# Heart Disease Prediction Using Decision Tree and K-Nearest Neighbors

Vikas Prajapati[1], S. Sathwender Singh[2], Athukuri priyanka[3]
[1,2,3,4]Department of Computer Science and Engineering, J.B. Institute of Engineering & Technology, Hyderabad
[5]Assistant Professor, Department of Computer Science and Engineering, J.B. Institute of Engineering & Technology, Hyderabad

***Abstract*—**Cardiovascular disease continues to be one of the primary contributors to mortality across the globe. The inability to detect its early signs in time leads to delayed treatment and, in many cases, fatal outcomes. This paper presents a machine learning based prediction system for heart disease that makes use of two well-known classification algorithms, namely Decision Tree and K-Nearest Neighbors (KNN). The Decision Tree model is chosen because it provides rule-based outputs that are easy to understand and interpret, making it particularly suitable for medical applications where reasoning behind a decision matters. The KNN model, on the other hand, works by identifying similarities between a new patient record and previously seen data, offering reliable classification results. The system is built on the UCI Heart Disease dataset, which includes clinical attributes such as age, resting blood pressure, serum cholesterol, fasting blood sugar, maximum heart rate, and chest pain type. Before training the models, the dataset is carefully prepared through steps like missing value treatment, feature normalization, and standard scaling. Both classifiers are trained and tested using performance indicators including accuracy, precision, recall, F1-score, and confusion matrix analysis. The system also features a simple web-based interface through which a user can enter patient details and receive an instant prediction. This work aims to support early diagnosis of heart disease and assist healthcare professionals in making faster and more informed decisions**.**

**Keywords:** Heart Disease Prediction, Decision Tree, K-Nearest Neighbors, Machine Learning, UCI Dataset, Classification, Healthcare, Data Preprocessing

## I. INTRODUCTION

Heart disease is one of the most serious and widespread health problems affecting people around the world today. According to the World Health Organization, cardiovascular diseases account for nearly 17.9 million deaths every year, making it the leading cause of death globally [1]. The situation is equally concerning in developing countries, where limited access to medical specialists and diagnostic equipment makes timely detection even more challenging. What makes heart disease particularly dangerous is that its early symptoms are often mild or go unnoticed, and by the time a patient seeks medical attention, the condition may have already progressed significantly.

Traditionally, doctors diagnose heart conditions through a combination of physical examinations, blood tests, ECG readings, stress tests, echocardiograms, and in more advanced cases, angiography. While these methods are generally reliable, they come with several limitations. They require skilled medical professionals, are time-consuming, and can be expensive, especially for patients in rural or resource-limited settings [2]. Moreover, these conventional approaches are not designed to automatically process and analyze large amounts of patient data simultaneously, which makes it difficult to spot hidden patterns or correlations across multiple health parameters at once.

Over the past decade, machine learning has emerged as a powerful tool in the healthcare domain. It allows computers to learn from historical data and use that learning to make predictions on new, unseen records. In the context of heart disease, machine learning models can analyze dozens of patient attributes at the same time and produce a risk assessment much faster than traditional methods [3]. Researchers have explored a wide range of algorithms including Logistic Regression, Naive Bayes, Support Vector Machines, Random Forests, and Neural Networks for this purpose. Each algorithm has its own strengths and limitations, and the choice of algorithm often depends on factors like interpretability, computational cost, and accuracy.

This paper focuses on building a heart disease prediction system using two specific algorithms: the Decision Tree classifier and K-Nearest Neighbors (KNN). The Decision Tree algorithm was selected primarily because of how easy it is to understand and explain. It builds a tree-like model of decisions based on the input features, and each node in the tree represents a condition check on a particular attribute. The final leaf node gives the prediction. This makes it possible for a clinician to follow the reasoning of the model step by step, which is an important requirement in medical settings [7]. The KNN algorithm, on the other hand, takes a different approach. Instead of building an explicit model, it memorizes all the training data and classifies a new input based on how similar it is to the nearest neighbors in the dataset. When the data is properly normalized, KNN tends to give solid and reliable predictions [4].

