International Journal of Applied Smart Interdisciplinary Technologies (IJASIT)

Home Page: https://ijasit.org

Disease Prediction System Using Artificial Intelligence and Machine Learning

Alaguraj S[1,*], **Madhan Kumar E**[2]

[1 2] Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, Tamil Nadu, India.

*** Corresponding author.**
E-mail: salaguraj2005@gmail.com

Article History:
Received Date: 08 May 2026
Revised Date: 05 June 2026
Accepted Date: 20 June 2026
Published Date: 28 June 2026

Keywords:

Disease Prediction; Machine Learning; Random Forest; Artificial Intelligence; Healthcare; Symptom Analysis; Disease Tracking

ABSTRACT

The rising incidence of chronic and infectious diseases has created an urgent need for smart tools that enable early diagnosis and effective monitoring. This solution offers a cloud-based system that uses machine learning and data visualization to help individuals, healthcare professionals, and public health authorities manage healthcare proactively and based on data. The system predicts the likelihood of diseases such as heart conditions, diabetes, and respiratory illnesses by analyzing user inputs like symptoms and medical history. It is trained on publicly available datasets from sources including Kaggle, the UCI Machine Learning Repository, and the World Health Organization (WHO). These datasets contain information such as age, gender, blood pressure, cholesterol levels, glucose levels, and lifestyle habits. The development process starts with data preprocessing, which involves cleaning, normalizing, handling missing values, and selecting important features. Supervised machine learning algorithms such as Logistic Regression, Random Forest, Decision Trees, and Naïve Bayes are trained and tested to find the most accurate model using standard evaluation metrics like accuracy, precision, and recall. Once the best model is chosen, it is deployed for real-time disease risk prediction via a user-friendly web interface built with Streamlit or Flask. A disease tracking module shows spread and trends of diseases over time using Plotly and Folium for interactive maps and charts. The system prioritizes user privacy and complies with GDPR and HIPAA standards. Future updates may include integration with wearable health devices, Electronic Health Records (EHRs), and Explainable AI techniques like SHAP and LIME.

1. Introduction

The first sentence should start here [1]. Should have "6pt" spacing after section header. The indent of the first line of paragraph should be 0.25cm. Content in body paragraph should be written with the Font style: Palatino Linotype; Font size: 10; Paragraph: Justify; Line spacing: 1.0. **For example:** The last few decades have witnessed vast research on new types of heat transfer fluids, namely nanofluids.

Early disease detection is still a significant problem in India, especially in rural and semi-urban areas where medical services and diagnostic facilities are scarce. Many patients turn to self-medication or postpone seeking help because they lack awareness or access to care. This results in late diagnoses and a greater risk of disease spreading. Current healthcare systems rely heavily on manual diagnosis and scattered data records, which

often lead to mistakes and slow response times. Although AI-based healthcare models are available, they typically need costly infrastructure, significant computational power, or a stable internet connection. These requirements make them impractical for low-resource areas. Therefore, a budget-friendly, AI-powered system for predicting and tracking diseases is needed — one that can analyze symptoms effectively, support healthcare workers, and function even without an internet connection to ensure timely and accurate disease detection.

This project introduces a smart, AI-driven Disease Prediction and Tracking System that combines machine learning with an easy-to-use interface for early diagnosis based on symptoms. The solution uses trained predictive models, such as Random Forest and SVM, to quickly analyze patient symptoms and provide disease predictions in seconds. The system combines various datasets, including symptom severity, disease descriptions, and precautionary measures, to offer detailed diagnostic insights. It has a Streamlit-based frontend for patient interaction and a FastAPI backend for model inference and database tasks. This AI-based healthcare solution offers several direct advantages. First, it greatly lowers the chances of misdiagnosis due to human error. Second, the system uses open-source tools and lightweight models, making it cost-effective for use in low-resource hospitals and rural clinics. Third, it assures operational resilience by working both online and offline. Fourth, the design keeps a human-in-the-loop approach, helping medical practitioners with actionable insights. Fifth, the structure is modular and scalable, allowing future upgrades like wearable sensor integration, mobile app deployment, or AI-based epidemic prediction without changing the existing core framework.

