

# CLOUD-BASED INTELLIGENT HEALTH RISK PREDICTION SYSTEM

**Ayish Nesrin P S**

Good Shepherd

College of Engineering and Technology

**Blesslin B**

Good Shepherd

College of Engineering and Technology

**Megharsha Mahesh**

Good Shepherd

College of Engineering and Technology

**Vishnu Priya J B**

Good Shepherd

College of Engineering and Technology

## ABSTRACT

Environmental hazards, such as air pollutants and toxic substances, pose significant long-term risks to human health, contributing to chronic diseases including respiratory illnesses, cardiovascular problems, and cancer. Often unnoticed, these exposures accumulate over time, impacting individual well-being. This project presents a Cloud-Based Intelligent Health Risk Prediction System that collects health parameters either through manual input controls on a web interface or via an Arduino Nano microcontroller board connected to the user's laptop by USB, which produces randomised health parameter values for the front-end. Environmental data is obtained automatically by the application using the OpenWeatherMap API, which retrieves real-time meteorological and air-quality metrics based on the user's latitude and longitude coordinates. Both health parameters and environmental data are processed together by a pre-trained machine learning model hosted in a Flask back-end. The model computes a health risk score (0–100) and leverages the DeepSeek LLM to generate personalized recommendations, with all outputs displayed on the web interface.

## KEYWORDS

Artificial Intelligence, Health Risk Prediction, Cloud Computing, Machine Learning, Flask, Arduino, Nano, Open Weather Map API, DeepSeek LLM, Environmental Monitoring, IoT.

## 1. INTRODUCTION

Cloud computing delivers services like servers, storage, databases, and analytics over the internet, enabling flexible, on-demand access without owning physical infrastructure, with providers like Amazon Web Services and Google Cloud. It supports advanced features such as natural language processing and predictive analytics. Environmental pollution—especially PM2.5, carbon monoxide, and ozone—poses serious health risks, contributing to respiratory and cardiovascular diseases and cancer. Current systems often monitor either health or environmental data separately, limiting accurate risk assessment. With affordable tools like Arduino Nano, open weather APIs, and large language models, it is now possible to combine real-time health and environmental data. This enables cloud-based systems to predict risks and provide proactive health recommendations.

## 2. LITERATURE REVIEW

This section reviews key prior work related to health risk prediction, environmental monitoring, machine learning in healthcare, and AI-driven recommendation systems.

### 2.1 Cloud-Based Ensemble Models for Disease Risk Prediction

**Author: V.K. Daliyaand, T.K. Ramesh Year: 2024**

This study introduces an optimised ensemble of Light Gradient Boosting Machine and K-Nearest Neighbour algorithms for predicting the progression of Type 2 Diabetes on an Azure cloud platform. The model uses 10-fold cross-

validation and grid search optimisation, achieving 83.2% AUC-ROC. The work demonstrates the feasibility of cloud-hosted ML models for chronic disease risk prediction and informed the risk scoring approach used in the current system.

### 2.2 Cardiovascular Disease Prediction Using Hybrid Machine Learning

**Author: M. Lorate Shiny Year: 2025**

This paper proposes a KNN-based Galactic Swarm Optimisation technique for early detection of cardiac disorders. The system achieves 96.44% accuracy during testing using the Cleveland dataset. The integration of IoT with cloud computing for remote health monitoring influenced the data acquisition architecture of the present system.

## **2.3 Machine Learning in IoT-Based Healthcare Applications**

**Authors: Hemantha, Ayush, and Vikas Year: 2021**

This review examines the integration of ML algorithms with Healthcare IoT (H-IoT), covering applications in diagnosis, monitoring, and assistive systems. It highlights the importance of accuracy and security in practical healthcare deployments. The modular design reviewed here shaped the module separation in the current system.

