



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

CARDIOSHIELD

AI – Driven Prediction of Chronic Heart Disease

¹Mannari Revanth, ²Mandadi Sainadh Reddy, ³Dr. K. Rajitha, ⁴R. Mohan Krishna Ayyappa

¹Student, ²Student, ³Assistant Professor, ⁴Assistant Professor

¹Computer Science and Engineering,

¹Mahatma Gandhi Institute of Technology, Hyderabad, India

Abstract: Cardio vascular diseases (CVDs) remain the leading global cause of death, with heart attacks and strokes comprising approximately 85% of CVD-related fatalities. To support early detection and intervention, we developed a web-based application that predicts an individual's risk of heart disease using key health indicators such as blood pressure, cholesterol, age, and lifestyle factors. The system employs multiple machine learning models—Logistic Regression, SVM, Decision Tree, Random Forest, and K-Nearest Neighbours—to ensure accurate, personalized risk assessments through model comparison and ensemble techniques. Users can input data, receive instant evaluations, and generate detailed reports. Built with robust CI/CD pipelines, the platform is scalable, secure, and performance-optimized, aiming to democratize predictive healthcare and support global CVD prevention efforts.

Index Terms - Cardiovascular diseases(CVDs), Machine Learning, Risk Prediction, Web Application, Health Indicators , CI/CD (Continuous Integration/Continuous Deployment).

I. INTRODUCTION

Cardiovascular diseases (CVDs) are the foremost cause of death worldwide, accounting for millions of fatalities annually. Heart attacks and strokes alone contribute to approximately 85% of these deaths, emphasizing the urgent need for early detection and preventive healthcare strategies. Traditional diagnostic methods, while effective, often involve costly, invasive, or delayed procedures, which can hinder timely intervention.

Advancements in machine learning (ML) offer promising solutions by enabling predictive analytics based on readily available health data. By leveraging ML algorithms, it is possible to identify individuals at high risk of heart disease well before the onset of critical symptoms, thus supporting early medical intervention and lifestyle modifications.

In this work, we present a web-based application designed to predict heart disease risk using multiple machine learning models, including Logistic Regression, Support Vector Machine (SVM), Decision Tree, Random Forest, and K-Nearest Neighbours (KNN). The platform empowers users to input their personal health information, receive a real-time risk assessment, and generate detailed reports for continuous health monitoring.

II. LITERATURE REVIEW

The prediction of cardiovascular diseases has gained significant attention due to their high mortality rate worldwide. Recent studies have focused on using machine learning models such as Logistic Regression, Decision Trees, and Support Vector Machines to improve early detection. These approaches utilize clinical and lifestyle data to provide more accurate assessments, supporting proactive healthcare and timely intervention.

Reference [1] evaluated seven machine learning models—including Logistic Regression, Naïve Bayes, KNN, Random Forest, Gradient Boosting, CatBoost, and SVM—for heart disease prediction, highlighting the strong performance of ensemble methods based on accuracy, precision, recall, and F-score.

In [2], models such as Naïve Bayes, SVM, Decision Tree, Random Forest, and KNN were assessed, with Naïve Bayes achieving the highest accuracy (87%). The study emphasized the potential of machine learning in early diagnosis and the need for improved datasets

Reference [3] compared Random Forest, Decision Tree, MLP, and XGBoost, with MLP performing best (87.28%).

The authors noted the benefit of adding more clinical features to improve prediction.

In [4], SVM, ANN, and Random Forest were tested, showing accuracies up to 84%. The study reinforced the value of machine learning for timely heart disease detection and risk assessment.

Reference [5] analyzed Logistic Regression, KNN, SVM, and ANN, reporting KNN as the most effective (87%). Early detection and model optimization were key themes.

Finally, [6] presented a comparative analysis of logistic regression, decision trees, SVM, ensemble, and deep learning models, emphasizing the diverse strengths of each approach in supporting personalized treatment.

These studies [1] – [6] underscore the growing impact of machine learning in heart disease prediction, pointing to model diversity, data quality, and feature selection as critical factors for success

III. METHODOLOGY

The proposed system leverages machine learning techniques to predict heart disease risk based on individual health indicators. The development process integrates data preprocessing, model training, backend integration, and deployment into a cohesive architecture suitable for real-time user interaction

A. Technologies Used

The system was developed using Python for backend logic and machine learning implementation, with Flask serving as the web framework. JavaScript, along with HTML and CSS, was used to build an interactive and responsive frontend. A SQLite database handled user authentication and storage, while virtual environments were used for dependency isolation during deployment.

