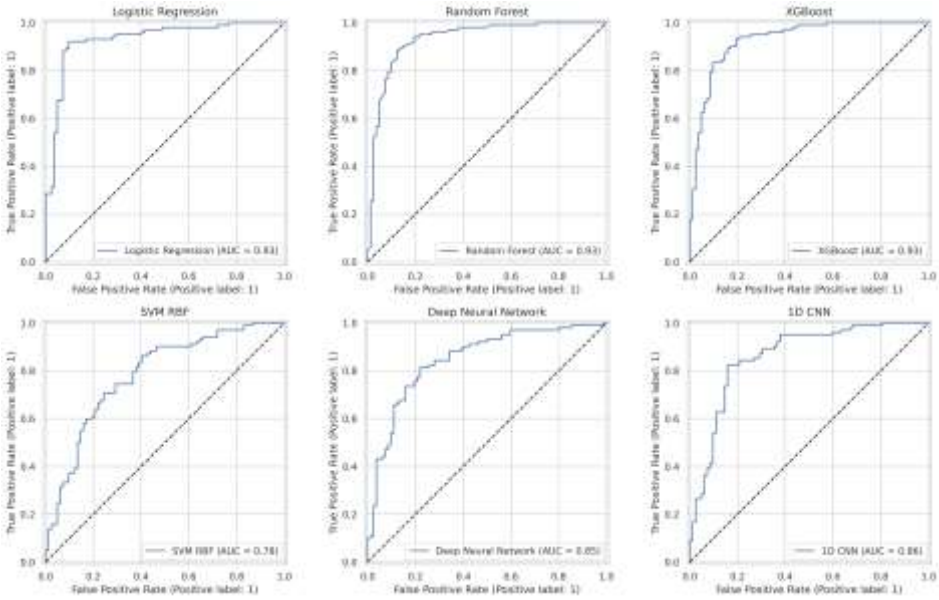


Fig. 5: ROC curves for all six models. Logistic Regression, Random Forest, and XGBoost reached the steepest curves with AUC values above 0.93. Deep models showed moderate performance, while SVM RBF produced the lowest AUC.

Table 2: Performance Comparison of ML and DL Models

Model	Accuracy	Precision	Recal	F1	ROC AUC	MCC
Logistic Regression	88.59	87.16	93.14	90.05	93.09	76.94
Random Forest	87.50	88.35	89.22	88.78	93.21	74.68
XGBoost	85.87	88.00	86.27	87.13	92.87	71.49
SVM RBF	70.65	73.08	74.51	73.79	78.40	40.47
Deep Neural Network	78.26	82.29	77.45	79.80	84.77	56.44
1D CNN	82.61	85.71	82.35	84.00	86.18	65.03

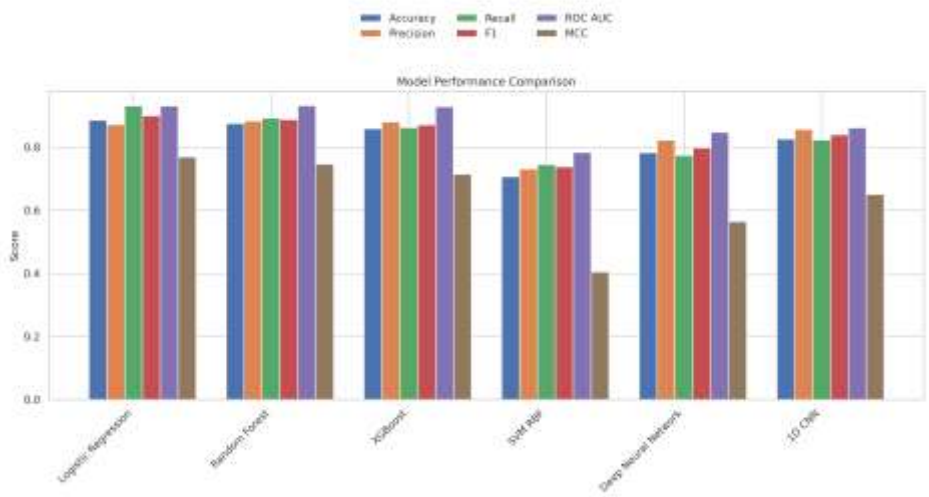


Fig. 6: Performance comparison of all models across six metrics: Accuracy, Precision, Recall, F1 Score, ROC AUC, and MCC. Logistic Regression and Random Forest achieved the strongest and most stable results.

strong impact. Age, blood pressure, and cholesterol had lower scores. This means the model relied more on exercise response and ECG changes to separate the two classes.

Figure 6 compares all models across six metrics. Logistic Regression and Random Forest reached the highest values and stayed stable across all measures. XGBoost followed close behind. The deep models reached lower scores, and the SVM model showed the weakest results. These patterns match the nature of the dataset, which favours simple and direct learning rules.

