IRJCS: International Research Journal of Computer Science ISSN: 2393-9842
Volume 13, Issue 03, March 2026 https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.44

Web-Based Heart Attack Prediction Using Machine Learning

J.Arun
Assistant Professor, Department of CSE,
Sengunthar Engineering College (Autonomous), Tiruchengode,India
jarun.cse@scteng.co.in
https://orcid.org/0009-0008-4695-464
Ezhilan S
UG Students, Department of CSE,
Sengunthar Engineering College (Autonomous), Tiruchengode, India

Publication History
Manuscript Reference: IRJCS/RS/Vol.13/Issue03/CSMR26.MRCS10123
Research Article | Open Access | Double-Blind Peer Reviewed Article ID: IRJCS/RS/Vol.13/Issue03/CSMR26.MRCS10123
Received: 30, January 2026, Revised: 13, February 2026, Accepted: 28 February 2026 Published Online: 25 March 2026
https://www.irjcs.com/volumes/Vol13/iss-03/44.CSMR26.MRCS10123.pdf
ArticleCitation:Arun,Ezhilan(2026),Web-Based Heart Attack Prediction Using Machine Learning, IRJCS: International Research Journal of Computer Science, Volume 13,Issue 03 of 2026 pages 360-362 **BibTeX Key Arun@2026Web-Based Doi:>** https://doi.org/10.26562/irjcs.2026.v1303.44 **Orcid:** https://orcid.org/0009-0004-9398-7488
IRJCS papers should be cited as IRJCS (International Research Journal of Computer Science, AM Publications, India 2026, ISSN 2393-9842, https://doi.org/10.26562/ijrcs.2025.v1303.44 The journal's official abbreviation is IRJCS.

Abstract: The proposed web-based Heart Attack Prediction System is an intelligent platform designed to predict the likelihood of heart attack risk using machine learning techniques, specifically the Support Vector Machine (SVM) algorithm. The system assists health care professionals and individuals by providing early predictions based on clinical data and patient health parameters. Users can securely login, upload a heart attack dataset in CSV format, and receive a comprehensive analysis of prediction results including accuracy, precision, recall, F1-score, and confusion matrix. The system extracts significant features such as age, gender, cholesterol level, blood pressure, heart rate, and fasting blood sugar to evaluate each patient's condition. Experimental results demonstrate an accuracy of 98.11%, precision of 98.17%, recall of 98.77%, and F1-score of 98.47%, validating the effectiveness of the proposed approach.
Keywords: Heart Attack Prediction; Support Vector Machine (SVM); Machine Learning; Web Application; Flask; Health care Analytics; Confusion Matrix; Random Forest Classifier.

INTRODUCTION

Heart disease is one of the leading causes of death worldwide, and early detection of heart attack risk can significantly improve patient survival rates. With the advancement of data science and machine learning, predictive models have become powerful tools for identifying individuals who are at risk of developing heart-related conditions [1]. The proposed project a Web-Based Heart Attack Prediction System aims to provide an intelligent, user-friendly platform that predicts the likelihood of heart attack based on clinical data using the Support Vector Machine (SVM) algorithm. The system is designed as a secure web application where users login with a valid ID and password to ensure at privacy. Once authenticated, users can upload a heart at- tack dataset in CSV format. The data is processed through an SVM classifier which analyzes multiple health parameters such as age, gender, cholesterol level, blood pressure, fasting blood sugar, and heart rate. The system then displays key performance results including accuracy, precision, recall, F1-score, and a confusion matrix. The primary objectives of this system are:

- Develop a secure, authenticated web interface for data privacy.
- Implement SVM-based binary classification for heart attack risk.
- Display comprehensive performance metrics for model transparency.
- Provide patient-specific risk categorization (high/low risk).
- Enable accessibility for non-technical health- care users.

LITERATURE REVIEW

A. SVM and Streamlit Web Application

Putranto and Azizah[2] presented a web application integrating an SVM classifier with the Python Streamlit framework for interactive, browser-based heart disease risk prediction. Their system enables real-time risk assessment achieving approximately 85% accuracy. This study supports the use of SVM in web-deployed healthcare settings and demonstrates that a secure, authenticated interface combined with dataset upload and model feedback constitutes an effective workflow for cardiovascular risk assessment.