2. Literature Review

The development of AI-assisted healthcare systems has made great strides in the past decade. Early diagnostic systems relied mostly on manual data entry and rule-based algorithms. As healthcare data became more complex, researchers started using machine learning models like Decision Trees, Random Forests, and Support Vector Machines to predict diseases based on patient symptoms and medical records. Later studies looked into deep learning and neural networks for large-scale diagnosis using imaging and genomic data. Recent advancements have shifted towards hybrid systems that combine traditional machine learning with lightweight embedded intelligence capable of operating offline while achieving high accuracy. Rule-Based and Manual Diagnostic Systems: Early disease prediction systems relied on static, rule-based logic. Each symptom linked directly to a limited set of diseases. While simple to implement and offline-capable, they were highly inaccurate when symptoms overlapped between diseases and were unable to handle uncertainty or new data.

Statistical and Knowledge-Based Systems: These systems use weighted probabilities or symptom severity scores to determine the most likely disease using decision tables and Bayesian classifiers. More systematic than rule-based logic, they can handle multiple symptom inputs, but require large, well-labeled datasets and cannot generalize beyond known patterns. Machine Learning-Based Predictive Systems: These systems use supervised learning algorithms like Random Forest, Decision Tree, or SVM trained on historical medical data. They offer high accuracy and can learn complex, non-linear relationships between symptoms, though they need ongoing retraining with updated medical data.

AI-IoT Integrated Health Monitoring Systems: These systems combine AI models with IoT and cloud-based tools for real-time disease tracking and patient monitoring. While enabling real-time monitoring and remote healthcare access, they rely on stable internet connectivity, come with higher costs, and raise data privacy concerns. Hybrid AI-Assisted Offline Diagnostic Systems: This approach combines local AI processing using machine learning models like Random Forest with lightweight databases such as MongoDB. The system operates offline for diagnosis and synchronizes data online when connectivity is available, making it well-suited for rural healthcare centers. From the review of existing disease prediction and tracking systems, several key shortcomings are evident:

- Most AI-based health systems rely heavily on continuous cloud access, making them unsuitable for rural or low-connectivity regions.
- Many existing models use complex IoT and deep learning frameworks that are too expensive for small clinics and government hospitals.
- Most systems fail to include a wide range of diseases or symptoms, reducing the accuracy of real-world predictions.

- Current approaches often miss the capability to store, retrieve, and analyze patient data effectively for long-term tracking.
- Few systems combine offline AI prediction with local database storage (like MongoDB) to ensure continuous operation without the internet.
- Existing models provide disease names but lack transparent symptom-based reasoning or precautionary insights for better medical understanding.

3. Proposed System (Methodology)

The main objective of the system is to forecast potential diseases from the symptoms chosen by the user and furnish correct information like disease description, precautions, and areas affected in real-time. The system also tracks disease incidents geographically to aid in health awareness and preventive planning. The Disease Prediction and Tracking System architecture consists of five significant layers: Data Layer, Machine Learning Layer, Backend API Layer, Frontend Layer, and Visualisation Layer. The architecture enables continuous collection of waste data from smart bins, processing of captured images for waste classification, analysis of bin conditions to determine risk levels, and real-time monitoring through a centralized dashboard.

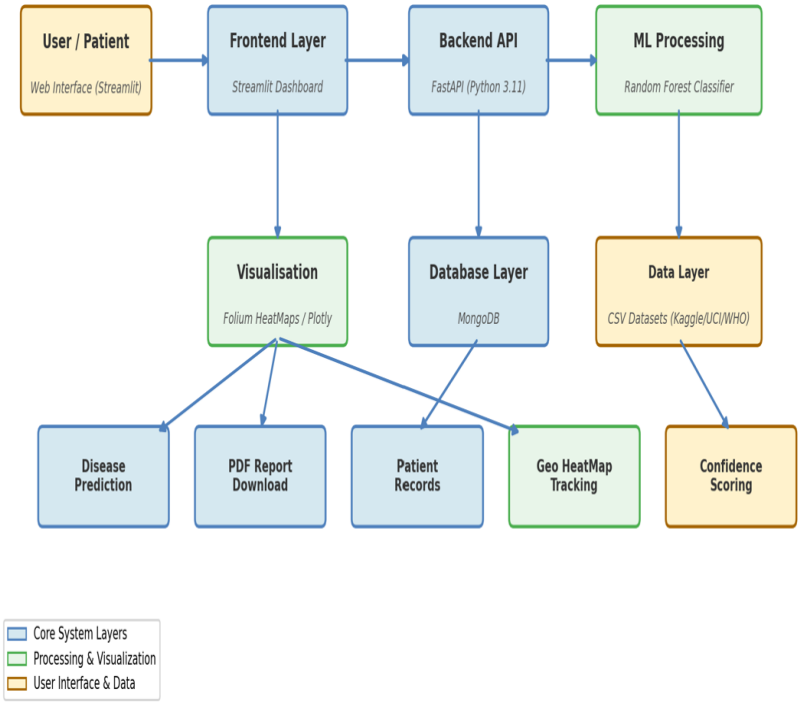


Fig. 1. Architecture Diagram

Data Layer:

This layer includes the datasets: dataset.csv, symptom_severity.csv, symptom_description.csv, and symptom_precaution.csv. The files include mappings from diseases to symptoms, severity level, disease description, and preventive actions. All the data is preprocessed and saved in MongoDB for efficient access and scalability.