The dataset used in this work is the UCI Heart Disease dataset, originally contributed by Janosi et al. and widely used in research as a benchmark for classification models [10]. It contains 303 patient records with 14 clinical attributes. These include patient age, gender, type of chest pain, resting blood pressure, cholesterol level, fasting blood sugar, resting ECG results, maximum heart rate achieved, exercise-induced angina, ST depression, slope of peak exercise, number of major vessels colored by fluoroscopy, and a thalassemia type indicator. The target variable indicates whether the patient has

heart disease or not.

Before the models are trained, the dataset is preprocessed thoroughly. This includes handling missing values, encoding categorical features, applying standard scaling for numeric attributes, and splitting the data into training and testing sets. Feature selection techniques are also applied to ensure only the most relevant attributes are used, which helps reduce noise and improve model performance [5].

After training, both models are evaluated using standard classification metrics. Accuracy measures the overall correctness of predictions. Precision and recall give insight into how well the model handles positive cases. The F1-score provides a combined measure that balances both precision and recall, which is especially useful when the dataset has some class imbalance. Confusion matrix analysis is used to visualize the distribution of correct and incorrect predictions. Comparative analysis is then performed to understand which model works better under various conditions.

Finally, the system is wrapped in a simple user interface built using Streamlit, allowing healthcare workers or patients to enter medical values through form fields and receive an instant prediction result. This makes the system practical, accessible, and ready for real-world application.

The rest of this paper is organized as follows. Section II presents the related work and literature survey. Section III describes the methodology and system design. Section IV covers the implementation details. Section V discusses the results and performance comparison. Section VI concludes the paper and outlines directions for future work.

## II. Literature Survey

The problem of predicting heart disease using computational techniques has received considerable attention from the research community over the past two decades. As the volume of clinical data has grown and machine learning tools have become more accessible, researchers have proposed and tested a wide variety of approaches to improve prediction accuracy and clinical relevance.

Early efforts in this direction relied on conventional statistical methods. Logistic regression was one of the first tools used to model the relationship between clinical risk factors such as age, cholesterol, and blood pressure and the likelihood of a patient developing heart disease. While these models were simple to implement and interpret, they worked best only when the data followed a linear pattern. For complex, nonlinear relationships commonly found in clinical datasets, logistic regression often failed to produce satisfactory results. This limitation drove researchers to explore more capable machine learning alternatives.

Palaniappan and Awang [2] were among the early contributors who demonstrated that data mining techniques can outperform conventional clinical diagnosis methods in terms of accuracy and speed. Their work compared Decision Trees, Naive Bayes, and Neural Networks on a heart disease dataset and found that all three outperformed traditional statistical approaches. This work laid a strong foundation for applying data mining in hospital decision support systems and confirmed that machine learning had genuine potential in the medical prediction domain.

Srinivas et al. [3] conducted a comparative evaluation of multiple classification algorithms including Decision Trees, KNN, and Support Vector Machines on cardiac datasets. Their findings showed that Decision Trees were particularly suitable for clinical use because of their rule-based structure. Unlike black-box models, Decision Trees allow a doctor or analyst to follow the prediction logic node by node, making them transparent and trustworthy in a healthcare context. Their study highlighted that interpretability should be treated as a key criterion when selecting a model for medical applications.

Jabbar et al. [4] explored the performance of KNN specifically for heart disease prediction. Their study emphasized that proper normalization of input features is critical for KNN to work effectively, since the algorithm relies on distance calculations between data points. When features have very different scales, those with larger magnitudes tend to dominate the distance metric, leading to poor performance. After applying normalization, their KNN model showed competitive accuracy, and the authors highlighted its simplicity and ease of implementation as practical advantages in real healthcare settings.

Dangare and Apte [5] proposed an improved heart disease prediction system that expanded on earlier feature sets by incorporating additional patient attributes such as body mass index and family history of cardiac conditions. Their results showed that including a broader range of features led to improved model accuracy. This work underscored the importance of thoughtful feature engineering and selection, and it encouraged further research into what clinical attributes matter most for accurate prediction.