## **2.4 Asthma Risk Prediction Using IoT and Smartphone Applications**

**Author: Gautham Year: 2021**

This paper presents a CNN-based asthma risk prediction tool using indoor particulate matter and weather data, implemented as a smartphone application with IoT support. The classification approach into safe, moderate, and high-risk categories was a direct influence on the risk status classification used in this project. Furthermore, the integration of real-time IoT sensor data with deep learning models demonstrated in this work inspired the adoption of a similar data pipeline architecture, reinforcing the feasibility of deploying intelligent health monitoring systems in resource-constrained mobile environments

## **2.5 IoMT Cloud-Based Breast Cancer Stage Prediction with Dee**

**Authors: Shahan Yamin and Junaid Year: 2021**

This work proposes an Internet of Medical Things cloud model for detecting breast cancer stages using deep learning, achieving 98.86% training accuracy. The approach of combining IoT data with a cloud-hosted AI model for medical prediction was a key reference for the architecture of the current system.

## **2.6 Remote Health Monitoring Using AI and Blockchain**

**Author: Gupta Year: 2023**

This paper discusses how AI and blockchain can resolve challenges in telemedicine including privacy, interoperability, and decision-making from large health datasets. The focus on preventive rather than reactive healthcare strongly aligns with the proactive risk prediction goal of the current system.

### 3.METHODOLOGY

The methodology is structured across six iterative phases: requirements definition, front-end development, Arduino programming, weather API integration, ML model training and deployment, and LLM recommendation integration.

### 4.SYSTEM ARCHITECTURE

The system follows a modular and layered architecture to ensure efficient data flow and easy maintenance. It consists of five main layers:

- Layer 1 — Physical/Data Source Layer: Arduino Nano (health data), OpenWeatherMap API (environmental data), DeepSeek via OpenRouter (LLM)
- Layer 2 — Logic Layer: Flask backend, ML inference engine, external API management
- Layer 3 — Network/Middleware Layer: Serial communication, API gateway, request routing
- Layer 4 — Access Layer: Web Serial API, HTTP request/JSON response, external service interface
- Layer 5—Presentation Layer: Browser frontend, user dashboard, visualisation and recommendations panel

### 5. MODULES

The system adopts a modular architecture where data collection, analytical processing, and AI recommendation work in a synchronised pipeline. TABLE 4 summarises all five modules with their respective technologies, key functions, and outputs

#### 5.1 DATA ACQUISITION MODULE (HEALTH & ENVIRONMENT)

This is the primary input stage of the system and consists of two sub-units. The Hardware Interface sub-module manages the USB connection with the Arduino Nano via the Web Serial API, receiving comma-separated strings of randomised health parameters (heart rate, SpO<sub>2</sub>, HRV, stress level, respiration rate, systolic and diastolic blood pressure) transmitted at 9600 baud. The Environmental Data Fetcher sub-module automatically captures the user's geographical coordinates (latitude and longitude) to call the OpenWeatherMap API, retrieving real-time air quality metrics (PM<sub>2.5</sub>, NO<sub>2</sub>, AQI) and meteorological conditions (temperature, humidity, wind speed).

## 5.2 Backend Processing Module (Flask)

The Flask back-end serves as the orchestration layer between raw data and AI models. It exposes five REST endpoints: /ports (list available COM ports), /connect (establish serial connection), /start (begin monitoring loop), /data (return latest readings), /predict (run ML inference), and /recommend (invoke LLM). The module is responsible for Data Preprocessing, where raw serial strings are parsed, type-validated, and assembled into NumPy feature vectors compatible with the serialised scaler and model pipeline.

## 5.3 ML Risk Inference Engine

The pre-trained Random Forest model performs the core quantitative analysis. Input features include temperature, humidity, AQI, heart rate, SpO<sub>2</sub>, respiration rate, and HRV. The model outputs a normalised stress/risk prediction that is converted to a Health Risk Score (0–100) using the formula:  $\text{Health Score} = 100 (\text{prediction} \times 10)$ . The score is then mapped to one of five classification categories — Excellent (0–20), Good (21–40), Moderate (41–60), Poor (61–80), and Critical (81–100) — providing an at-a-glance health status indicator. SHAP (Shapley Additive explanations) values are employed to identify the top contributing features for each prediction, enhancing clinical interpretability

## 5.4 Llm Recommendation Engine (DeepSeek)

This module provides the 'Intelligence' layer of the system. The Flask back-end constructs a structured prompt incorporating the user's health parameters, environmental metrics, risk score, and top SHAP-identified risk drivers. This prompt is transmitted to the DeepSeek large language model via the OpenRouter API using an HTTP POST request with JSON payload. The model returns human-readable, actionable lifestyle recommendations organised across six dimensions: (1) Diet, (2) Exercise, (3) Sleep, (4) Stress Management, (5) Environmental Precautions, and (6) Preventive Medical Advice. These are rendered on the front-end dashboard in a styled recommendation panel.

