International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 10 | Oct - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

# A Machine Learning-Based Web Application for Real-Time Patient Risk Level Monitoring

**Ms.A.Nandhini[1], K.Jayaabirami[2]**
Assistant professor-SG, Department of Computer Applications, Nehru College of Management, Coimbatore, Tamil Nadu, India. nandhinimca20@nehrucolleges.com
Student of II MCA, Department of Computer Applications, Nehru College of Management, Coimbatore, Tamil Nadu, India. Jayaabi8825@gmail.com

## ABSTRACT

In today's healthcare environment, continuous monitoring of patient vitals is crucial for preventing medical emergencies and improving treatment outcomes. However, due to the increasing patient load and shortage of medical staff, timely identification of high-risk patients remains a major challenge. This project introduces a Machine Learning–based web application that predicts patient risk levels in real time using vital parameters such as temperature, heart rate, blood pressure, oxygen saturation ($SpO_2$), and blood sugar. The proposed system employs the Random Forest algorithm implemented using Scikit-learn, ensuring accurate and efficient classification of patient conditions into four categories — Normal, Medium, Severe, and Critical. The web application integrates a FastAPI backend for model deployment and a frontend interface built with HTML, CSS, and JavaScript, supported by visualization tools like Chart.js and Plotly for dynamic data representation.

Testing with sample healthcare datasets demonstrated an accuracy above 90%, proving the reliability of the model. The system allows healthcare professionals to monitor patients in real time, receive visual alerts, and prioritize treatment efficiently. This intelligent solution transforms patient monitoring into a proactive, data-driven process that enhances safety, responsiveness, and decision-making in clinical environments.

## KEYWORDS

Machine Learning, Random Forest, Scikit- learn, FastAPI, Chart.js, Plotly, Real-time Patient Monitoring, Healthcare Analytics.

## 1. INTRODUCTION

Nowadays, technology plays a very big role in improving the quality of healthcare. Hospitals collect large amounts of patient data, but without proper analysis, that data is not useful. Manual monitoring can be slow and sometimes lead to late responses in emergencies.

This project introduces a machine learning– based web application that predicts the risk level of patients based on their vital signs. When a patient's details are entered, the system automatically checks their condition and classifies them as Normal, Medium, Severe, or Critical. This helps doctors quickly identify patients who need urgent care.

The system is developed using FastAPI for the backend, and Scikit-learn with the Random Forest algorithm for prediction. The results are displayed using Chart.js and Plotly, which create colorful and easy-to- understand graphs. The web app is designed to be fast, accurate, and user-friendly, allowing medical staff to monitor multiple patients efficiently.

This project makes healthcare more intelligent and proactive by providing a system that continuously monitors, predicts, and helps save lives.

## 1.1 PROBLEM STATEMENT

Traditional hospital monitoring systems rely on manual observation and periodic recording of vitals, which can delay the identification of critical health changes. With increasing patient loads and limited medical staff, early risk detection becomes difficult. Existing systems often lack predictive intelligence, real-time alerts, and integrated dashboards.

Hence, there is a clear need for a smart healthcare system that can continuously analyze patient data, predict risk levels automatically, and alert medical professionals in time to take appropriate action.

## 1.2 OBJECTIVES OF THE STUDY

- To build a machine learning–based web system that predicts patient risk levels.
- To use Scikit-learn and Random Forest for accurate classification.
- To integrate FastAPI for real-time backend operations.
- To design a user-friendly dashboard using Chart.js and Plotly for visualization.
- To help doctors identify critical patients quickly and improve decision-making.
- To reduce manual work and improve hospital efficiency.

## 1.3 SCOPE OF THE PROJECT

The project focuses on creating an intelligent system that automatically monitors patient health and predicts risk levels using data like temperature, heart rate, oxygen, and blood pressure.

The system includes:

- A machine learning model for prediction.
- A web-based dashboard for doctors to view patient details.
- FastAPI backend for fast communication.
- Chart.js and Plotly for real-time graphical visualization.

It can be used in hospitals, clinics, and telemedicine platforms.

In the future, it can be expanded with IoT sensors, SMS alerts, and mobile app access to make it more useful and advanced.

## 2. LITERATURE REVIEW

Several studies have demonstrated the role of machine learning in healthcare. Researchers such as Rajkomar et al. (2019) and Esteva et al. (2019) explored how predictive models improve diagnosis and patient monitoring. Traditional systems are limited by manual data entry and delayed response times.

Recent frameworks integrating machine learning algorithms like Random Forest and SVM have shown improved accuracy in identifying high-risk patients. Visualization tools such as Chart.js and Plotly enhance usability, while FastAPI provides an efficient way to integrate predictive models with web interfaces.