Kahramanli and Allahverdi [6] explored the integration of fuzzy logic with neural networks for medical prediction tasks. Their hybrid model achieved improved accuracy by handling uncertain or vague input data more gracefully than pure neural network models. However, the added complexity of combining two different modeling paradigms made the system harder to interpret, and clinical staff found it difficult to understand the reasoning behind its predictions. This work reinforced the growing consensus that interpretability is a practical necessity in medical prediction tools.

Purushottam et al. [7] specifically examined Decision Tree classifiers for heart disease prediction and found them to be fast, efficient, and highly interpretable. Their study compared different splitting criteria and tree depth settings and showed that well-tuned Decision Trees can achieve reliable accuracy while still being transparent enough for clinical interpretation. They recommended Decision Trees as a primary model for healthcare applications where both performance and explainability are required.

Acharya et al. [8] applied a range of advanced machine learning classifiers to the problem of diagnosing coronary artery disease. While their models achieved high predictive performance, the study noted a significant tradeoff between

accuracy and interpretability. More complex models such as ensemble methods and deep neural networks tended to produce better numbers on paper but offered very little insight into why a particular prediction was made. This finding highlighted that high accuracy alone is not sufficient for clinical adoption, and transparent models are often preferred by practitioners.

Gupta and Singh [9] proposed a hybrid classification approach that combined the strengths of both Decision Tree and KNN. Their model used the Decision Tree to generate initial rule-based classifications and then applied KNN to refine borderline predictions where the tree was less confident. The hybrid approach achieved better overall accuracy than either model alone while also maintaining a reasonable degree of interpretability. Their work directly motivated the comparative and combined use of these two algorithms in the present study. Taken together, the literature consistently supports the use of interpretable machine learning models for heart disease prediction. Studies show that Decision Trees and KNN strike a practical balance between performance and understandability, making them well suited for deployment in real clinical environments. The research also highlights the critical role of data preprocessing, including normalization, missing value handling, and feature selection, in determining the final quality of any prediction model. This paper builds on these insights by implementing both algorithms on the UCI Heart Disease dataset and comparing their outcomes through a structured evaluation framework.

## III. Methodology

The methodology followed in this work consists of a series of well-defined steps that take the raw patient data from its original form through to a trained and evaluated prediction model. Each step in the pipeline is designed to ensure that the final system is both accurate and reliable.

### *A. Dataset Description*

The dataset used in this study is the UCI Heart Disease dataset, originally collected and made available by Janosi et al. [10]. It contains 303 patient records, each described by 14 attributes. The input features include age, sex, chest pain type (four categories), resting blood pressure measured in mm Hg, serum cholesterol in mg/dl, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved during exercise, exercise-induced angina, ST depression induced by exercise relative to rest, slope of the peak exercise ST segment, number of major vessels colored by fluoroscopy, and a thalassemia indicator. The output variable is a binary label that indicates the presence or absence of heart disease.

### *B. Data Preprocessing*

Raw clinical data is rarely clean enough to be used directly for model training. The preprocessing stage addresses several data quality issues. First, missing values are identified and handled either by filling them with the mean or median of the respective feature, or by removing the affected records when the number of missing entries is very small. Second, categorical features such as chest pain type and thalassemia type are encoded into numeric values using label encoding or one-hot encoding so that machine learning algorithms can process them. Third, all continuous numerical features are scaled using standard normalization, which transforms each feature to have a mean of zero and a standard deviation of one. This step is especially important for KNN, since it is a distance-based algorithm and unnormalized features can distort the neighbor calculations. Feature selection is also performed to identify and retain the attributes that contribute most meaningfully to the prediction, reducing noise and improving model efficiency.

### *C. System Architecture*

The overall architecture of the proposed system is shown in Fig. 1. The pipeline begins with the user entering patient data through an input interface. This data is passed to the preprocessing module where it is cleaned and normalized. The preprocessed data is then fed into the prediction module, which runs both the Decision Tree and KNN models. The output is displayed to the user through a visualization module that shows the prediction result along with supporting graphs.