## 5.5 Frontend Presentation Dashboard

The front-end is implemented using HTML5, CSS3, and vanilla JavaScript. It leverages the Web Serial API to enable direct browser-to-Arduino communication without requiring backend intermediary for serial port selection. The dashboard features: a Device Connection panel with COM port dropdown; six health parameter display cards (Heart Rate, SpO<sub>2</sub>, HRV, Stress, Respiration, Blood Pressure); a Health Score display rendered as a prominent percentage; and a Lifestyle Recommendation panel with formatted AI-

generated advice. A 5-second set Interval polling loop calls the /data endpoint to maintain near-real-time parameter display.

## 6. RESEARCH CHALLENGES AND OPEN ISSUES

Despite the system's demonstrated functionality, several research challenges and limitations must be acknowledged and addressed in future iterations.

### 6.1 Simulated Vs. Real Physiological Data

The current implementation employs an Arduino Nano programmed to generate randomised, clinically plausible health parameter values rather than actual sensor readings. Consequently, the health risk scores Good Shepherd College of Engineering & Technology, Maruthamparai – 629101 | Anna University, Chennai 600025 | April–May 2026 and recommendations reflect simulated physiological states. The transition to genuine wearable sensor data (e.g., MAX30102 for SpO<sub>2</sub>/heart rate, DS18B20 for temperature) is the most critical prerequisite for clinical validation. Plausible solution: Replace the Arduino simulation firmware with a sensor-interfaced firmware stack, beginning with commercially available health-monitoring breakout boards.

### 6.2 ML Model Generalisation And Dataset Bias

The pre-trained Random Forest model was developed using synthetic health-environment datasets constructed from publicly available clinical ranges. Without validation on real-world patient cohorts, the model's generalisation capability is uncertain. Furthermore, the absence of demographic variables (age, sex, pre-existing conditions) limits personalisation depth. Plausible solution: Collect de-identified real-world health-environment data and retrain the model with proper train/validation/test splits, incorporating demographic features and applying fairness-aware training techniques.

### 6.3 Environmental Data Granularity

The OpenWeatherMap API provides city-level environmental data, which may not accurately reflect the microenvironment of an individual user (e.g., indoor vs. outdoor, street-level vs. rooftop). Urban air quality can vary significantly over short distances. Plausible solution: Supplement the weather API with a fixed-location low-cost air quality sensor (e.g., PMS5003 for PM<sub>2.5</sub>, MQ-135 for NO<sub>2</sub>) installed at the user's location.

## 6.4 Llm Reliability And Hallucination Risk

Large language models, including DeepSeek, are susceptible to hallucination generating plausible-sounding but factually incorrect medical advice. In a health context, this risk is clinically significant. The current system mitigates this by providing structured, factual input prompts and including a clear disclaimer that recommendations are AI-generated and should not replace professional medical consultation. Plausible solution: Implement retrieval-augmented generation (RAG) using verified medical knowledge bases (e.g., WHO guidelines, PubMed abstracts) as grounding documents for the LLM.

## 6.5 Data Privacy and Security

Health data is inherently sensitive and subject to regulatory frameworks including HIPAA (USA) and the DPDP Act (India). The current prototype transmits health parameters over HTTP without encryption. Plausible solution: Implement HTTPS with TLS 1.3, user authentication (JWT or OAuth2), data anonymisation pipelines, and explicit user consent mechanisms before any production deployment.

# 7. SOFTWARE AND HARDWARE DESCRIPTION

## 7.1 Core Technology Framework

The system is built using a combination of frontend web technologies and a Python-based backend. HTML provides the structure of the web interface through its standard elements. CSS handles visual styling including layout, colours, and fonts. JavaScript enables dynamic interaction, real-time data updates, and communication with the Flask backend through asynchronous API calls. Together they form a responsive and clearly functional web application.