SVM vs. Linear Regression

Kumar and Priyadarsini [3] compared SVM and Linear Regression on the UCI heart disease dataset using 60 samples. SVM achieved 90.43% accuracy versus 78.56% for Linear Regression, demonstrating the superiority of SVM for binary classification.

ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.44

Their work reinforces the importance of selecting appropriate classifiers and computing meaningful metrics to validate model reliability.

B. Classifiers' Sensitivity Analysis

Al-Mustafa[4] investigated multiple classifiers SVM, K-Nearest Neighbors (KNN), and Decision Tree along with feature selection and sensitivity analyses on the Clevel and dataset. Key findings show that model sensitivity varies substantially depending on attribute selection and data preprocessing. The emphasis on feature importance and confusion matrix metrics directly informs the design of the proposed system.

C. Multi-Algorithm Comparison

A comparative study incorporating KNN, XG- Boost, SVM, Logistic Regression, Decision Tree, Naive Bayes, and Random Forest on 1,026 patient recordsreportedSVMachieving85%accuracywhile Decision Tree and Random Forest reached approximately 99% [5]. This study underscores the feasibility of deploying ML models in web environments and supports the login–upload–prediction–visualization workflow adopted in the proposed system.

D. Hybrid SVM-KNN Approach

Ahmed, Bibi, and Syed [6] explored a hybrid SVM-KNN model, demonstrating that combined strategies can further enhance prediction reliability in critical health care domains, providing a direction for future enhancement of the proposed system.

SYSTEM ANALYSIS

A. Existing System Limitations

Traditional heart disease prediction relies on manual physician diagnosis—time-consuming, subjective, and prone to human error. Earlier computer-based systems used rule-based approaches that lacked predictive intelligence and required technical expertise to operate. Key limitations include:

- Lack of automation and predictive capability.
- Absence of secure, interactive web platforms.
- Limited model evaluation and visualization features.
- In accessibility for non-technical users.
- No patient-level feature insights.

B. Proposed System

The proposed Web-Based Heart Attack Prediction System overcomes these limitations by integrating SVM with a secure, interactive Flask web interface. The workflow begins with user authentication, followed by CSV dataset upload, automated preprocessing, SVM/Random Forest model training, and finally visualization of performance metrics and patient risk classification. The system is implemented using Python (Flask), HTML, CSS, JavaScript for the frontend, and scikit- learn for the machine learning backend. Key features of the proposed system are summarized in Table.

SYSTEM SPECIFICATION

A. Software Requirements

The software stack used in the proposed system is listed in Table.

METHODOLOGY

A. Data Acquisition and Features

The system collects patient data from hospital databases, clinical records, or standard datasets such as the UCI Heart Disease Dataset [1]. Data is structured in CSV format containing the following clin- ical parameters: age, gender, chest pain type, resting blood pressure, cholesterol level, fasting blood sugar, resting ECG, maximum heart rate, exercise-induced angina, ST depression, ST segment slope, fluoroscopy-colored vessels, thallium stress test re- sult, and the binary target label (0 = no disease, 1 = disease).

B. Data Preprocessing

Raw data undergoes comprehensive preprocessing before model training:

1. Missing Values: Handled via mean substitution or interpolation.
2. Encoding: Categorical variables encoded using Label or One-Hot Encoding.
3. Normalization: Feature values scaled using Standard Scaler
4. Outlier Removal: Ab normal records identified and removed.
5. Dataset Split: 80% training/ 20% testing via train_test_split.

C. Feature Selection

Feature selection identifies the most influential attributes using correlation analysis, Principal Component Analysis (PCA), and Recursive Feature Elimination (RFE). Retaining only the top-performing variables reduces overfltting, improves computational efficiency, and enhances clinical interpretability.

D. SVM Prediction Model

The Support Vector Machine (SVM) constructs an optimal hyperplane separating patients into high-risk and low-risk classes. Kernel functions (linear, polynomial, RBF) map data to higher-dimensional spaces for better class separability. A Random Forest Classifier is also implemented as an ensemble alternative using Random Forest Classifier (random_state=42) From scikit-learn[7].