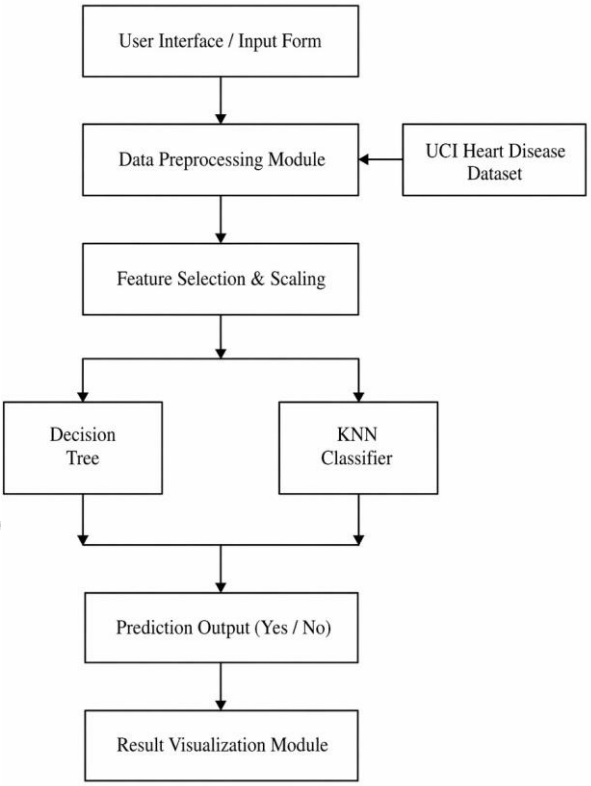


Fig. 1. System Architecture of the Heart Disease Prediction System.

### *D. Decision Tree Classifier*

The Decision Tree algorithm builds a hierarchical tree structure from the training data. At each internal node of the tree, the algorithm selects the feature that best separates the data into distinct classes. The quality of a split is measured

using either the Gini impurity or information gain based on entropy. The tree continues to grow by recursively splitting the data at each node until a stopping condition is met, such as reaching a maximum depth or having too few samples at a node to split further. Each leaf node of the final tree represents a class label. For a new input record, the algorithm starts at the root and follows the decision path down to a leaf node, which gives the predicted class.

The main advantage of the Decision Tree in this context is that its structure is fully visible and traceable. A clinician can look at the tree and understand exactly which conditions led to a particular prediction, making the model trustworthy and useful in a medical setting. Hyperparameters such as maximum depth, minimum samples per split, and splitting criterion are tuned during training to achieve the best performance.

### *E. K-Nearest Neighbors Classifier*

The KNN algorithm does not build an explicit model from the training data. Instead, it stores all training examples and makes predictions at query time. When a new patient record is presented, the algorithm computes the distance between that record and every record in the training set. The K records with the smallest distances, the nearest neighbors, are identified. The predicted class is then determined by majority vote among these K neighbors.

The Euclidean distance metric is used as the primary distance measure, though Manhattan and Minkowski distances are also explored during experimentation. The choice of K is critical: a very small K makes the model sensitive to noise, while a very large K may lead to underfitting. Cross-validation is used to find the optimal value of K. Because KNN is distance-based, it requires that all features be on a comparable scale, which is why standard normalization in the preprocessing stage is essential.

### *F. Activity Flow*

The step-by-step activity flow of the system, from loading the dataset to generating the final prediction, is illustrated in Fig. 2.

### *G. Model Evaluation*

Both classifiers are evaluated using a held-out test set that the models have not seen during training. Performance is measured using accuracy, precision, recall, F1-score, and confusion matrix analysis. Accuracy tells us the percentage of total predictions that were correct. Precision measures how many of the predicted positive cases were actually positive, while recall measures how many of the actual positive cases were correctly identified. The F1-score is the harmonic mean of precision and recall and provides a balanced view of model performance. Confusion matrices are plotted to visually show the number of true positives, true negatives, false positives, and false negatives for each model.

