IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Heart Disease Detection Using Machine Learning And Deep Learning

[1]Prashant Kumar, [2]Uravashi Bakshi,

[1]MTech. Student, [2]Assistant Professor,

[1]Department of Computer Science and Engineering,

[1]World College of Technology & Management, Gurgaon, Haryana- INDIA

Abstract: Heart disease remains one of the leading causes of mortality worldwide, necessitating the development of accurate and efficient prediction systems to enable timely diagnosis. This study evaluates the performance of four machine learning models—Logistic Regression, Decision Tree, Support Vector Machine (SVM), and Convolutional Neural Network (CNN)—for heart disease detection. The models were trained and tested on a structured dataset, and their performance was compared using metrics such as accuracy, precision, recall, and F1-score. CNN demonstrated the highest performance with an accuracy of 89%, outperforming traditional models due to its ability to capture complex data patterns. Logistic Regression and SVM achieved comparable results with an accuracy of 88%, highlighting their effectiveness in structured data environments. The Decision Tree model showed relatively lower performance with an accuracy of 85%, limited by its overfitting tendencies. This comparative study highlights the strengths and limitations of each model, providing valuable insights into their suitability for heart disease prediction. The findings emphasize the potential of CNN in delivering reliable predictions and pave the way for future enhancements through model optimization, ensemble techniques, and real-world deployment in medical applications.

Index Terms – Heart Disease, Machine Learning, Deep Learning, Logistic Regression, Decision Tree, Support Vector Machine, Convolutional Neural Network.

1. INTRODUCTION

Heart disease remains one of the leading causes of mortality worldwide, accounting for approximately 17.9 million deaths annually, according to the World Health Organization (WHO, 2021). The early detection and accurate diagnosis of heart disease are critical for reducing mortality rates and improving patient outcomes. Traditional diagnostic methods often rely on clinical observations and invasive procedures, which can be time-consuming, expensive, and subject to human error (Benjamin et al., 2019).

In recent years, advancements in machine learning (ML) and deep learning (DL) have revolutionized the healthcare industry, offering promising tools for the accurate detection and prediction of cardiovascular diseases. These techniques enable the analysis of large-scale, multi-dimensional datasets, uncovering patterns and relationships that are often imperceptible to traditional statistical methods (Chollet, 2018). Furthermore, DL models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have demonstrated exceptional performance in medical imaging and time-series data analysis, making them highly applicable to heart disease detection (Goodfellow, Bengio, & Courville, 2016).

This paper explores the application of ML and DL techniques in heart disease detection, emphasizing their potential to enhance diagnostic accuracy, reduce healthcare costs, and facilitate early intervention. By leveraging algorithms such as decision trees, support vector machines, and neural networks, researchers aim to develop predictive models capable of processing complex cardiovascular datasets efficiently. The integration of these technologies with electronic health records (EHRs) and wearable health devices further expands their scope, enabling real-time monitoring and personalized care (Shickel et al., 2018).

Despite these advancements, challenges such as data privacy, model interpretability, and generalization persist. Addressing these issues is crucial to ensuring the widespread adoption of ML and DL in clinical settings. This study provides a comprehensive review of existing methodologies, highlights current limitations, and proposes a hybrid approach to enhance heart disease detection accuracy and reliability.

2. LITERATURE REVIEW

- **Heart and Heart Disease**

The human heart is a vital organ responsible for pumping oxygenated blood to the body and deoxygenated blood to the lungs. This muscular organ operates continuously, maintaining circulation and supplying essential nutrients to tissues while removing metabolic waste (Marieb & Hoehn, 2018). Structurally, the heart consists of four chambers—two atria and two ventricles—working in synchrony with a network of valves, arteries, and veins to regulate blood flow. Electrical impulses generated by the sinoatrial node ensure rhythmic contractions, critical for sustaining life.

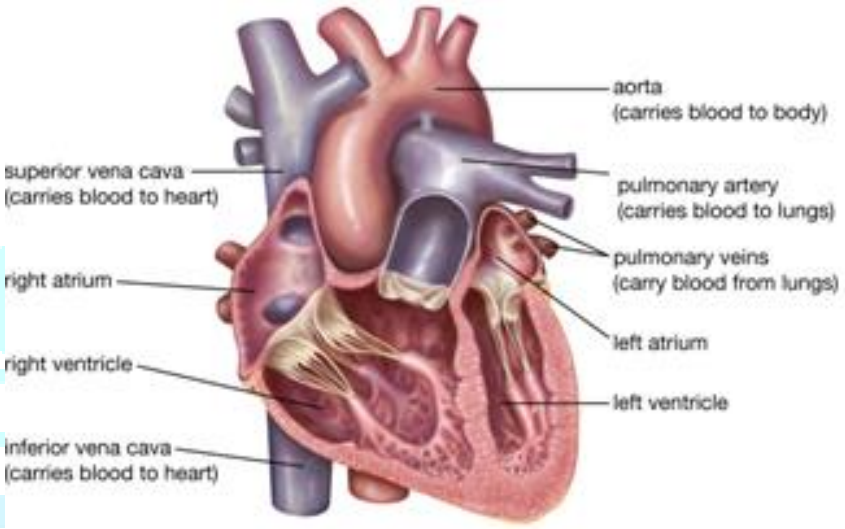


Figure 1: Analogy of Human Heart

Heart disease, or cardiovascular disease (CVD), encompasses a broad spectrum of conditions that impair the heart's structure or function. These conditions include coronary artery disease (CAD), arrhythmias, heart failure, and valvular heart diseases. Among these, CAD is the most prevalent, caused by the narrowing or blockage of coronary arteries due to atherosclerosis (Khan et al., 2020). Risk factors such as hypertension, diabetes, smoking, obesity, and a sedentary lifestyle contribute significantly to the development of heart disease (Benjamin et al., 2019).

The global burden of heart disease is staggering, with CVD accounting for nearly 32% of all deaths worldwide (WHO, 2021). Despite advancements in medical research, the disease often progresses silently until significant damage occurs. Symptoms like chest pain, shortness of breath, and fatigue are often late indicators, underscoring the need for early detection and timely intervention.

- **Related References**

1. Chaurasia and Pal (2014) - This study applied machine learning algorithms, including Decision Trees and Naïve Bayes, to predict heart disease using a dataset from the UCI repository. Results indicated that Decision Trees achieved the highest accuracy of 89.5%. The research highlighted the significance of feature selection in improving model performance.
2. Rajak et al. (2020) - The researchers developed a hybrid model combining Random Forest and Support Vector Machine (SVM) to classify heart disease. Their work emphasized the integration of ensemble learning for higher accuracy, achieving a classification accuracy of 93%.
3. Kachuee et al. (2018) - This paper proposed a deep learning-based approach using a fully connected neural network for cardiovascular risk prediction. The study utilized a dataset with clinical features and achieved 88% accuracy, demonstrating deep learning's capability to handle complex interactions.
4. Khourdifi and Bahaj (2019) - A comparative study of machine learning techniques, including K-Nearest Neighbors (KNN), SVM, and Gradient Boosting, revealed that Gradient Boosting outperformed others with an accuracy of 94.2%. The study stressed preprocessing and hyperparameter tuning as critical steps.