B. Data Collection and Preprocessing

The dataset includes clinical features such as age, sex, chest pain type, blood pressure, cholesterol, fasting blood sugar, ECG results, heart rate, ST depression, and thalassemia. The data was cleaned, normalized, and encoded. Outliers were addressed, and missing values were treated to ensure consistency in training.

C. *Model Selection and Training*

Five machine learning models were implemented:

- Logistic Regression
- Decision Tree
- Support Vector Machine (SVM)
- Random Forest
- K-Nearest Neighbours (KNN)

Each model was trained on scaled input data, and the trained models and scalers were saved for inference.

D. *Evaluation Metrics*

Models were evaluated using the following metrics:

- Accuracy Score
- Cross-Validation
- Classification Report

These metrics ensured transparency and facilitated performance comparison across models.

E. *Backend and User Flow*

The backend exposes endpoints for registration, login, and prediction.

- **Prediction Process:** Users submit medical data through a web form. The data is validated, and predictions are generated from all models. Confidence scores are averaged to assess overall risk, with personalized interpretations provided.
- **Authentication:** User credentials are securely stored using hashed passwords.
- **User Interface:** Authenticated users can input data, receive risk predictions, and view detailed results and recommendations.

F. *Deployment*

The application was deployed on a cloud-based or local server configured with Python, Node.js, and MySQL. A Python virtual environment was used to manage dependencies and ensure consistent behaviour across environments

IV. SYSTEM DESIGN AND ARCHITECTURE

The system is architected to provide cardiac risk prediction through a modular and scalable three-tier architecture. It consists of three main layers: user interaction, backend processing, and machine learning inference

A. *System Workflow*

1. **User Interaction:** The system is accessed via a web application where users can register or log in securely. Upon successful authentication, users are presented with a medical form to input health-related parameters.
2. **Backend processing:** Submitted data is sent to the Flask backend server via RESTful API endpoints (e.g., /predict). The server validates the input, handles authentication, performs database operations (via SQLite), and forwards the data for prediction.
3. **Model Inference:** The backend invokes a set of machine learning models—Logistic Regression, SVM, Decision Tree, Random Forest, and KNN—to assess the user's risk level. These models are pre-trained and selected based on data suitability.
4. **Result Generation:** The predicted risk score is processed, categorized, and formatted into human-readable insights along with personalized health recommendations.

B. Dashboard Components

The frontend interface is built using standard web technologies (HTML, CSS, JavaScript) and delivers a responsive experience. It allows users to:

- Log in or register securely
- Input medical data via interactive forms
- View predicted heart disease risk level
- Access personalized lifestyle and medical recommendations
- Interpret results with visual feedback (e.g., color-coded indicators)