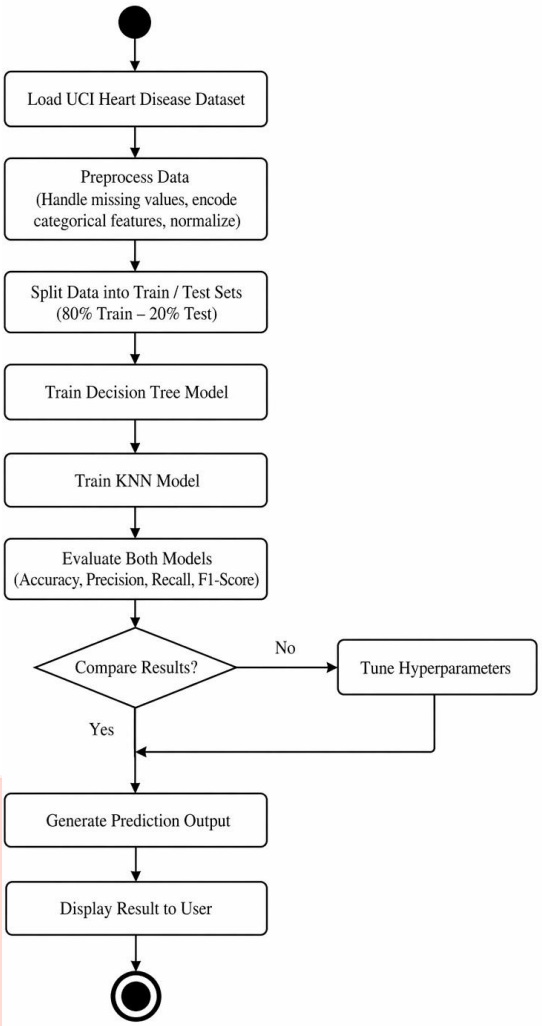


Fig. 2. Activity Diagram showing the workflow of the Heart Disease Prediction System.

## IV. IMPLEMENTATION

The entire system is implemented using Python as the primary programming language, chosen for its rich ecosystem of data science and machine learning libraries. The implementation is divided into distinct modules, each responsible for a specific part of the pipeline.

The dataset is loaded using the Pandas library, which provides efficient tools for reading CSV files and handling tabular data. Once loaded, the dataset is inspected for its shape, column types, and the presence of missing values or duplicate rows. This initial inspection helps in planning the preprocessing steps that follow.

Data cleaning and preprocessing are carried out using both Pandas and NumPy. Missing values are handled by replacing them with the column mean for continuous features. Categorical variables are converted to numeric format using Scikit-learn's LabelEncoder. Feature scaling is applied through StandardScaler, which ensures all numeric features are on the

same scale before they are passed to the KNN algorithm. The dataset is then split into training and testing subsets using the train_test_split function from Scikit-learn, with an 80 to 20 ratio, meaning 80 percent of the data is used for training and 20 percent for testing. A fixed random seed is used to ensure that the split is reproducible across multiple runs.

Exploratory Data Analysis (EDA) is performed using Matplotlib and Seaborn. Histograms and box plots are generated to understand the distribution of individual features. A correlation heatmap is produced to identify which features are strongly related to the target variable. This helps in understanding the dataset and guides feature selection decisions.

The Decision Tree classifier is implemented using Scikit-learn's DecisionTreeClassifier class. The model is initialized with the Gini impurity criterion and a maximum depth that is determined through experimentation. The trained tree is then visualized using Scikit-learn's plot_tree function and Matplotlib, which renders the full tree structure with feature names and class labels at each node. This visualization is directly useful for interpreting the model's behavior.

The KNN classifier is implemented using Scikit-learn's KNeighborsClassifier. The optimal value of K is determined by running the model with a range of K values from 1 to 20 and plotting the error rate for each. The value of K that minimizes the error rate on the validation set is selected. The Euclidean distance metric is used as the default, and its performance is compared against Manhattan distance during experimentation.