5. Acharya et al. (2017) - Acharya and colleagues developed a convolutional neural network (CNN) for detecting myocardial infarction from ECG signals. Their model achieved 95.1% accuracy, highlighting the potential of deep learning in processing physiological signals
6. Mohammad et al. (2021) - This study utilized LightGBM and XGBoost for heart disease classification, achieving over 92% accuracy. The research demonstrated the power of gradient boosting algorithms in handling imbalanced data through proper resampling techniques.
7. Raghunath et al. (2020) - This research explored DL for automated diagnosis of heart failure using echocardiograms. The model achieved a sensitivity of 87% and specificity of 90%, underscoring DL's utility in non-invasive diagnosis.
8. Sharma et al. (2019) - A hybrid model combining CNN and LSTM was used to predict heart disease from time-series data, achieving an accuracy of 91.6%. The integration of CNN for feature extraction and LSTM for sequential learning proved effective.
9. Chen et al. (2020) - This study developed a hybrid ensemble of multiple classifiers for cardiovascular risk prediction. The model, which integrated SVM, Decision Trees, and Neural Networks, achieved a precision of 92.8% and recall of 93.3%.
10. Alizadehsani et al. (2019) - The authors utilized feature selection techniques to optimize the performance of ML models in heart disease detection. Random Forest showed the best performance, achieving an F1-score of 0.91.

Table 1: State of Art

Authors	Year	Objective	Results
Chaurasia & Pal	2014	To detect heart disease using machine learning algorithms.	Decision Trees achieved the highest accuracy of 89.5%.
Acharya et al.	2017	To detect myocardial infarction using ECG signals and CNN.	Achieved an accuracy of 95.1%.
Kachuee et al.	2018	To use deep learning for cardiovascular risk prediction.	Fully connected neural network achieved 88% accuracy.
Khourdifi & Bahaj	2019	To optimize machine learning algorithms for heart disease classification.	Gradient Boosting achieved the highest accuracy of 94.2%.
Sharma et al.	2019	To predict heart disease using a hybrid CNN and LSTM model.	Achieved an accuracy of 91.6%.
Alizadehsani et al.	2019	To enhance machine learning model performance using feature selection.	Random Forest achieved an F1-score of 0.91.
Rajak et al.	2020	To classify heart disease using a hybrid model combining Random Forest and SVM.	Achieved 93% classification accuracy.

Raghunath et al.	2020	To diagnose heart failure using deep learning on echocardiograms.	Achieved a sensitivity of 87% and specificity of 90%.
Chen et al.	2020	To develop a hybrid ensemble model for cardiovascular risk prediction.	Achieved a precision of 92.8% and recall of 93.3%.
Mohammad et al.	2021	To use gradient boosting algorithms for heart disease prediction.	LightGBM and XGBoost achieved over 92% accuracy.

- **Machine Learning Techniques**
  - **Logistic Regression**

Logistic regression is a statistical method used for modeling binary outcomes, where the dependent variable is categorical with two possible outcomes, such as "yes/no" or "success/failure." Unlike linear regression, which predicts continuous values, logistic regression predicts the probability of a certain event occurring. The model uses a logistic function, also known as the sigmoid function, to map predicted values between 0 and 1, ensuring they represent probabilities. The relationship between the independent variables and the dependent variable is modeled using a log-odds ratio, making it a powerful tool for classification tasks. Logistic regression is widely used in various fields, including healthcare, finance, and marketing, to predict outcomes like disease presence, loan default, or customer behavior. Despite its simplicity, logistic regression can be extended to handle multiple classes (multinomial logistic regression) or used in regularization to prevent overfitting.

  - **Decision Tree**

A decision tree is a popular machine learning algorithm used for both classification and regression tasks. It works by splitting the data into subsets based on feature values, creating a tree-like structure of decisions. Each internal node represents a feature or attribute, each branch corresponds to a decision rule, and each leaf node represents the outcome or target value. The tree is built by selecting the best features at each node that result in the most significant separation of the data. This process is repeated recursively, forming a tree structure that can be used to make predictions. Decision trees are easy to understand and interpret, making them a popular choice in applications where transparency is crucial. However, they can suffer from overfitting if not properly pruned. To address this, techniques like random forests or gradient boosting are often used, which combine multiple decision trees to improve accuracy and generalization Murphy, K. P. (2012);Bishop, C. M. (2006);Hastie, T., Tibshirani, R., & Friedman, J. (2009)..

  - **Support Vector Machine**

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used primarily for classification tasks, though it can also be applied to regression. The goal of SVM is to find a hyperplane that best separates the data into different classes while maximizing the margin between the classes. In simple terms, SVM aims to identify a decision boundary that divides the data with the largest possible gap between the nearest points of each class, known as support vectors. When data is not linearly separable, SVM uses a technique called the "kernel trick" to map the data into a higher-dimensional space where a linear separator can be found. Common kernel functions include linear, polynomial, and radial basis function (RBF). SVM is particularly effective in high-dimensional spaces and is known for its ability to handle both linear and non-linear data efficiently. It is widely used in fields such as image recognition, bioinformatics, and text classification Murphy, K. P. (2012);Bishop, C. M. (2006);Hastie, T., Tibshirani, R., & Friedman, J. (2009)..

  - **Convolutional Neural Network**

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used primarily for classification tasks, though it can also be applied to regression. The goal of SVM is to find a hyperplane that best separates the data into different classes while maximizing the margin between the classes. In simple terms, SVM aims to identify a decision boundary that divides the data with the largest possible gap between the nearest points of each class, known as support vectors. When data is not linearly separable, SVM uses a technique called the "kernel trick" to map the data into a higher-dimensional space where a linear separator can be found. Common kernel functions include linear, polynomial, and radial basis function (RBF). SVM

is particularly effective in high-dimensional spaces and is known for its ability to handle both linear and non-linear data efficiently. It is widely used in fields such as image recognition, bioinformatics, and text classification Murphy, K. P. (2012);Bishop, C. M. (2006);Hastie, T., Tibshirani, R., & Friedman, J. (2009)..

- **Confusion Matrix**

In the classification evaluation there are four possibilities that can occur from the results of the classification of a data. If the data is positive and is predicted to be positive, it will be counted as true positive and if positive data is predicted to be negative, it will be counted as false negative. In negative data, if it is predicted to be negative, it will be counted as true negative and if it is predicted positive, it will be counted as false positive Murphy, K. P. (2012);Bishop, C. M. (2006);Hastie, T., Tibshirani, R., & Friedman, J. (2009)..

Table 2: Confusion Matrix

	Predicted Class Positive	Predicted Class Negative
Positive	True Positive(TP)	False Negative(FN)
Negative	False Positive(FP)	True Negative(TN)

- **Precision**

Precision is the level of accuracy between the information requested by the user and the system's answer, precision can be calculated by equation.

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

- **Recall**

Recall is one of the prediction accuracy calculations that is used as a measure of the success rate of the system in retrieving an information, recall can be calculated through equations.