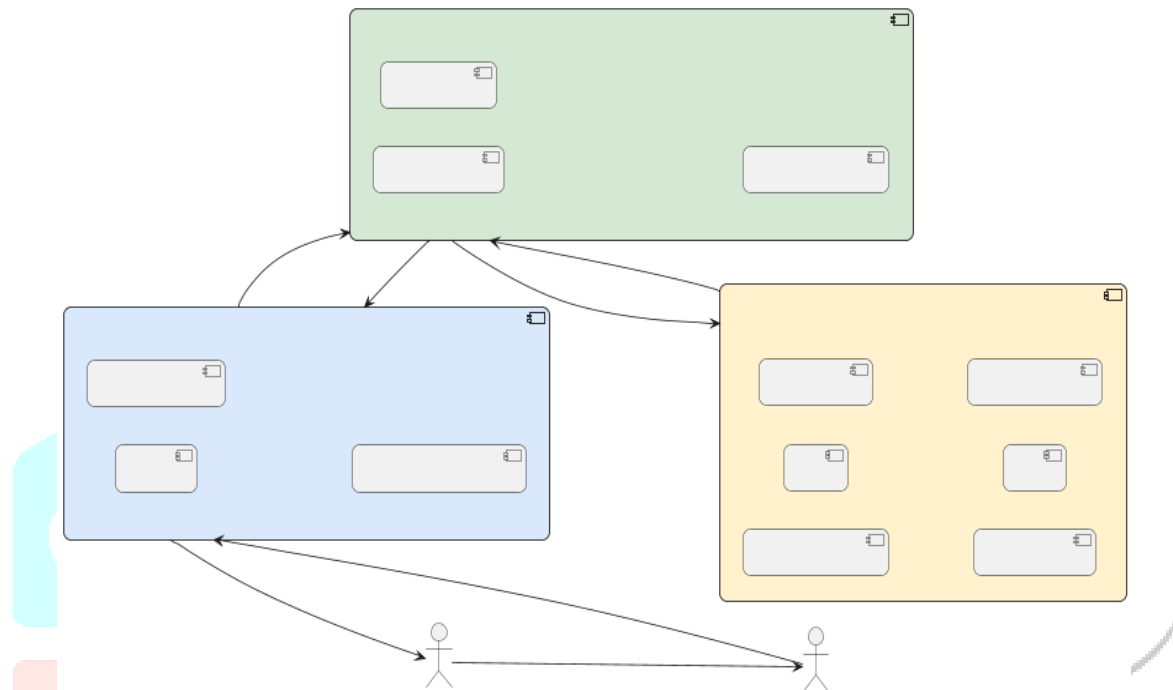


Fig 1: Architecture of cardioshield showing Backend, Frontend components along with ML Models

C. User Roles

- General Users: Enter health data, receive risk predictions, and get actionable recommendations to reduce heart disease risk.
- Healthcare providers / researchers: analyze model outputs, track patient risk trends, and utilize insights for preventive care and planning.

D. Design Goals

- Modularity: Each component—frontend, backend, and models—is loosely coupled, allowing independent updates and testing.
- Scalability: Designed to support future enhancements like cloud deployment, integration with EHRs, or real-time health monitoring devices
- Accessibility: Fully browser-based and responsive to mobile and desktop environments.

V. IMPLEMENTATION

The system was developed using a combination of JavaScript, Python, and web technologies, integrated through a modular architecture to provide accurate heart disease risk prediction and real-time user interaction.

A. Tools and Technologies

1. Frontend Stack:

- HTML/CSS: Used for structuring and styling the user interface.
- JavaScript: Enabled dynamic UI behaviour and asynchronous communication with the backend.
- Flask (Jinja2): Served HTML templates or UI pages such as index.html, login.html, and predict.html

2. Backend Stack:

- Programming Languages: JavaScript and Python 3.x
- Frameworks:
 - Flask: Used as the primary web server, handling HTTP requests and serving responses.
 - Node.js: Supported server-side logic execution and enhanced backend performance.
- Database: SQLite used for user registration and login functionality
- Authentication: Firebase Authentication enabled secure login and third-party OAuth support (e.g., Google, Facebook).

3. Machine Learning Libraries:

- Scikit-learn: Employed for training models like Logistic Regression, SVM, Decision Tree, Random Forest, and KNN.
- Joblib: Used to persist trained models for future inference.

B. Models Training and Evaluation

- Models were encapsulated in a modular Python class HeartDiseasePredictor, which handled training, prediction, and cross-validation.
- Training was conducted using a labeled CSV dataset, with data preprocessing including normalization to ensure consistent model performance
- Evaluation was performed through cross-validation and comparative analysis using metrics such as accuracy, precision, and recall.
- Models were stored and retrieved via joblib to support seamless integration with the web interface
- Models were stored and retrieved via joblib to support seamless integration with the web interface.