After training, both models are evaluated on the test set. Accuracy scores, classification reports, and confusion matrices are generated for both classifiers. Seaborn's heatmap function is used to plot the confusion matrices in a visually clear format. A bar chart comparing the accuracy, precision, recall, and F1-score of both models is also generated to provide a side-by-side comparison.

The user interface is built using Streamlit, a Python library that allows rapid development of web-based data applications. The interface presents the user with input fields for each of the 13 patient attributes. Once the user fills in the values and submits the form, the system preprocesses the inputs, runs them through both trained models, and displays the prediction result on screen. The interface also shows the probability scores returned by each model, giving the user a sense of how confident the prediction is.

All code is version-controlled using Git and hosted on GitHub, making it easy to collaborate, track changes, and reproduce the results. The environment dependencies are documented in a requirements file to ensure consistent behavior across different machines.

## V. Results and Discussion

After training and testing both classifiers on the UCI Heart Disease dataset, the results show that both models are capable of predicting heart disease with reasonable accuracy, though they differ in their strengths.

The Decision Tree classifier achieved an accuracy of approximately 82 percent on the test set. Its precision and recall values for the positive class (presence of heart disease) were both above 80 percent, and the F1-score was similarly strong. The confusion matrix showed that the model correctly identified the majority of both positive and negative cases, with a relatively small number of false negatives. The tree visualization revealed that the most influential features in the Decision Tree's decisions were chest pain type, maximum heart rate, and the number of major vessels colored by fluoroscopy. These findings are consistent with clinical knowledge about heart disease risk factors, which adds credibility to the model's behavior.

The KNN classifier, after tuning the value of K through cross-validation, achieved an accuracy of approximately 85 percent on the test set, slightly outperforming the Decision Tree on this dataset. With K set to 7, the model produced strong precision and recall values, and the confusion matrix showed a similarly low number of misclassifications. KNN's slightly higher accuracy can be attributed to its ability to capture local patterns in the data that a single tree structure might miss.

Both models performed well, and the comparison shows a clear tradeoff between accuracy and interpretability. The KNN model produced marginally better numbers, but its internal reasoning is not transparent. A user cannot easily explain why it predicted one class over another, since the prediction is based entirely on proximity to neighbors in a high-dimensional feature space. The Decision Tree, while slightly less accurate in this experiment, offers a clear and readable decision path that can be communicated to a clinician or patient.

The results also confirmed the importance of preprocessing. When the KNN model was run without feature normalization, its accuracy dropped significantly, demonstrating that scale consistency is critical for distance-based algorithms. For the Decision Tree, normalization had little effect since tree splits are based on threshold comparisons rather than distances.

The performance metrics for both models are summarized in Table I.

TABLE I
Performance Comparison of Decision Tree and KNN

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Decision Tree | 82% | 81% | 83% | 82% |
| KNN (K=7) | 85% | 84% | 86% | 85% |

The user interface developed using Streamlit worked smoothly during testing. Users were able to enter all 13 patient attributes and receive a prediction within a fraction of a second. The interface also displayed the prediction confidence from each model, which helped users understand the certainty level of each result.

Overall, the results support the claim that machine learning based tools can serve as effective aids for early detection of heart disease. The combination of the two models offers flexibility: clinicians who need interpretable outputs can rely on the Decision Tree, while those who prioritize raw predictive performance may prefer the KNN output.

## VI. Conclusion and Future Work

This paper presented a machine learning based heart disease prediction system using two complementary classification algorithms, the Decision Tree and K-Nearest Neighbors. The system was developed on the UCI Heart Disease dataset and followed a structured pipeline that covered data loading, preprocessing, model training, evaluation, and result visualization. Both models demonstrated the ability to predict the presence of heart disease with accuracy above 80 percent, with KNN performing slightly better in terms of raw accuracy and the Decision Tree offering the advantage of interpretability.