Flask serves as the central backend framework. It routes HTTP requests, manages serial communication with the Arduino Nano, calls the OpenWeatherMap API, runs ML inference, and communicates with the DeepSeek LLM through Open Router. Its lightweight nature makes it ideal for this type of multi-integration application.

## 7.2 System Requirements

### 7.2.1 Hardware Requirements

The primary hardware component is the Arduino Nano microcontroller, which is connected to the user's laptop via USB. The Arduino Nano generates sample health

parameter values including heart rate, SpO<sub>2</sub>, blood pressure, HRV, respiration rate, and stress level within clinically plausible ranges. No other dedicated hardware sensors are required, as environmental data is fetched automatically through the OpenWeatherMap API.

### **7.2.2 Software Requirements**

The software stack includes VS Code as the development environment, the Arduino IDE for programming the Arduino Nano, and Google Collab for training and evaluating the machine learning model. The Flask application is run locally or on a cloud server with Python and all required libraries installed. No separate cloud inference infrastructure is needed at runtime.

## **8. RESULTS AND DISCUSSION**

### **8.1 System Results**

The proposed system was tested across several scenarios to evaluate its performance and usability. The system successfully performs the following functions:

- Collects health parameters from the web interface or Arduino Nano via USB
- Fetches real-time environmental data including AQI, temperature, and humidity
- Computes a health risk score between 0 and 100 using the ML model
- Classifies risk status from Excellent to Critical
- Generates personalised, natural language recommendations via DeepSeek LLM
- Displays all results in real time on the frontend dashboard

### **8.2 Frontend Dashboard Observations**

The web dashboard displays six health parameters including heart rate, SpO<sub>2</sub>, HRV, stress level, respiration rate, and blood pressure. When the Arduino Nano is connected and monitoring is started, these values update in real time. Manual entry mode also works correctly, allowing users without Arduino hardware to use the system. The Result button triggers the ML prediction and LLM recommendation generation, displaying a health score percentage and a detailed lifestyle recommendation panel.

### 8.3 ML Model Performance

The machine learning model was trained on a health-environment dataset combining physiological vitals with environmental metrics. Random Forest was selected as the final model based on its performance compared to Logistic Regression and LSTM. The model produces consistent risk scores and correct status classifications across a wide range of input combinations. The use of SHAP values allows the system to identify which factors contribute most to each prediction, improving transparency.

### 8.4 LLM Recommendation Quality

The DeepSeek model, accessed via the Open Router API, generates detailed, structured recommendations covering six categories: diet, exercise, sleep, stress management, environmental precautions, and preventive advice. The recommendations are appropriate to the user's risk score and environmental conditions. For example, when AQI is high and SpO2 is borderline, the system advises limiting outdoor activity and using air filtration indoors. The recommendations are generated once per session to avoid unnecessary API calls and are locked until the next analysis cycle.

### 8.5 Discussion

The results show that combining personal health monitoring with environmental data analysis significantly improves the quality and relevance of health risk predictions. The lightweight ML model ensures fast inference without needing expensive cloud computing infrastructure at runtime. The LLM-generated recommendations provide genuine value by translating complex risk scores into simple, actionable advice that users can easily understand and follow.

## 9. SCOPE OF THE PROJECT

The scope of future development focuses on moving from simulated data to genuine real-time monitoring. This will begin with replacing randomized Arduino output with real physiological readings by integrating a wearable IoT health band. Fixed geolocation coordinates will be replaced with GPS-based environmental monitoring for more accurate, location-aware context. Accessibility will be expanded through a mobile application, and the core analytics will be improved through advanced deep learning models that deliver more tailored and robust actionable insights.

## 10. FUTURE ENHANCEMENTS

Future iterations of this system will focus on several key enhancements. The first priority is integrating medical-grade sensors such as the MAX30102 for genuine physiological readings, replacing the current simulated Arduino data with accurate, real-time vitals.

A dedicated mobile application will be developed to improve accessibility, allowing users to monitor their health risk status from any location without requiring a laptop or desktop. A persistent database such as Amazon DynamoDB will be integrated to track long-term health trends and help refine the LLM's clinical recommendations over time based on historical patterns.