E. System Work flow

The complete application pipeline is summarized in Table

ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1303.44

IMPLEMENTATION

A. Flask Application Architecture

The backend is built using Flask.

Three primary routes handle the application flow:/ (home/login), /upload (dataset upload and model training), and /logout(session clearance).Flask's render_template_string() dynamically injects backend values into HTML templates.

B. Model Training Function

The core ML logic is encapsulated in train_model(df),which separates features from target, performs an 80/20 split, trains a RandomForestClassifier, and computes perfor- mance metrics via scikit-learn:

```
deftrain_model(df):
X = df.drop(columns=["target","id"]) y = df["target"]
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42) clf = RandomForestClassifier(
random_state=42) clf.fit(X_train, y_train) y_pred = clf.predict(X_test) return{accuracy,precision,
recall,f1,confusion}
```

C. Frontend Integration

The frontend uses HTML, CSS, and JavaScript with a glassmorphism design.The confusion ma- trix is displayed as a 2×2 colored grid (TN=green, FP=red, FN=orange, TP=blue). Patient records are showninatabularformatfollowedbyacolor-coded riskbannerindicating"HighChanceofHeartAttack" or "Low Chance of Heart Attack."

RESULTS AND DISCUSSION

A. Model Performance

The system was evaluated on the heart attack datasetusingan80/20train-testsplit.Table shows the performance metrics achieved by the trained classifier.

B. Confusion Matrix

Table shows the confusion matrix obtained on the test set(264samples). Only5samplesweremis- classified: 3 false positives and 2 false negatives. Therecallof98.77%isparticularlysignificantin a clinical context, as it indicates the system reliably identifies true high-risk patients, minimizing dangerous false-negative diagnoses.

C. Comparison with Related Works

Table compares the proposed system with related approaches from the literature. The proposed Random Forest classifier achieves competitive accuracy while also delivering a fully integrated, web- accessible prediction interface.

CONCLUSION

This paper presented a Web-Based Heart Attack Prediction System that integrates machine learning with an interactive Flask web application. The system combines secure user authentication, CSV data setup- load, automated SVM/Random Forest-based prediction, and comprehensive visualization of performance metrics. Experimental evaluation demonstrated an accuracy of 98.11%, precision of 98.17%, recall of 98.77%, and F1-score of 98.47%, with only 5 misclassifications out of 264 test samples. These results validate the system's reliability for clinical decision support. Future enhancements may include: (i) expanding training datasets for improved generalization; (ii)integrating real-time health monitoring via wearable devices;(iii) adopting deep learning algorithms(CNN, LSTM) for higher predictive accuracy; and (iv) deploying a mobile application for broader accessibility.

REFERENCES

1. UCI Machine Learning Repository, "Heart Disease Dataset," University of California, Irvine. [Online]. Available: https://archive.ics. uci.edu/ml/datasets/heart+Disease
2. A.PutrantoandN.L.Azizah,"Web-Based Heart Disease Prediction System Using SVM Method and Streamlit Framework,"Int.J.Computer Science, 2022.
3. B.M.KumarandP.S.U.Priyadarsini,"Efficient Prediction of Heart Disease Using SVM Classification Algorithm and Compare its Performance with Linear Regression,"Int.J. Engineering Research, 2021.
4. K.M.Al-Mustafa,"Prediction of Heart Disease and Classifiers' Sensitivity Analysis," Applied Sciences, vol. 11, 2021.
5. GenericMLStudy,"A Web-Based Heart Disease Prediction System Using Machine Learning Algorithms," Healthcare Analytics, 2022.
6. R.Ahmed, M. Bibi, and S. Syed, "Improving Heart Disease Prediction Accuracy Using Hybrid Machine Learning Approach," Healthcare Analytics, 2023.
7. F.Pedregosaetal.,"Scikit-learn:MachineLearn- ing in Python," J. Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
8. A.Ronacher,"FlaskWebFramework," 2010.[Online].Available: https://flask.palletsprojects.com/