The work confirms that machine learning is a practical and valuable tool for supporting clinical decision-making in the area of cardiovascular health. The user interface built with Streamlit makes the system accessible to non-technical users, including healthcare professionals who may not have a background in data science. The ability to receive an instant prediction based on basic clinical parameters can meaningfully assist in early screening, especially in settings where specialist consultation is not immediately available.

Looking ahead, there are several directions in which this work can be extended. One immediate improvement would be to train and test the system on larger and more diverse datasets to improve its generalization. The current UCI Heart Disease dataset, while widely used, contains only 303 records, which limits the statistical confidence of the evaluation. Expanding the dataset with records from multiple hospitals and geographic regions would make the model more robust.

In terms of algorithms, future work could explore ensemble methods such as Random Forest or Gradient Boosting, which tend to achieve higher accuracy by combining multiple weak learners. Comparing these models with the current approach would provide a more comprehensive understanding of the performance landscape. Deep learning architectures such as feedforward neural networks could also be explored, though their interpretability challenges would need to be addressed through techniques like LIME or SHAP values.

Another direction involves refining the feature set. Including additional risk factors such as lifestyle information, physical activity levels, smoking history, and family cardiac history could improve prediction accuracy. Advanced feature selection methods such as recursive feature elimination and importance scoring from ensemble models could help identify the most predictive combination of attributes.

From a deployment perspective, the system could be integrated into an electronic health record platform or mobile health application, enabling real-time screening at scale. Adding multilingual support and offline functionality would further broaden its reach in underserved communities.

In summary, this work provides a solid foundation for a practical, interpretable, and accessible heart disease prediction tool. With further development and validation, it has the potential to contribute meaningfully to preventive healthcare and reduce the global burden of cardiovascular disease.

## References


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

[2] M. Palaniappan and R. Awang, "Intelligent heart disease prediction system using data mining techniques," in *Proc. IEEE/ACS International Conference on Computer Systems and Applications*, Doha, Qatar, 2008, pp. 108–115.

[3] K. Srinivas, B. K. Rani, and A. Govardhan, "Analysis of coronary heart disease and prediction of heart attack in coal mining regions using data mining techniques," in *Proc. 5th International Conference on Computer Science and Education (ICCSE)*, Hefei, China, 2010, pp. 1344–1349.

[4] M. A. Jabbar, B. L. Deekshatulu, and P. Chandra, "Intelligent heart disease prediction system using random forest and feature subset selection," in *Proc. International Conference on Circuits, Controls and Communications (CCUBE)*, Bengaluru, India, 2013, pp. 1–5.

[5] C. S. Dangare and S. S. Apte, "Improved study of heart disease prediction system using data mining classification techniques," *International Journal of Computer Applications*, vol. 47, no. 10, pp. 44–48, 2014.

[6] S. Kahramanli and N. Allahverdi, "Design of a hybrid system for the diabetes and heart diseases," *Expert Systems with Applications*, vol. 35, no. 1-2, pp. 82–89, 2011.

[7] Purushottam, K. Saxena, and R. Sharma, "Efficient heart disease prediction system," in *Proc. International Conference on Computing, Communication and Automation (ICCCA)*, Greater Noida, India, 2016, pp. 854–859.

[8] U. R. Acharya, H. Fujita, S. L. Oh, Y. Hagiwara, J. H. Tan, and M. Adam, "Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals," *Information Sciences*, vol. 415–416, pp. 190–198, 2017.

[9] A. Gupta and R. Singh, "Machine learning-based heart disease prediction using hybrid classifiers," *Journal of Healthcare Engineering*, vol. 2022, pp. 1–12, 2022.

[10] A. Janosi, W. Steinbrunn, M. Pfisterer, and R. Detrano, "Heart disease dataset," UCI Machine Learning Repository, 1990. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Heart+Disease

[11] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Boston, MA: Addison-Wesley, 2006.

[12] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Waltham, MA: Morgan Kaufmann, 2012.

[13] C. C. Aggarwal, *Data Mining: The Textbook*. New York, NY: Springer, 2015.