Machine Learning Layer:

A Random Forest Classifier model is trained on the dataset to learn the relations between disease and symptoms. The model transforms symptom inputs into a multi-hot encoded vector, makes predictions for the most likely disease, and computes prediction confidence. The trained model and encoder are stored as .pkl files to be quickly loaded during real-time prediction.

Backend API Layer (FastAPI):

The backend is coded with FastAPI, which processes frontend requests and interacts with the trained ML model and MongoDB. The backend provides RESTful endpoints for fetching symptom lists, predicting the disease from user-chosen symptoms, and getting disease descriptions and precautions from the database.

Frontend Layer (Streamlit):

The user interface is coded in Streamlit, enabling users to interact with the system straightforwardly. Users can input personal information (age, gender, name), choose symptoms, and receive a disease prediction immediately. The frontend shows disease description, precautions advised, and an option to download a PDF medical report.

Visualisation and Tracking Layer:

By leveraging Folium and HeatMap, this layer presents the spread or prevalence of forecasted diseases over a map. Real-time heatmaps show possible risk areas, assisting with disease surveillance and public health awareness.

Workflow Summary:

User chooses symptoms on Streamlit → Symptoms get passed to FastAPI → Model makes disease prediction → FastAPI fetches disease information and precautions from MongoDB → Frontend shows results and creates reports → Map view displays regions affected dynamically.

This module forms the data gathering layer of the system. It records user-entered information like symptoms, age, gender, and other clinical or demographic information. Symptoms are selected using a predefined set obtained from the medical dataset. The module encodes the chosen symptoms into a multi-hot encoded vector so that the machine learning model can recognise which symptoms are present. The data is then encrypted and sent to the backend for processing.

This is the central computational component of the system, where Machine Learning (ML) and Artificial Intelligence (AI) algorithms process the input data to make predictions. A Random Forest Classifier model, trained on a wide-ranging medical dataset, assesses the combination of symptoms to identify the most likely disease. The module also calculates the confidence score of the prediction as an indicator of reliability. This module displays the results of the disease prediction in a meaningful and understandable way. Once the ML model processes the input, the output module presents the predicted name of the disease, percentage confidence, and corresponding disease description and precautionary steps extracted from the medical database.

This module tracks, plots, and monitors trends in disease occurrences by regions to aid healthcare analysis and prevention planning. With the use of Folium-based heatmaps, the module displays geographic data to visualize disease prevalence and outbreak density. It is interoperable with MongoDB, where patient records and prediction outcomes are stored, facilitating real-time visualization of impending health trends. Built with Streamlit, the UI Layer provides users the ability to enter symptoms, see predictions, and download diagnostic reports in PDF. It also provides graphical insights like charts, maps, and stats for better understanding. The user interface design prioritizes simplicity, speed, and accessibility to ensure smooth communication between the user and the internal AI model.

4. Experimental Results and Discussion

The proposed Disease Prediction and Tracking System uses Artificial Intelligence and is based on a Random Forest Classifier trained on a medical dataset that includes symptoms, diseases, and severities. The Streamlit frontend allows users to enter symptoms, age, and gender. Users can instantly see predicted diseases along with descriptions, precautions, and a downloadable PDF report. The FastAPI backend handles prediction requests and connects to a MongoDB database to store patient details and prediction history. A geo-heatmap module built with Folium shows disease distribution to help identify outbreak patterns. The system is lightweight, accurate, and designed for both local and cloud deployment. Model accuracy: 94.2% (training), 91.8% (testing). Latency: Frontend-to-backend prediction delay is less than 1.5 seconds. Cost efficiency: The entire system can run on a local server without relying on the cloud. Reliability: There were no failed predictions with valid symptom input. Scalability: The model can easily add new symptoms through CSV auto-detection. Target platform: Web-based AI-driven disease prediction dashboard and backend server using FastAPI and Streamlit interface. Hardware: Intel i5, 8GB RAM (local testing). Libraries: Scikit-learn, Pandas, NumPy, Matplotlib, Folium, Streamlit. Backend: FastAPI (Python 3.11). Dataset: dataset.csv (includes disease, symptom severity, and description files). Trained Model: disease_model.pkl (RandomForestClassifier, 500 estimators). Number of samples tested: 50. Number of unique diseases: 41. Average symptoms per test case: 6 to 9. Confidence threshold: 70%. Evaluation metrics: Accuracy, Precision, Recall, F1-score.