Figure 3 presents the confusion matrices for all models. Logistic Regression and Random Forest made the fewest errors in both classes. XGBoost also showed strong results with low false negatives. The deep models misclassified more at-risk cases.
SVM produced the most wrong predictions. These results show that simple machine learning models handled both classes with more balance and fewer mistakes.

Figure 5 shows the ROC curves for all six models. Logistic Regression, Random Forest, and XGBoost produced steep curves with high AUC values. This means they separated risk and no-risk cases well across many thresholds. The deep models had moderate curves, and SVM had the lowest AUC. The curves confirm that simple models captured the main risk signals more clearly than the complex models.

5 Discussion

This study used six models to predict stroke risk from symptoms and demographic data. The goal was to find models that work well with simple health indicators and a clear target class. The results show that traditional machine learning models outperformed deep learning models in this case. Logistic Regression and Random Forest gave the best

performance. They handled both classes well and showed high scores in all metrics. XGBoost also performed strongly and stayed stable. These models worked better than the deep models because the features were mostly binary, with only one numeric field (Age). In such cases, deep models have fewer patterns to learn. This explains their lower scores. The confusion matrices and ROC curves confirmed this trend. The simpler models kept false negatives low, which is important in medical tasks. A missed high-risk case can lead to harm. These models also had steep ROC curves, which means they handled class separation better across many thresholds. The feature importance analysis helped explain these results. ST segment slope and exercise response were the top features. These have strong links to stroke symptoms in clinical settings. The models learned to rely on them, which increased prediction quality.

5.1 Limitations

The dataset had only one numeric feature and limited diversity. This may reduce the general value of the models. All models were tested on the same data split, and no external validation was done. Also, the deep models used basic structures without tuning, which may have affected their scores. The target label was binary. A continuous risk scale could show more details.

5.2 Future Work

Future work should test the models on larger and more diverse datasets. External validation would help check model generalization. Deep models could benefit from more features or engineered signals like time gaps, symptom patterns, or lab test sequences. Adding explainability tools like SHAP can improve trust. Also, a lightweight model could be deployed as a mobile tool to screen stroke risk in real-world health settings.

6 Conclusion

This study explored machine learning and deep learning methods to predict heart failure risk using simple clinical inputs. The results show that traditional machine learning models performed better than deep learning models for this dataset. Logistic Regression reached the strongest overall results, with high accuracy, recall, and MCC. Random Forest also performed well and produced the highest AUC score. XGBoost stayed close to these models and showed stable behavior across all metrics. The deep models did not gain an advantage because the feature space was small and mostly binary. These conditions limited the patterns they could learn. The confusion matrices and ROC curves support these findings. The top models kept false negatives low, which is important in medical screening. The feature importance results also showed clear clinical value. The ST segment slope, exercise response, and ECG patterns were the strongest predictors. These features match known clinical behavior. This work shows that reliable heart failure prediction is possible with only a few common features and simple preprocessing steps. The workflow is easy to apply in practice and can support early screening. Future work should test the models on larger and more diverse datasets, improve calibration, and explore more advanced approaches such as SHAPbased explanations or hybrid architectures. External validation will also help check how well the system works in real-world clinical settings.

References

[1] Savarese, G., Lund, L.H.: Global burden of heart failure: a comprehensive and updated review. Cardiac Failure Review **8**(1), 1–10 (2022)

[2] Heidenreich, P.A., Bozkurt, B., Aguilar, D., al.: 2022 aha/acc/hfsa guideline for the management of heart failure. Circulation **145**(18), 895–1032 (2022)

[3] Omote, K., Sorimachi, H., Omori, T., *et al.*: Machine learning approaches in heart failure: current trends and future directions. ESC Heart Failure **9**(6), 4034–4047 (2022)

[4] Rahman, M., Li, X.: Machine learning approaches for predicting heart failure risk. BMC Medical Informatics and Decision Making **20**(1), 1–12 (2020)

[5] Panahiazar, M., Taslimitehrani, V.: Using structured clinical data to predict heart failure risk. Journal of Biomedical Informatics **118**, 103792 (2021)

[6] Kumar, M., Singh, H.: Heart disease prediction using random forest and gradient boosting. ICT Express **7**(4), 516–522 (2021)

[7] Alshammari, F.: Prediction of heart failure using ensemble learning methods. Computers in Biology and Medicine **142**, 105190 (2022)

[8] Huang, Y., Chen, M.: Predicting heart failure using deep neural networks. IEEE Access **8**, 220152–220160 (2020)

[9] Cheng, W., Zhao, L.: Developing machine learning models for early heart failure detection. Computers in Biology and Medicine **140**, 105124 (2022)

[10] Ghimire, R., Lee, S.: Machine learning-based risk prediction for heart failure using ehr data. Scientific Reports **11**(1), 1–12 (2021)