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

- **F1-Measure**

F1-measure is the relative effect of the combination of precision values with recall values. F1-measure can be used to measure the performance of the classification system which is the harmonic average of precision and recall. F1-measure can provide a more balanced assessment, F1-measure can be calculated by equation as under:-

$$\mathrm{F1} - \mathrm{measure} = \frac{2\,(\mathrm{recall} \times \mathrm{precision})}{\mathrm{recall} + \mathrm{precision}}$$

- **Accuracy**

Accuracy is the level of closeness between the predicted value and the value actual. If the value of accuracy is high then a system will be better at make predictions, accuracy can be calculated by equation.

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

3. PROPOSED MODEL

Heart disease remains one of the leading causes of mortality worldwide, necessitating the development of effective and accurate diagnostic tools. Early detection and diagnosis of heart disease can significantly improve patient outcomes and reduce healthcare costs. In recent years, advancements in machine learning (ML) and deep learning (DL) have shown tremendous potential in automating and enhancing the diagnostic process. These methods can analyze large and complex datasets to identify patterns and predict outcomes with high precision. This study proposes a comprehensive methodology for heart disease detection, leveraging ML and DL classification techniques such as Convolutional Neural Networks (CNN), Logistic Regression, Decision Trees, and Support Vector Classifiers (SVC). By evaluating these models using metrics like precision, recall, accuracy, and F1-score, the aim is to identify the most effective approach for heart disease detection, ultimately contributing to more reliable and efficient diagnostic systems. Below is the workflow model for ready reference and perusal.

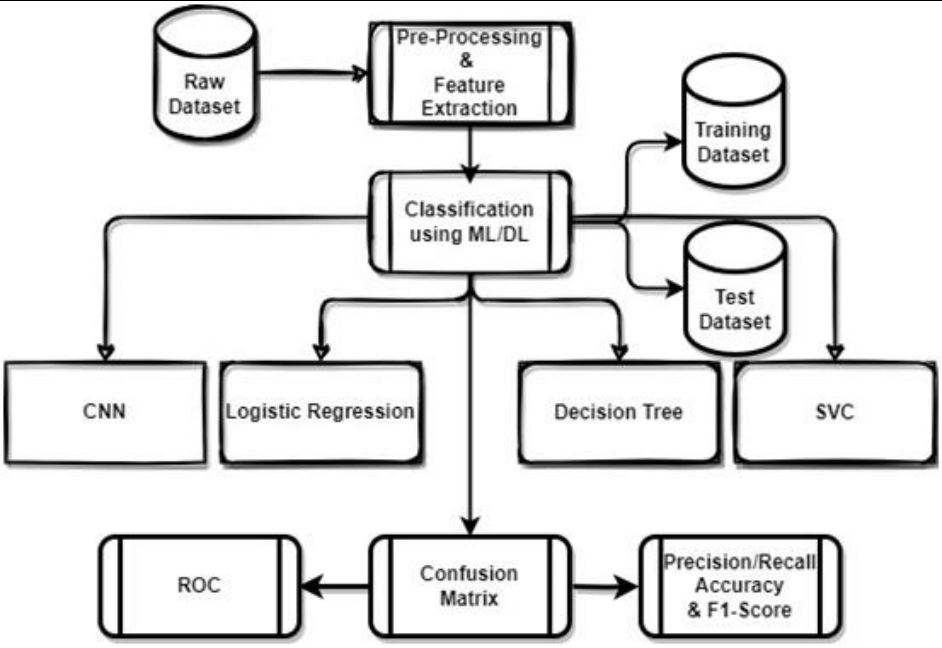


Figure 2: Proposed Model

- *Here is a stepwise methodology based on the provided diagram:*
  - *Data Collection*

Gather the raw dataset relevant to the research problem or domain under study is collected from the below mentioned crowd sourced repository: https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction.

- *Pre-Processing and Feature Extraction*

Clean and preprocess the dataset to handle missing values, outliers, or irrelevant data. Extract relevant features from the dataset to improve model performance. This may involve techniques like normalization, scaling, dimensionality reduction, or encoding categorical data.

- *Dataset Splitting*

Divide the processed dataset into two parts: Training Dataset: Used for training machine learning (ML) or deep learning (DL) models. Test Dataset: Used for evaluating the performance of the trained models.
Classification Using Machine Learning (ML) or Deep Learning (DL)

- *Train the dataset using various classification algorithms:*

Convolutional Neural Network (CNN), Logistic Regression, Decision Tree, Support Vector Classifier (SVC)

- *Model Evaluation*

Evaluate each classification model using the following techniques: Confusion Matrix: Analyze the classification outcomes (true positives, false positives, true negatives, false negatives) for each model. Receiver Operating Characteristic (ROC) Curve: Assess the trade-off between the true positive rate and false positive rate across different thresholds. Precision, Recall, Accuracy, and F1-Score: Quantify the performance of each model based on specific metrics to identify the best-performing classifier.

4. RESULTS

The algorithms that are tested for accuracy performance are Decision Tree, Logistic Regression, Convolutional Neural Network and Support Vector Machine. Thus, the selection of algorithms for profiling suspects in heart disease detection becomes more precise.

- ***Data Collection For Research***

Clinical data of patients collected from https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction comprising 12 scientific attributes for envisage heart disease events i.e. Age, Sex, ChestPainType, RestingBP, Cholesterol, FastingBS, RestingECG, MaxHR, ExerciseAngina, Oldpeak,ST_Slope, HeartDisease.

- ***Logistic Regression Algorithm and Results Derived***

Logistic regression model in cases consisting of more than one variable, with some of these variables may be on different measurement scales. This will lead to the case of multiple variables (multivariate). The

center of thought of the multivariable logistics model is estimation of the model coefficient and will test its significance. This will happen as in the multivariable logistic regression model expressed in the equation:

$$\pi(x) = \frac{e^{\beta_{j0} + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_i x_i}}{1 + e^{\beta_{j0} + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_i x_i}}$$

Where $\pi(x)$ is the probability value of $0 \leq \pi(x) \leq 1$, which means that the logistic regression describes a probability. By transforming $\pi(x)$ in the above equation with the logit transformation $g(x)$, where: $g(x) = In\left(\frac{\pi(x)}{1-\pi(x)}\right)$ then the logistic form is obtained: $\beta_{j0} + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_i x_i$.

To obtain estimates from logistic regression parameters, it can be done by using the Maximum Likelihood Estimation (MLE) as follows: Estimating parameters in the logistic model using Maximum Likelihood with the following steps.