Edge computing will be deployed for offline inference, enabling the system to provide risk scores even without internet connectivity. Blockchain technology will be explored for decentralised data security, ensuring that sensitive health data remains private and tamper-proof.

### Key planned enhancements:

- Integration of MAX30102 or similar medical-grade sensors for real physiological data
- Mobile application for cross-platform accessibility
- Persistent cloud database for long-term health trend analysis
- GPS-based dynamic geolocation for precise environmental context
- Edge computing support for offline risk inference
- Blockchain-based data security for user privacy protection
- Advanced deep learning models for improved prediction accuracy

## 11. CONCLUSION

This project successfully develops a Cloud-Based Intelligent Health Risk Prediction System that bridges the gap between environmental monitoring and personalised healthcare. By combining real-time data from the OpenWeatherMap API with physiological inputs from Arduino Nano hardware, the system provides a complete view of an individual's health risks in relation to their surrounding environment.

The dual-layered AI approach, using machine learning for quantitative risk scoring and the DeepSeek LLM for qualitative, actionable recommendations, transforms raw sensor and API data into meaningful, life-improving information. The system is fully accessible through a browser, requires no complex installation, and can be deployed on a local or cloud server for use in schools, colleges, offices, and other environments. This project demonstrates how accessible, integrated technology can empower individuals to navigate polluted environments more safely, supporting a proactive approach to long-term health management and chronic disease prevention.

## **12. ACKNOWLEDGEMENT**

The authors sincerely thank the Almighty for His blessings in enabling the successful completion of this endeavour. Sincere gratitude is extended to Chairman Mr. G.D. Mitra, M.Sc., B.Ed., for providing outstanding academic opportunities; Principal Dr. Robin Kumar Samuel, M.E., Ph.D., for his invaluable support; Head of Department and Supervisor Mrs. Aiswarya Lekshmi A.C., M.E., for her guidance, insights, and encouragement throughout the project; all faculty members of the Department of Computer Science and Engineering for their timely suggestions; and the authors' parents and friends, whose unwavering support made this work possible.

## **13. REFERENCES**

1. V.K. Daliyaand, T.K. Ramesh, 'A Cloud-Based Optimised Ensemble Model for Risk Prediction of Diabetic Progression – An Azure Machine Learning Perspective', 2024.
2. M. Lorate Shiny, 'Cloud Based Proficient Cardiovascular Disease Prediction Using Hybrid Machine Learning Models', 2025.
3. Hemantha, Ayush and Vikas, 'Role of Machine Learning in Enabling IoT Based Healthcare Applications', 2021.
4. Gautham, 'Machine Learning Based Asthma Risk Prediction Using IoT and Smartphone Applications', 2021.
5. Shahan Yamin and Junaid, 'IoMT Cloud Based Intelligent Prediction of Breast Cancer Stages Empowered with Deep Learning', 2021.
6. Jones and Brown, 'Real-Time Support: The Role of Chatbots in Healthcare', 2020.
7. John Smith, 'Development of a Healthcare Chatbot for Patient Consultation', 2025.

8. Maria Garcia, 'A Review of Chatbot Applications in Chronic Disease Management', 2020.
9. Sarah Lee, 'Integrating AI Chatbots into Telehealth Platforms for Remote Patient Monitoring', 2025.
10. Sophia Patel, 'Ethical Considerations in the Development and Deployment of Healthcare Chatbots', 2023.
11. Daniel Kim, 'Improving Chatbot Performance through Continuous Learning and User Feedback Integration', 2023.
12. Jennifer Martinez, 'Utilizing Chatbots for Health Promotion and Behaviour Change: A Scoping Review', 2023.
13. Chang and Wu, 'Improving Health Literacy Through Chatbots', 2023.
14. Gupta, 'Remote Health Monitoring Using Chatbots', 2023.

A large, light blue watermark logo is centered on the page. It features a stylized lightbulb shape with a circular top and a rectangular base. Inside the circle, there are three vertical lines of varying heights, each ending in a small circle. The letters 'IJRTI' are printed in a bold, white, sans-serif font across the middle of the rectangular base of the lightbulb.

IJRTI