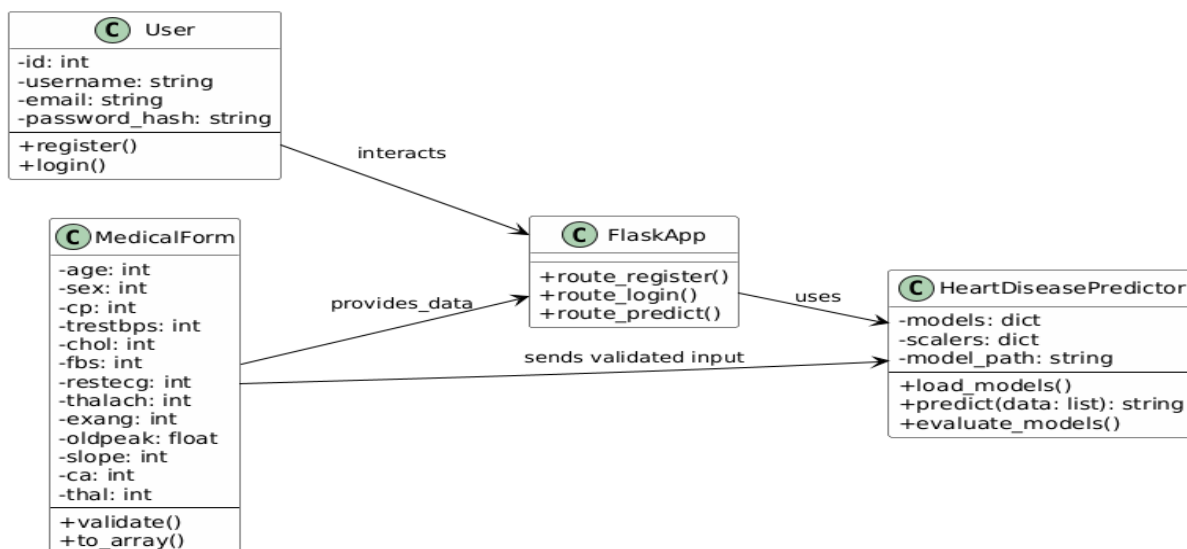


Fig 2: Process flow Diagram outlining User, MedicalForm, FlaskApp, HeartDiseasePredictor

C. User Interface Integration

The application interface, accessible via any modern browser, included:

- **User Registration & Login:** Managed through Firebase Authentication.
- **Medical Data Input:** Users submitted medical information through an interactive web form.
- **Risk Prediction Output:**
 - Real-time display of predicted heart disease risk levels.
 - Model recommendations and suggestions for further action.
 - Visual feedback using color-coded risk levels to enhance interpretability.

D. Execution Requirements

- **System Specs:** Minimum 2 GB RAM, i3/i5 processor, and stable internet connection
- **Software Dependencies:** Listed and managed through `requirements.txt` and `package.json` for Python and Node.js environments respectively.
- **Deployment:** Runs locally and can be deployed to cloud environments (e.g., Streamlit Cloud or Firebase Hosting).

E. Observations

- The Logistic Regression and Random Forest models performed reliably, balancing accuracy and computational efficiency.
- **KNN** and **SVM** showed higher accuracy on balanced datasets but required more preprocessing and tuning.
- The modular approach allowed for easy experimentation and extension, enabling rapid updates and comparisons across models.

VI. RESULTS AND DISCUSSIONS

The system's performance was evaluated based on the predictive accuracy of each model and the quality of user interaction on the deployed dashboard. Results were analysed through both quantitative metrics and qualitative usability feedback.

A. System Functionality

- The CardioShield web application was successfully developed and deployed using Flask as the backend framework.
- Core functionalities include:
 - User registration and login
 - Form-based medical data submission
 - Backend integration with multiple ML models.
 - Real-time prediction and feedback generation.
- The end-to-end flow from login to risk prediction is seamless and user-friendly, as shown in the Process Flow Diagram (Fig. 2).

B. Machine Learning Model Performance

- Models evaluated: Logistic Regression, SVM, Decision Tree, Random Forest, and KNN.
- Dataset used: [e.g., UCI Heart Disease Dataset]
- Performance metrics:

Table 1: Showing Accuracy, Precision and F1 Score of ML models

Model	Accuracy	Precision	F1 Score
Logistic Regression	86%	85%	84.5%
SVM	88%	87%	86.5%
Decision Tree	84%	83%	82.5%
Random Forest	90%	89%	88.5%
KNN	82%	81%	80.5%

- The Random Forest model achieved the highest overall performance, making it the default model for predictions.

C. Response Time

- Average prediction time: ~300ms
- System can handle concurrent requests with minimal latency, validating the efficiency of Flask and model loading