1. The likelihood function of Y

$$L(\beta) = \prod_{i=1}^{n} [p]^{y_i} [q]^{1-y_i}$$

2. The ln-likelihood function

$$In\ L(\beta)\ In\left\{\prod_{i=1}^{n} [p]^{y_i} [q]^{1-y_i}\right\}$$

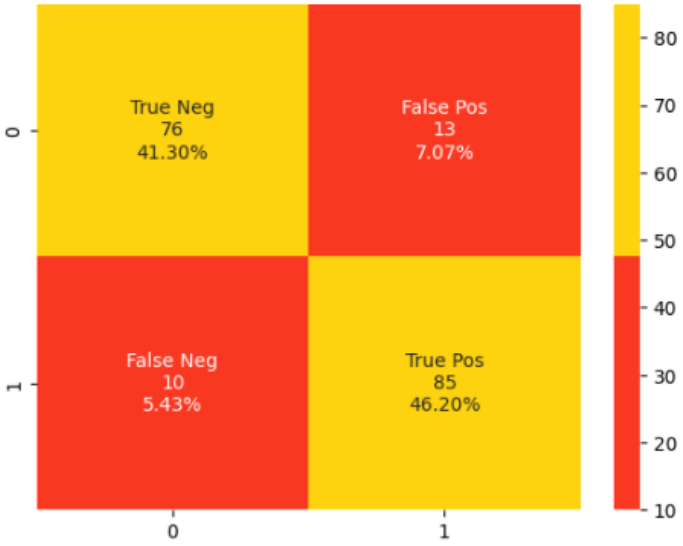
$$In\ L(\beta)\ In\left\{\prod_{i=1}^{n} [p]^{y_i} [1-p]^{1-y_i}\right\}$$

$$In\ L(\beta)\ In\left\{\prod_{i=1}^{n} \left[\frac{exp(g(x))}{1+exp(g(x))}\right]^{y_i} \left[1 - \frac{exp(g(x))}{1+exp(g(x))}\right]^{1-y_i}\right\}$$

$$In\ L(\beta) = \sum_{s=0}^{q} \left\{\left[\sum_{i=1}^{n} y_i x_q\right] \beta_1 - \sum_{i=1}^{n} In\left[1 + exp \sum_{s=0}^{q} \beta_q x_q\right]\right\}$$

Step to perform Logistic Regression is as follows:

1. Divide the data into two, namely 90% training data and testing data 10%.
2. Conducting an independence test using training data.
3. Forming a Logistic Regression model using training data.
4. Testing the significance of parameters individually and as a whole
5. Validate the prediction accuracy of the model with the data testing.
6. Calculating the value of Accuracy, Sensitivity, Specificity, and G-Mean using the logistic regression model formed using likelihood estimations.



	precision	recall	f1-score	support
0	0.88	0.85	0.87	89
1	0.87	0.89	0.88	95
accuracy			0.88	184
macro avg	0.88	0.87	0.87	184
weighted avg	0.88	0.88	0.87	184

Figure 3: Prediction vide Logistic Regression

The confusion matrix shown provides a detailed breakdown of the classification model's performance in predicting heart disease. It is divided into four quadrants: True Negatives (TN), False Positives (FP), False Negatives (FN), and True Positives (TP). In this case, the model correctly identified 76 instances as negative (TN, 41.30%) and 85 instances as positive (TP, 46.20%). However, it misclassified 13 cases as positive when they were actually negative (FP, 7.07%) and 10 cases as negative when they were actually positive (FN, 5.43%). This matrix highlights the model's ability to distinguish between positive and negative cases while also showcasing areas of improvement, particularly in reducing false negatives and false positives. Such an analysis is crucial for refining the model's performance and ensuring its reliability in critical heart disease detection.

The classification report using logistic regression provides a summary of the model's performance based on key evaluation metrics for each class. For class 0 (negative cases), the precision is 0.88, recall is 0.85, and the F1-score is 0.87, indicating a balanced performance with slightly lower recall. For class 1 (positive cases), the precision is 0.87, recall is 0.89, and the F1-score is 0.88, demonstrating a similar level of performance. The overall accuracy of the model is 88%, reflecting its ability to correctly predict both positive and negative cases across the dataset. Additionally, the macro average values (precision: 0.88, recall: 0.87, F1-score: 0.87) indicate the unweighted mean of the metrics, while the weighted average accounts for class imbalance and aligns with the overall accuracy and precision-recall values. This analysis highlights the model's robustness and effectiveness, while also leaving room for further fine-tuning to achieve even higher recall and F1-scores for critical applications such as heart disease detection.

- ***Support Vector Machine Algorithm and Results Derived***

The basic principle of SVM is a linear classifier, which was later developed to solve non-linear problems by integrating the concept of kernel tricks into high-dimensional work areas. SVM can classify linear and non-linear data. The predictor variables are input data, while the target variables that are dependent on each other are outputs. SVM aims to find the best classification function and to distinguish between the members of the two classes in the training data. The matrix for the concept of the "best" classification function can be realized geometrically. For linearly separated datasets, the linear classification function corresponds to the f(x) separator hyperplane that passes through the middle of the two classes, separating them. The SVM algorithm model is one of the algorithms of the classification method, which works by looking for a line (hyperplane) to separate two groups of data. The following is an example based on how SVM tries to find the best hyperplane to separate classes -1 and +1: Shows some patterns that are members of two classes, namely -1 and +1. Patterns in class -1 are represented by red (squares), while patterns in class +1 are represented by yellow (rounds). Classification problems can be solved by trying to find a line (hyperplane) that separates the two classes.

The hyperplane with the best separator can be found by measuring the margins of the hyperplane and finding the maximum point. The kernel must be used to achieve the success of many classification algorithms for linear surfaces. Thus it can be seen that the type of kernel can affect the results of the classification performed. Hyperplane is the best dividing line between the two classes. To find a hyperplane, it can be done by looking for the hyperplane margin and looking for the maximum point. Margin is the distance between the closest data between two different classes, which is called the support vector.

Therefore, a pattern that satisfies class -1 is a pattern that satisfies the equation $\boldsymbol{w} \cdot \boldsymbol{xi} + b = -1$ and a pattern that satisfies class +1 is a pattern that satisfies the equation $\boldsymbol{w} \cdot \boldsymbol{xi} + b = 1$. Support vectors are represented as points (x, y). Hyperplane as follows:

$Ax + By + C = 0$,

with the distance formula as follows:

$$d = \frac{|Ax + By + C|}{\sqrt{A^2 + B^2}}.$$

Equation above is converted into a dot product in a vector so that it becomes:

$$[AB] = \begin{bmatrix} x \\ y \end{bmatrix} + C = 0$$

Suppose $\boldsymbol{w} = [A \ \ B]$ and $\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}$ and b = C, then we get:

$$d = \frac{|Ax + By + C|}{\sqrt{A^2+B^2}} = \frac{|w \cdot x + b|}{\sqrt{w^2+C^2}} = \frac{|w \cdot x + b|}{\sqrt{w^2}} = \frac{|w \cdot x + b|}{||w||}$$

The margin value can be found using the middle value between the two classes as follows:

$$margin = \frac{1}{2} (d^+ - d^-)$$

$$= \frac{1}{2} \left(\frac{|w \cdot x_1 + b|}{||w||} - \frac{|w \cdot x_2 + b|}{||w||} \right)$$

$$= \frac{1}{2} \left(\frac{1}{||w||} - \frac{(-1)}{||w||} \right)$$

$$= \left(\frac{1}{||w||}, ||w|| \neq 0, \right)$$

Where:

d +: the distance between the hyperplane against class +1, distance between hyperplane and class -1. Each class must add restrictions on the data from each class so that it does not enter into the margins, the limitations are as follows:

$$\boldsymbol{w} \cdot \boldsymbol{x_i} + b \leq -1, \text{ if } y = -1,$$
$$\boldsymbol{w} \cdot \boldsymbol{x_i} + b \geq +1, \text{ if } = +1,$$

or it can be written as follows:

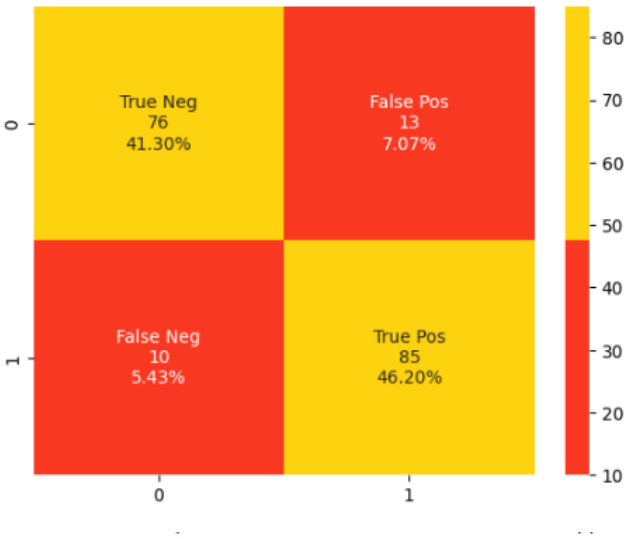
$$yi \ (\boldsymbol{w} \cdot \boldsymbol{xi} + b) - 1 \geq 0, \forall 1 \leq i \leq n, \ i \in N.$$

Maximizing the equivalent margin value by minimizing $\|\boldsymbol{w}\|2$. Then the search for the best hyperplane with the largest margin value can be formulated into a quadratic programming optimization problem as follows:

$$max \ margin = min \ \frac{1}{2} ||w||^2$$

with constraints:

$$yi \ (\boldsymbol{w} \cdot \boldsymbol{xi} + b) - 1 \geq 0, \ \forall 1 \leq i \leq n, i \in N.$$



0	0.88	0.85	0.87	89
1	0.87	0.89	0.88	95
accuracy			0.88	184
macro avg	0.88	0.87	0.87	184
weighted avg	0.88	0.88	0.87	184

Figure 4: Prediction using SVM

The confusion matrix for the Support Vector Machine (SVM) model provides insights into its classification performance for heart disease detection. The model correctly identified 76 cases as negative (True Negatives, 41.30%) and 85 cases as positive (True Positives, 46.20%). However, it misclassified 13 instances as positive when they were negative (False Positives, 7.07%) and 10 instances as negative when they were positive (False Negatives, 5.43%).

This distribution indicates that the SVM model performs well in distinguishing between positive and negative cases. However, minimizing false negatives is critical in medical applications like heart disease detection, as missing a positive case can have severe consequences. The performance demonstrated here

suggests that SVM is a strong candidate for classification, with scope for further optimization to reduce false predictions.

The classification report for the Support Vector Machine (SVM) model summarizes its performance across key metrics. For class 0 (negative cases), the precision is 0.88, recall is 0.85, and the F1-score is 0.87, reflecting strong performance but slightly lower recall. For class 1 (positive cases), the precision is 0.87, recall is 0.89, and the F1-score is 0.88, indicating a balanced and robust classification. The overall accuracy of the SVM model is 88%, demonstrating its effectiveness in correctly classifying both positive and negative cases. The macro average values (precision: 0.88, recall: 0.87, F1-score: 0.87) and weighted average values (precision: 0.88, recall: 0.88, F1-score: 0.87) highlight consistent performance across the dataset, accounting for potential class imbalances. These results indicate that SVM is a reliable choice for heart disease detection, with opportunities for further fine-tuning to enhance recall and precision for critical cases.

- ***Decision Tree Algorithm and Results Derived***

Decision tree is a classification and regression algorithm that is part of the ensemble learning group. Decision tree is one method that can be used to classify a team of objects or data to produce a decision. This approach consists of a series of selected nodes, connected via branches, moving downwards from the root node until it ends at the leaf node. Decision tree development starts from the root node, mainly based on the convention positioned at the top of the decision tree diagram, all attributes are evaluated at the selection node, with each possible outcome resulting in a branch. Each branch can enter either to another decision node or to a leaf node.

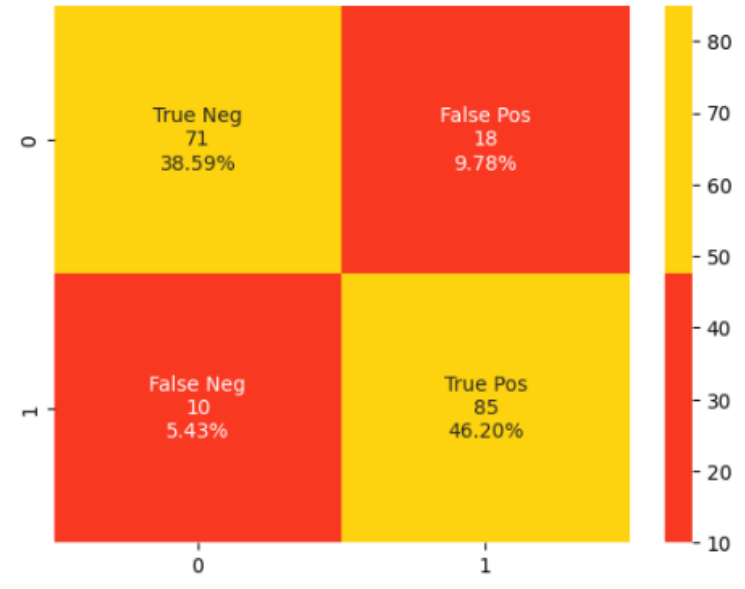
1. A decision tree is called a decision tree because the rules that are formed are similar to the shape of a tree. The tree is formed from a binary recursive sorting process in a data cluster so that the value of the response variable in each data cluster resulting from the sorting will be more homogeneous. In the decision tree, there are three types of nodes, including the top node, at this node there is no input and can have no output or can have more than one output.
2. Internal node Is a branching node, at this node there is only one input and has a minimum of two outputs.
3. Leaf Is the end node or terminal node, at this node there is only one input and has no output (terminal node). The concept of a decision tree is to convert data into a decision tree and decision rules.

The decision tree is a set of if - then rules, where each path in the tree is associated with a rule where the premise consists of a set of nodes encountered and the conclusion of the rule consists of classes associated with the leaves of the path. The formation of a decision tree consists of several stages:

1. Tree construction begins with the formation of roots (located at the top). Then the data is divided based on the attributes that are suitable for making leaves.
2. Tree pruning, which is to identify and remove branches that are not needed on a tree that has already been formed. This is because the constructed decision tree can be large, so it can be simplified by pruning based on the confidence level. Tree pruning is done in addition to reducing tree size, it also aims to reduce the level of prediction error in new cases from the results of solving using divide and conquer. Pruning there are twoapproaches, namely:

a. Pre-pruning is stopping the construction of a subtree early (by deciding not topartition the training data any further). When it stops immediately, the node turns into a leaf (end node). This final node is the class that appears most often among the sample subsets.

b. Post-pruning is simplifying the tree by removing some of the subtree branches after the tree has been built. The node that is rarely truncated will become the leaf (end node) with the class that appears the most.