This project builds upon these findings by implementing a real-time risk prediction system that is lightweight, scalable, and user- friendly for clinical use.

## 3. PROPOSED SYSTEM

The proposed system collects patient data, processes it through a trained Random Forest model, and classifies the results into risk categories. As shown in Figure 1, the architecture of the proposed system illustrates the data flow between the web frontend, FastAPI backend, and the machine learning model, demonstrating how patient information is processed and visualized in real time.It consists of:

- A web frontend for user interaction.
- A FastAPI backend for data processing.
- A machine learning model for prediction.
- Visualization modules for real-time dashboards.

This structure ensures high accuracy, easy maintenance, and fast response times.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 10 | Oct - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

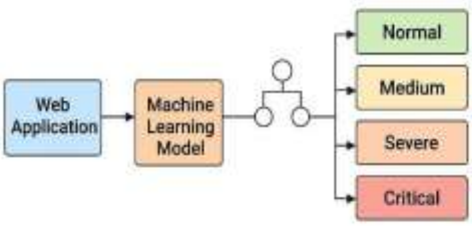


Figure 1: Architecture of the Proposed System

## 3.1 SYSTEM ARCHITECTURE

The system architecture of the proposed web application is designed to ensure efficient data flow, high performance, and real-time risk prediction. It consists of four main layers: the Input Layer, Processing Layer, Output Layer, and Database Layer, as shown in Fig. 2. Each layer plays a vital role in transforming raw patient data into meaningful health insights.

1. Input Layer:
This layer serves as the user interface where healthcare professionals enter patient details such as temperature, heart rate, blood pressure, oxygen level ($SpO_2$), and blood sugar through web forms. The input data is validated to prevent errors and ensure accuracy before being processed.

2. Processing Layer:
Once the data is submitted, it is handled by the backend developed using FastAPI. The machine learning model—trained with historical patient data using the Random Forest algorithm—analyzes the input values and predicts the patient's current health risk level. This layer performs the core logic and ensures that the prediction is generated quickly and accurately.

3. Output Layer:
After processing, the predicted result is displayed on a web dashboard in a clear and visually intuitive manner. The dashboard includes color-coded indicators (e.g., green for normal, yellow for medium, orange for severe, and red for critical) to help doctors and nurses immediately identify patients who need urgent attention. Charts and graphs are also used for better data visualization.

4. Database Layer:
All patient records, including their vitals and predicted risk levels, are stored securely in the database. This allows for easy retrieval,updates, and historical analysis of patient data. The database also supports continuous monitoring and performance tracking for future model improvements.

The interaction between these layers ensures seamless data communication, real-time processing, and accurate decision support. As illustrated in Fig. 2, the architecture visually represents how each layer is interconnected starting from data input and processing to final output visualization and secure database storage. This modular design makes the system scalable, reliable, and adaptable for future enhancements such as mobile access.

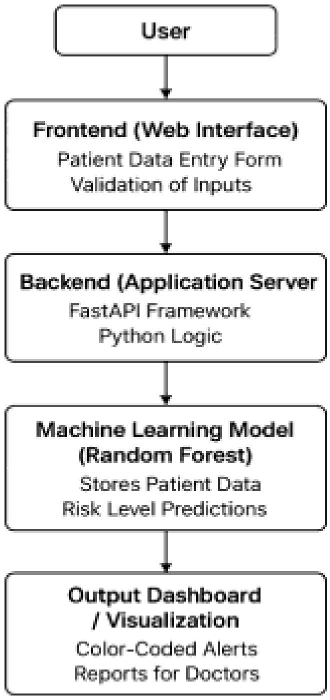


Fig. 2: Common System Architecture

The figure 2 shows how patient data flows through the system. The web interface collects inputs, which are processed by the FastAPI backend and the Random Forest model to predict risk levels. The results are stored in the database and displayed on a dashboard with color-coded alerts for doctors.

### 3.2 DATABASE AND KNOWLEDGE BASE

The database stores user credentials, patient information, and historical vitals.

- **User Table:** Maintains login, roles, and authentication details.

- **Patient Table:** Contains attributes like temperature, heart rate, blood pressure, and predicted risk level. This structured knowledge base supports quick retrieval, updating, and continuous learning for model refinement.

## 4. METHODOLOGY

The project follows an iterative model with the following phases:

1. Requirement analysis and system design.
2. Data preprocessing and model training using Scikit-learn.
3. Integration of ML with FastAPI backend.
4. Frontend design using HTML, CSS, and Bootstrap.
5. Testing and cloud deployment.

This stepwise approach ensures flexibility, rapid development, and system reliability.