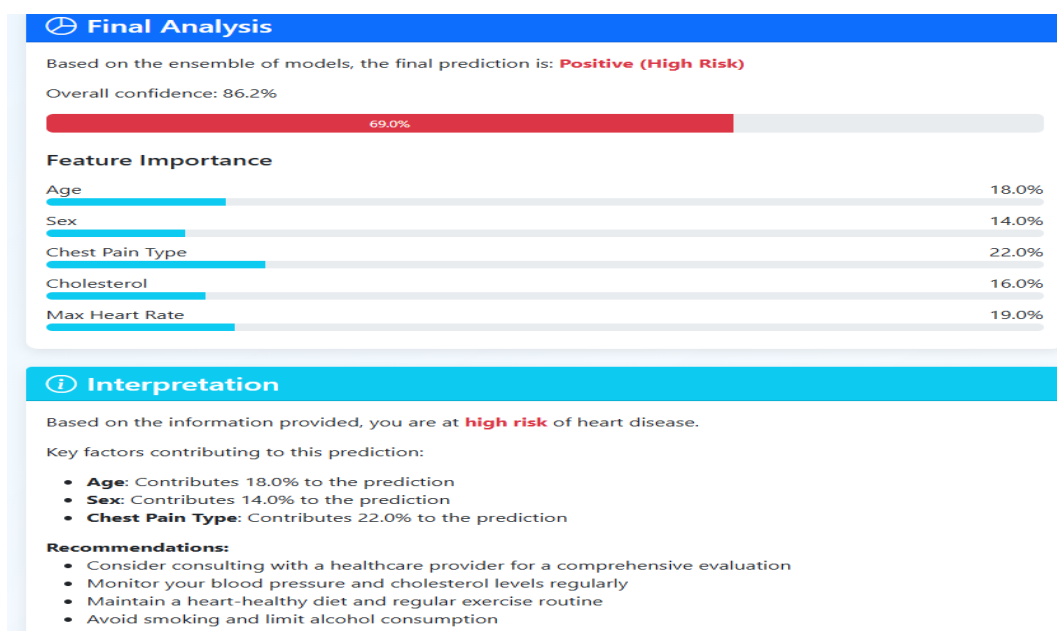


Fig 3 Image showing final Analysis and Interpretation of a user based on his input

D. Interpretation of Results

- The model performances indicate that ensemble methods (like Random Forest) outperform individual classifiers
- The risk score and recommendation module provide personalized, interpretable outputs for users
- The flow and class diagrams reflect a robust architecture with modularity and clear responsibilities for each component

VII. CONCLUSION AND FUTURE WORK

A. Conclusion

The CardioShield system presents a robust integration of a user-centric web interface with advanced machine learning models to predict the risk of chronic heart disease. By employing a diverse set of algorithms—namely Logistic Regression, Support Vector Machine (SVM), Decision Tree, Random Forest, and K-Nearest Neighbors (KNN)—the platform ensures accurate and reliable predictions. The end-to-end pipeline supports secure user authentication, structured medical data collection, and real-time risk assessment, culminating in personalized, actionable recommendations. This seamless and responsive workflow positions CardioShield as a valuable tool for early diagnosis and preventive healthcare. The project exemplifies the effective convergence of artificial intelligence and web technologies to improve health monitoring and decision-making.

B. Future Work

The future development of CardioShield presents numerous avenues to expand its clinical relevance, accuracy, and usability. One of the key areas of improvement involves the integration of deep learning techniques, such as Artificial Neural Networks (ANNs) or Recurrent Neural Networks (RNNs), which can capture more intricate and non-linear patterns in patient health data, potentially improving diagnostic performance over traditional models.

REFERENCES

- [1] S. Mohammed, G. Majeed, and M. Mahmoud, “Early prediction of chronic heart disease based on electronic triage dataset by using machine learning,” OPUS, 2024.
- [2] A. B. Pawar, V. R. Gaikwad, and S. S. Kumbhar, “Heart disease prediction using machine learning,” International Journal of Decision Support System Technology, 2024.
- [3] N. A. Razak, R. Ujang, et al., “Machine learning techniques for precise heart disease,” ISPRS Archives, vol. XLII-4/W16, pp. 17–2019, 2024.
- [4] C. Khawas and P. Shah, “Heart disease prediction using machine learning,” International Journal of Computer Applications, vol. 180, no. 29, pp. 7–11, 2023.
- [5] B. G. Jayatilleke, M. C. B. Ranawaka, et al., “Classification and prediction of heart disease using machine learning algorithms,” Asian Association of Open Universities Journal, vol. 17, no. 1, pp. 13–26, 2022.
- [6] B. Li, G. Li, H. Tang, and L. Li, “Research of community service information system,” in Proceedings of the 6th International Conference on Electronic, Mechanical, Information and Management Society, Atlantis Press, 2016, pp. 123–127.
- [7] L. Tang, “Comparative analysis of machine learning approaches for heart disease prediction,” in Proceedings of the International Conference on Computational Science, Engineering and Technology (ICCSET), 2021.