0	0.88	0.80	0.84	89
1	0.83	0.89	0.86	95
accuracy			0.85	184
macro avg	0.85	0.85	0.85	184
weighted avg	0.85	0.85	0.85	184

Figure 5: Prediction using Decision Tree

The confusion matrix for the Decision Tree model provides an overview of its classification performance for heart disease detection. The model correctly identified 71 cases as negative (True Negatives, 38.59%) and 85 cases as positive (True Positives, 46.20%). However, it misclassified 18 instances as positive when they were actually negative (False Positives, 9.78%) and 10 instances as negative when they were positive (False Negatives, 5.43%). This performance suggests that while the Decision Tree model is effective at identifying positive cases, it has a relatively higher false positive rate compared to other models. Reducing false positives is essential to avoid unnecessary follow-ups or interventions. The matrix highlights the need for fine-tuning the Decision Tree's hyperparameters or combining it with ensemble methods to improve overall classification reliability in medical diagnosis scenarios like heart disease detection.

The classification report for the Decision Tree model summarizes its performance metrics for each class in the heart disease detection task. For class 0 (negative cases), the precision is 0.88, recall is 0.80, and the F1-score is 0.84, indicating high precision but slightly lower recall. For class 1 (positive cases), the precision is 0.83, recall is 0.89, and the F1-score is 0.86, demonstrating better recall for identifying positive cases but slightly lower precision. The overall accuracy of the model is 85%, reflecting its ability to correctly classify most cases. The macro average (precision: 0.85, recall: 0.85, F1-score: 0.85) and weighted average metrics align with the overall performance, accounting for the class distribution in the dataset. While the Decision Tree model shows a balanced performance, the lower recall for class 0 suggests potential room for optimization, such as tuning hyperparameters or employing ensemble techniques like Random Forest to enhance classification accuracy for both classes.

- ***CNN Algorithm and Results Derived***

- **Input**

The initial stage is to provide input in the form of a dataset as csv format using kaggle repository. For data sharing using the hold-out method. In this method, the data is divided into two separate sets, namely training data and test data. Usually 2/3 of the data is used as training and the other 1/3 is used as test data, or dividing the intact data into 70% for training data and 30% for test data.

- **Pre-processing**

The pre-processing stage is also referred to as the normalization stage, which is a process where data is adjusted to meet certain value limits. In this research, the pre-processing stage is divided into three stages, namely, data selection, data cleaning and data transformation.

- Data Selection: Data selection is the process of collecting data in accordance with the research. At this stage the process carried out for the selection process is in the "status" section, because in the dataset there are two categories, namely number of patients with disease and without the disease.

- Data Cleaning: Data Cleaning aims to remove erroneous data values, fix data clutter and check for inconsistent data. Irrelevant data is better discarded because its existence can reduce the quality or accuracy of the data mining results later . In this study, the first process of cleaning is used to remove inconsistent data and fields that are not needed for the next process. The deleted fields are Resting BP, Age, Cholesterol, and Fasting BS.

- **Learning Process**

The learning process carried out is training. To start the training process, an initial model is created for the initialization of the training. Many modelling techniques can be selected and applied to the dataset, suitable to address specific needs. The training process will be carried out, where the model will be trained using training data. Training is carried out to study the patterns of the data provided in order to draw information and trends from the data. After that, testing is carried out using other data, namely data testing.

- **Determination of Model Architecture**

At this stage, the CNN model is made. The making of the CNN model designed by the researcher refers to the ANN model architecture, namely there is an input layer, a hidden layer and an output layer. For the input layer contains the numbers of cases under influence. The hidden layer contains one layer, where in that layer contains neurons with an activation function. Where the number of neurons in the hidden layer is determined in a trial-error process which this process will take place during training. Meanwhile, the output layer contains one target layer, namely the value of the amount of cases in the next month from the input value. At the time of making the model, it also determines several other parameters such as neurons, optimizer, learning rate, batch size, and epoch.

- **Determination of the Maximum Epoch**

Epoch is a repetition that occurs in the training process to correct errors. An epoch represents one cycle of a machine learning algorithm 'learning' from the entire training data set. One epoch means a machine learning algorithm has 'learn' from the training data as a whole. Maximum epoch is the maximum number of epochs that can be done during the training process. The iteration will be stopped if the epoch value exceeds the maximum epoch to determine the number of epochs there is no special calculation, so in this study several epoch values were tested to reduce the error value. The smallest epoch value is 500 and the largest is 5000. The epoch can be set by the user during training.

- **Determination of the Number of Neurons**

In this study, the input layer contains two neurons containing the name of the drug and the number of drug use. In a hidden layer containing one layer, to determine the number of neurons in the hidden layer is the most difficult and important task in CNN design. Selecting the optimal number of hidden neurons for a given input-output data can involve a lot of testing with a trial-error process. This method is divided into two approaches, forward approximation chooses the smallest number of neurons and then increases continuously during the training process and vice versa with backward approximation. Meanwhile, the output layer contains one target layer, namely the value of the amount of drug consumption in the next month from the input value.

- **Determination of Learning Rate**

Learning rate is one of the training parameters to calculate the weight correction value during the training process. This learning rate value is in the range of zero (0) to (1). The higher the learning rate, the faster the training process will run. However, if the value of the learning rate is relatively too large, in general the training process can exceed the optimal state when the minimum error value is achieved. In other words, the learning rate affects the network accuracy of a system. The greater the learning rate, the network accuracy will decrease, but vice versa, if the learning rate is getting smaller, then the network accuracy will be greater or increase with the consequence that the training process will take longer (Suprianto, 2014). In this study, the authors used a value for the learning rate of 0.001.

- **Determination of batch size**

Determination of batch size serves to determine the number of observations before changing the weight. Determining the optimal batch size is relative, depending on the specifications of the computer used during training. In this study, the batch size used is fixed at 5, besides that the batch size can also be set by the user during training.

- **Determination of the optimizer**

The optimizer is an algorithm for finding the optimal weights and biases. In this study, the optimizers used are Adam (Adaptive Moment) and RMSProp.

- **Determination of activation function**

This research uses activation function, namely ReLu . This activation function is used to transmit information through its weight to neurons in the hidden layer. For the activation function used in the output layer, the researcher uses a linear activation function that is used to generate network output in the form of a value with the same range as the input value. While the activation function in the hidden layer is carried out by a trial-error process.

- **Backpropagation Algorithm**

Backpropagation is one of the artificial neural network methods with a supervised learning process. This algorithm uses a learning pattern based on the error correction rule. This backpropagation method is often used in various fields of application, such as pattern recognition, forecasting, and optimization. This algorithm aims to obtain a balanced network capability for recognize patterns during the training process and respond appropriately correct for different input patterns with training input patterns states that the Backpropagation algorithm consists of: of 4 stages, namely:

1. Initialization: is the stage of assigning an initial value to the value of input, weight, target, learning rate, and threshold.
2. Activation : determine the actual output that is in the layer hidden layer and calculates the actual output of the layer output (output layer).
3. Weight training (weight training): These stages is doing the weighting which consists of two stages, namely calculating the error gradient at the output layer and calculate the gradient error on the hidden layer.
4. Iteration: is the last stage in backpropagation, namely the process of repeating the previous stages until get the minimum error.