### 4.1 ALGORITHMS AND TECHNIQUES USED

- Algorithm: Random Forest Classifier Reason for Selection: High accuracy, robustness, and ability to handle nonlinear relationships.

- Technique: Ensemble learning with multiple decision trees, using majority voting for classification.

- Performance Metric: Achieved over 90% accuracy in testing.Supporting tools include Pandas for data handling, NumPy for computation, and Plotly/Chart.js for visualization.

## 5. RESULTS AND DISCUSSION

The system was implemented and tested using multiple patient datasets.

Fig. 3: Patient Creation Form

This figure 3 shows the form used to register new patients into the system. It includes fields for vital details such as name, age, gender, and key health parameters. The structured input ensures that all necessary data is captured accurately for subsequent risk assessment.

Fig. 4: Registration Confirmation

Once data was entered, successful creation was confirmed (Fig. 4).

Fig. 5: Risk Prediction Results

After submission, the machine learning model processed the vitals and predicted the patient's risk category (Fig. 5).

Fig. 6: Dashboard View

A dashboard view (Fig. 6) displayed all patients at once with color-coded indicators for Normal, Medium, Severe, and Critical cases. This visualization helped healthcare providers quickly identify patients who required urgent attention.

Fig. 7: Update/Modification Form

The system also allowed existing records to be updated through a modification form (Fig. 7). Each update triggered recalculation of the risk level, ensuring real-time accuracy.

Table 5.1 summarizes the testing results for three patient datasets. The majority of patients were classified as Normal or Medium, while a smaller proportion fell into Severe or Critical categories. This distribution reflects real-world clinical settings, where fewer patients are in critical condition compared to those who are stable.

Table 5.1: Testing Results of Patient Datasets

| Dataset | No.of Record | Normal | Medium | Severe | Critical |
|---|---|---|---|---|---|
| Patient Dataset 1 | 50 | 22 | 15 | 8 | 5 |
| Patient Dataset 2 | 60 | 28 | 20 | 7 | 5 |
| Patient Dataset 3 | 55 | 25 | 18 | 7 | 5 |

The machine learning model achieved a classification accuracy of over 90%. This demonstrates its robustness in identifying patient risk levels. Furthermore, the dashboard and update features made the system practical for hospital use by enabling real-time triage and adaptive monitoring.

Overall Findings,

The system registers patients through the creation form (Fig. 3) and confirms successful entries (Fig. 4). The machine learning model then predicts each patient's risk category (Normal, Medium, Severe, or Critical) as shown in Fig. 5. The dashboard (Fig. 6) provides a real-time, color-coded overview. In contrast, updates via the modification form (Fig. 7) automatically recalculate risk levels. Testing across multiple datasets (Table 5.1) revealed that most patients were classified as Normal or Medium, with fewer cases classified as Severe or Critical, and the model achieved an accuracy rate of over 90%. Overall, the system enables efficient, reliable, and timely patient monitoring for informed clinical decisions.

## 6. CONCLUSION AND FUTURE SCOPE

### 6.1 CONCLUSION

The project successfully demonstrates how machine learning can enhance healthcare through real-time monitoring and predictive analytics.
By automating patient risk classification, it reduces manual workload, improves efficiency, and assists medical staff in making faster decisions.
The system's modular design ensures adaptability for future healthcare technologies.

### 6.2 FUTURE SCOPE

- Integration with **IoT sensors** for automatic data capture.
- **Mobile app version** for on-the-go patient tracking.

- Implementation of **AI-driven alerts** via SMS or email.
- **EHR integration** for unified patient data management.
- **Multi-language support** for accessibility across regions.

These future improvements can transform the system into a comprehensive, real-time digital health assistant.

## References

[1] J. Wiens et al., "Machine learning for healthcare: On the verge of a major transformation?" J. Am. Med. Inform. Assoc., vol. 24, no. 6, pp. 1125–1132, 2017.

[2] R. Johnson and L. Wang, "Real-time health monitoring systems using IoT and machine learning," J. Med. Syst., vol. 43, no. 6, pp. 145–156, 2019.

[3] J. Chen and C. Wang, "Model algorithm research based on Python FastAPI," Front. Sci. Eng., vol. 3, no. 9, pp. 202–210, 2022.

[4] S. Brown, "Random Forest and XGBoost: Comparison in healthcare predictive analytics," J. Healthc. Inform. Res., vol. 4, pp. 67–85, 2020.

[5] T. Shaik et al., "Remote patient monitoring using artificial intelligence: A review," WIREs Comput. Mol. Sci., vol. 13, no. 1, e1485, 2023.

[6] A. G. Mohapatra et al., "IoT-driven remote health monitoring system with sensor integration," J. Electr. Eng. Technol., vol. 20, no. 5, pp. 1456–1465, 2025.