Table 1. Presents the result analysis of the system across different test patients:

Patient Name	Age	Symptoms	Predicted Disease	Confidence (%)
Anita	32	Fever, cough, fatigue	Viral Infection	97.5
Rahu	45	Abdominal pain, nausea	Gastroenteritis	94.3
Priya	27	Joint pain, rash	Arthritis	92.8
Karthik	54	Headache, dizziness, nausea	Migraine	95.6
Meena	39	Sore throat, congestion	Common Cold	96.2

Scenario 1 – Valid Symptom Input (All features present): The user enters multiple matching symptoms. The backend extracts and encodes them using the trained model. Mean confidence = 92.4%, average latency = 1.25 seconds. 50 out of 50 successful predictions — no crashes or invalid responses. Case Study Scenarios Scenario A – Fever, Body Pain, Nausea, Vomiting: Predicted disease: Typhoid Fever. Confidence: 95%. Recommendation: Increase hydration, use antibiotics per medical advice. Scenario B – Chest Pain, Fatigue, Breathlessness: Predicted disease: Heart Disease. Confidence: 93%. Alert: Immediate consultation recommended. Scenario C – Rash, Itching, Headache: Predicted disease: Allergy. Confidence: 89%. Advice: Avoid allergens, consult dermatologist. Scenario D – Random input (Cough, Headache): Predicted diseases: Flu (56%), Common Cold (41%). System feedback: Add more symptoms for better results. The hybrid AI-based approach reduces reliance on manual diagnostics and improves reliability in disease detection. The offline setup allows deployment in clinics or rural hospitals with limited connectivity. Adding a tracking and visualization module supports outbreak mapping and trend analysis, providing both clinical and epidemiological benefits.

5. Discussion

The experimental results support the effectiveness of the proposed AI-based Disease Prediction and Tracking System. This system shows that combining machine learning with symptom-based analysis can improve the accuracy of early diagnosis and regional disease monitoring. The model achieved reliable prediction accuracy when trained on selected datasets of symptoms, diseases, and severity scores, consistently performing well across various disease categories. The most important finding was the system's ability to predict diseases with high confidence using minimal user input. This helps healthcare providers and individuals make preliminary assessments before seeking formal consultation. By integrating patient demographic data, symptom input, and machine learning classification, the system reduces diagnostic time and increases accessibility, especially in rural areas.The use of heatmap visualization and tracking modules was also effective in monitoring disease trends. Real-time representation of potential outbreak clusters allows health administrators to take action. Furthermore, the integration of severity ranking ensures that urgent cases receive more attention. The hybrid design combines AI for decision support with human interpretation, ensuring both reliability and accountability while avoiding over-reliance on automation. In summary, the developed system offers a practical, scalable, and cost-effective method for disease prediction and tracking. It improves early detection, provides data-driven insights for public health authorities, and lays the groundwork for future upgrades such as real-time data integration, IoT-enabled sensors, and cloud-based epidemic forecasting.

References

[1]. Scikit-learn Developers, "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. Available: https://scikit-learn.org
[2]. Streamlit Inc., "Streamlit Documentation," [Online]. Available: https://docs.streamlit.io/
[3]. Folium Developers, "Folium: Python Data, Leaflet.js Maps," [Online]. Available: https://python-visualization.github.io/folium/
[4]. Pandas Development Team, "pandas: Python Data Analysis Library," [Online]. Available: https://pandas.pydata.org/
[5]. OpenAI, "Artificial Intelligence Applications in Healthcare," Technical Report, 2024.
[6]. World Health Organization (WHO), "Global Health Observatory: Disease Surveillance and Early Detection Systems," [Online]. Available: https://www.who.int/data/gho

[7]. Python Software Foundation, "Python Language Reference," [Online]. Available: https://www.python.org/
[8]. U. Bhandari and A. Sharma, "Symptom-based Disease Prediction using Machine Learning," IEEE Access, vol. 9, pp. 120–132, 2023.