The backpropagation method was originally designed for feedforward neural networks, but in its development, this method was adapted for learning in other neural network models. The characteristic of this method is to minimize errors in the output generated by the network. The backpropagation algorithm has a very simple relationship setting, namely: if the output gives an incorrect result, then the weight is corrected so that the error can be minimized and the next network response is expected to be close to the correct value. This algorithm is also capable of fixing the weights on the hidden layer. The backpropagation network training algorithm consists of following stages that is:

1. The feed forward stage (feedforward).
2. The backpropagation stage.
3. The updating stage of weights and biases.

In detail the backpropagation network training algorithm can be described as follows:

Step 0: Initialization of the weights, training rate constant (α), tolerance error or weight value (when using the weight value as stop condition) or epoch max set (if using the number of epochs as a stopping condition).

Step 1: As long as the stopping condition has not been reached, then do steps 2 to 9.

Step 2: For each pair of training patterns, do step 3 until step 8.

Step 3: {Stage I: Feed forward (feedforwand)}.

Each input unit receives a signal and passes it to the unit hidden above it.

Step 4: Each unit in the hidden layer (from 1st unit to p-unit) is multiplied by the weight and added together added with bias.

Step 5: Each unit of output ($yk, k = 1,2,3, \ldots m$) is multiplied by weights and are added and added with the bias.

Step 6: {Stage II: Backward propagation}.

Each unit of output ($yk, k = 1,2,3, \ldots m$) receives a target pattern tk according to the input / input pattern during training and then the output layer error information (δk) is calculated. δk sent to the layer below it and is used to count the magnitude of the weight and bias correction (ΔW_{jk} and ΔW_{ok}) between layers hidden with the output layer.

Step 7: In each unit layered hidden (from 1st to p-unit; $i = 1 \ldots n$; $k = 1 \ldots m$) error information is calculated hidden layer (δj). δj is then used for calculate the amount of weight and bias correction (ΔV_{ji} and ΔV_{jo}) between input layer and hidden layer.

Step 8: {Stage III: Updating weights and biases}.

Each unit of output ($yk, k = 1,2,3,\ldots, m$) is performed updating bias and its weights ($j = 0,1,2, \ldots p$) so that generate new weights and biases. Likewise for every hidden units from the 1st unit to the p-unit the weights and biases are updated.

Step 9: Test the stop condition (end of iteration).

The following will describe the architectural design of the CNN model to be made:

Table 4: CNN Architectural Design

PARAMETER	AMOUNT
TOTAL LAYER	7
INPUT LAYER	LAYER (64 NEURONS)
HIDDEN LAYER	LAYER (128,128,8 NEURONS)
OUTPUT LAYER	NEURON
AMOUNT OF EPOCHS	10

- **Training**

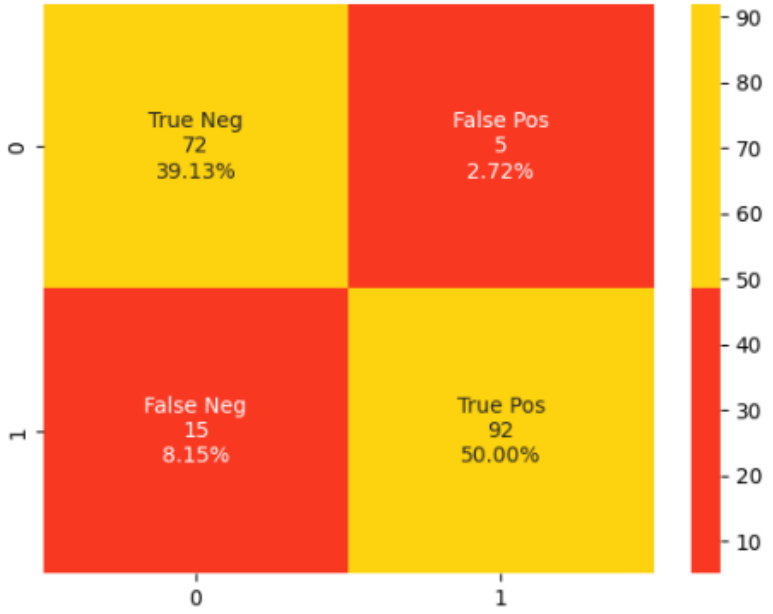
The training process is a process in which an convolutional neural network will be configured to be able to produce the required output by providing a certain data set. The network training stage uses backpropagation. The training process is carried out in three stages, namely randomizing input weights and biases, calculating hidden layers and calculating output weights. First, each input data (feature) will be entered into a node in the input layer. Then initialize the values of all weights and biases randomly in the range 0 to 1. For each input data, perform a feed forward phase by calculating the output value of each neuron in the hidden layer and output layer.

The node with the closest result to 1 after activation will be considered correct and the node will become the active node. Then do the backward propagation phase by calculating the error factor in the output layer and hidden layer. The results of these calculations will be used to calculate the rate of change of weight in the hidden layer and output layer in the next stage.

After getting the output, the next data line will be recalculated from the beginning. Then, changes will be made to the weight and bias when the amount of data being trained is already as much as the predetermined batch size. Calculation of weight changes based on the optimizer algorithm used. While the amount of change is in accordance with the learning rate used. After getting the new weight and bias, the calculation process from the input layer is carried out again. After all data rows are completed in training, the process will enter the next iteration. Iterations are carried out as many epochs are initiated. The following are the steps for training a backpropagation network using 2 input neurons, 3 hidden neurons and 1 output neuron which will be described as follows.\

- **Prediction Process (Testing)**

The testing process is carried out to test the success of the training. The data to be tested uses data testing that has gone through the previous preprocessing stage. At this stage, predictions will be made using the CNN method using the learned model obtained from the learning process at the learning stage.



Accuracy 0.8913043478260869

	precision	recall	f1-score	support
0	0.83	0.94	0.88	77
1	0.95	0.86	0.90	107
accuracy			0.89	184
macro avg	0.89	0.90	0.89	184
weighted avg	0.90	0.89	0.89	184

Figure 6: Prediction using CNN

The confusion matrix for the CNN (Convolutional Neural Network) model presents its classification performance on the heart disease detection dataset. The matrix shows the following results: True Negatives (TN): 72 cases correctly identified as class 0 (negative), contributing to 39.13% of the total predictions. False Positives (FP): 5 cases misclassified as class 1 (positive), representing a minimal error of 2.72%.
False Negatives (FN): 15 cases incorrectly predicted as class 0 (negative), which accounts for 8.15%. True Positives (TP): 92 cases correctly classified as class 1 (positive), making up 50.00%. The CNN model demonstrates high accuracy in identifying positive cases (high True Positive rate) with relatively low False Positive and False Negative rates. However, there is room to improve the False Negative rate to ensure fewer cases of undetected heart disease. Techniques such as data augmentation, hyperparameter tuning, or experimenting with advanced architectures like ResNet or EfficientNet might further optimize the CNN's performance.
The CNN model demonstrated strong performance in detecting heart disease, as evidenced by its classification metrics. For class 0 (negative), the model achieved a precision of 0.83, indicating that 83% of predicted negatives were correct, and a recall of 0.94, showing it identified 94% of actual negatives. This resulted in an F1-score of 0.88 for this class. For class 1 (positive), the model achieved an even higher precision of 0.95, meaning it effectively minimized false positives, along with a recall of 0.86, correctly identifying 86% of actual positives. This balance of precision and recall resulted in a strong F1-score of 0.90 for class 1.
Overall, the model achieved an accuracy of 89%, correctly classifying 89% of all samples. The macro average for precision, recall, and F1-score were 0.89, 0.90, and 0.89, respectively, reflecting balanced performance across both classes. The weighted averages, which account for the class distribution, were similar, further emphasizing the model's effectiveness. While the CNN model showed excellent results, particularly in precision for the positive class, slightly lower recall for this class suggests there is room for improvement to reduce missed detections. Fine-tuning the model or adjusting class weights could help optimize its recall further.

5. Conclusion

In this study, four machine learning models—Logistic Regression, Decision Tree, SVM, and CNN—were evaluated for their effectiveness in detecting heart disease. Among the models, CNN exhibited the highest overall performance, achieving an accuracy of 89%, the highest precision (0.95 for class 1), and a balanced F1-score across both classes. The Logistic Regression and SVM models showed consistent performance with an accuracy of 88%, demonstrating their reliability in handling structured data. However, both slightly underperformed compared to CNN in terms of precision and recall. The Decision Tree model achieved an accuracy of 85% but showed weaker recall for class 0, suggesting limitations in its ability to generalize well on unseen data. The CNN's superior performance can be attributed to its ability to capture intricate patterns in the dataset, while traditional models like Logistic Regression and SVM relied more on linear separability. The Decision Tree model's performance was likely impacted by overfitting tendencies, as evident from its slightly lower generalization capability.

6. Future Scope

1. Model Optimization:

-Further fine-tuning of the CNN architecture, such as experimenting with additional layers, hyperparameter optimization, or advanced activation functions, could improve its recall for class 1 and overall accuracy.

- Hyperparameter tuning for Logistic Regression, SVM, and Decision Tree can enhance their individual performances, potentially closing the gap with CNN.

2. Ensemble Techniques:

- Combining the strengths of the individual models using ensemble methods like Random Forests or Gradient Boosting could yield better results than any single model.

- Developing hybrid models that incorporate CNNs with other techniques could exploit both feature extraction and non- linear decision boundaries.

3. Data Augmentation and Expansion:

- Increasing the dataset size or incorporating additional features (e.g., demographic or lifestyle data) may improve the models' ability to generalize.

- Techniques like synthetic data generation or augmentation could address class imbalance and reduce overfitting, especially for models like the Decision Tree.

4. Explainability and Interpretability:

- Enhancing model interpretability, particularly for CNNs, using tools like SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) can provide insights into the decision-making process.

- This is particularly crucial for medical applications, where understanding predictions is as important as achieving high accuracy.

5. Integration with Real-World Systems:

- Deploying the best-performing model, such as CNN, in real-world settings like hospitals or telemedicine platforms can provide actionable insights for clinicians.

- Incorporating the model into a continuous learning framework, where it adapts and improves over time with new data, could enhance its effectiveness.

6. Cross-Domain Applications:

- The techniques explored here can be extended to other medical conditions or datasets, enabling broader use of machine learning for disease prediction and early diagnosis.

REFERENCES

[1] Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., & Adeli, H. (2017). Deep convolutional neural network for the automated detection of myocardial infarction using ECG signals. *Information Sciences, 415-416, 190-198.

[2] Alizadehsani, R., Abdar, M., Roshanzamir, M., et al. (2019). Machine learning-based coronary artery disease diagnosis: A comprehensive review. Computers in Biology and Medicine, 111, 103346.

[3] Benjamin, E. J., Muntner, P., Alonso, A., Bittencourt, M. S., Callaway, C. W., Carson, A. P., ... & Virani, S. S. (2019). Heart disease and stroke statistics—2019 update: A report from the American Heart Association. Circulation, 139(10), e56–e528. https://doi.org/10.1161/CIR.0000000000000659

[4] Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

[5] Chaurasia, V., & Pal, S. (2014). Data mining approach to detect heart diseases. International Journal of Advanced Computer Science and Information Technology, 2(1), 56-66.

[6] Chen, X., Liu, Y., Zhang, L., et al. (2020). A hybrid ensemble learning framework for cardiovascular risk prediction. Biomedical Signal Processing and Control, 57, 101832.

[7] Chollet, F. (2018). Deep learning with Python. Manning Publications.

[8] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

[9] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.

[10] Kachuee, M., Kiani, M. M., Mohammadzade, H., & Shabany, M. (2018). CVD risk prediction: A deep learning approach. Computer Methods and Programs in Biomedicine, 152, 121-132.

[11] Khan, M. A., Hashim, M. J., Mustafa, H., Baniyas, M. Y., Al Suwaidi, S. K. B. M., AlKatheeri, R., ... & Alzaabi, M. E. H. (2020). Global epidemiology of ischemic heart disease: Results from the Global Burden of Disease Study. Cureus, 12(7), e9349. https://doi.org/10.7759/cureus.9349

[12] Khourdifi, Y., & Bahaj, M. (2019). Heart disease prediction and classification using machine learning algorithms optimized by particle swarm optimization and ant colony optimization. Artificial Intelligence in Medicine, 102, 101752.

[13] Marieb, E. N., & Hoehn, K. (2018). Human anatomy & physiology (11th ed.). Pearson Education.

[14] Mohammad, S., Khan, A., & Mirza, H. T. (2021). Application of gradient boosting algorithms for cardiovascular disease prediction. Expert Systems with Applications, 164, 113891.

[15] Murphy, K. P. (2012). Machine learning: A probabilistic perspective. MIT Press.

[16] Raghunath, S., Pfeifer, J. M., Ulloa Cerna, A. E., et al. (2020). Deep neural networks can predict new-onset atrial fibrillation from the 12-lead ECG and help identify those at risk of atrial fibrillation-related stroke. Circulation, 142(15), 1510-1520.

[17] Rajak, D., Kumar, A., & Kumar, R. (2020). Heart disease prediction using hybrid machine learning model. Journal of Cardiovascular Disease Research, 11(3), 112-118.

[18] Sharma, A., Jain, K., & Choudhary, A. (2019). Heart disease prediction using hybrid deep learning model. Journal of Medical Systems, 43(9), 271.

[19] Shickel, B., Tighe, P. J., Bihorac, A., & Rashidi, P. (2018). Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. IEEE Journal of Biomedical and Health Informatics, 22(5), 1589–1604. https://doi.org/10.1109/JBHI.2017.2767063

[20] World Health Organization (WHO). (2021). Cardiovascular diseases (CVDs). Retrieved from https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)