International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4819-4824
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0634

AI-Assisted Healthcare Kiosk: Offline Disease Prediction, Medicine Recommendation, and Telemedicine Access for Rural India

Thanu Shree M N [1], Vijayalakshmi M M[2]

[1]PG Student, Department of Computer Science Engineering, AMC Engineering College, Bengaluru, India.

[2]Assistant Professor, Department of Computer Science Engineering, AMC Engineering College, Bengaluru, India

Email: *mnthanushree2002@gmail.com[1]*

Abstract

Rural communities in India face ongoing challenges in accessing primary healthcare. These challenges include a lack of doctors, geographical isolation, and high consultation costs. This paper discusses an AI-assisted healthcare kiosk, a web-based, fully offline system that provides automated disease prediction, medication recommendations, BMI assessment, cardiovascular risk evaluation, and doctor referrals. It does all this without needing a permanent physician or internet connection. The system uses a Decision Tree classifier along with several machine learning models trained on a structured dataset of 4,920 samples, 132 symptom features, and 41 disease categories. It achieved 100% classification accuracy on the test set. The Flask-based web application features offline speech recognition through Kiosk, text-to-speech output using pyttsx3, TF-IDF vectorisation for natural language processing, and a MySQL backend for managing patient sessions. All ten functional test cases were successful, with an end-to-end response time of about one second. The proposed system proves it is possible to implement AI-powered primary healthcare tools in rural and resource-limited areas.

Keywords: *disease prediction, machine learning, decision tree, healthcare kiosk, offline AI, Flask, Kiosk, TF-IDF, rural healthcare, Python.*

1. Introduction

Access to quality primary healthcare remains a critical global challenge. The World Health Organisation (2021) estimates that at least half of the world's population lacks access to essential health services [1]. In India, the disparity is stark: approximately 65% of the population resides in rural areas, yet only 30% of healthcare infrastructure is located outside urban centres [2]. Rural patients face long travel distances, physician shortages, high consultation fees, and limited health literacy factors that collectively result in delayed diagnoses and preventable morbidity. Artificial intelligence (AI) has emerged as a transformative force in healthcare, offering the potential to automate clinical triage, symptom assessment, and patient intake in settings where physician presence cannot be guaranteed [3]. This paper presents an AI-assisted healthcare kiosk designed to address rural healthcare constraints through fully offline, browser-based technology with no cloud dependency. The primary contributions of this work are:

- A fully offline AI healthcare kiosk deployable on commodity hardware (Intel Core i5, 8 GB RAM).
- A Decision Tree classifier trained on 4,920 samples across 41 disease categories and 132 symptom features, achieving 100% test accuracy.
- A validated functional test suite of ten categories confirming end-to-end system robustness with ~1 second response latency.

2. Related Work

2.1. AI Symptom Checkers

Early symptom-checking tools relied on rule-based expert systems. Semira et al. [4] evaluated 23 online symptom checkers and found significant variability in accuracy. Razzak et al. [5] compared AI triage against general practitioners, reporting comparable

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4819-4824
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0634

diagnostic accuracy for common conditions. Chen et al. [6] achieved 82% triage accuracy using an NLP-driven outpatient symptom checker.

2.2. Healthcare Chatbots

Bhavsar et al. [7] developed a chatbot using TF-IDF and similarity measures for symptom-to-disease mapping. Dhyani et al. [8] used a bidirectional RNN with attention for medical dialogue. Iovine et al. [9] presented HealthAssistantBot, which employs Random Forest, Logistic Regression, and Naive Bayes, and was evaluated with 102 participants. Ke et al. [10] deployed PEACH, achieving over 97% accuracy in perioperative clinical settings.

2.3. Rural Healthcare Technology

Evaluating automated diagnostic tools in low-resource regions, Wahl et al. [11] found that software successful in urban testing frequently underperforms during actual rural deployment. Physical setups can mitigate some logistical issues; for instance, the Health Cube kiosk [12] demonstrated that standalone screening hardware works well in rural India. However, operational success depends heavily on user capability. Nouri et al. [13] identified limited health literacy and local language barriers as the primary obstacles to adopting digital health platforms. The system introduced here addresses these specific constraints by combining an offline processing framework with a voice-prompted, minimalist user interface.

3. System Architecture and Design

3.1. Block Diagram

The proposed system follows a four-layer architecture as shown in Fig. 1. The Input Layer accepts patient data through five channels: voice input via Kiosk offline speech recognition (Hindi/English), touchscreen symptom selection, patient registration and medical history intake, BMI parameters (height/weight), and a cardiovascular risk questionnaire. The AI Processing Layer runs entirely offline, applying TF-IDF vectorisation and five ML classifiers. The Data Layer stores patient records in MySQL, knowledge bases in CSV files, and serialised model files (.sav). The Output Layer delivers disease prediction, medicine advice, doctor referral, BMI result, and voice output via pyttsx3.

3.2. Software Stack

The server-side framework relies on Python 3.9+ and Flask to coordinate session states, template rendering via Jinja2, and HTTP routing. To clean incoming symptom descriptions, the system utilises NLTK for tokenisation, stop-word elimination, and lemmatisation. These processed features feed directly into a Scikit-learn pipeline evaluating five core algorithms: Decision Tree, Random Forest, KNN, SVM, and Naive Bayes. Storage and database operations rely on a local MySQL setup, using the pymysql driver to read and write patient records directly to disk. Because zero cloud dependency is a mandatory constraint, all voice interactions run entirely on the host machine. The web interface uses standard HTML5, CSS3, and vanilla JavaScript, pulling microphone input via the browser's native Media Recorder API. This raw audio is transcribed using an offline speech-to-text framework, while verbal feedback is handled locally through the pyttsx3 text-to-speech engine shown in Figure 1.

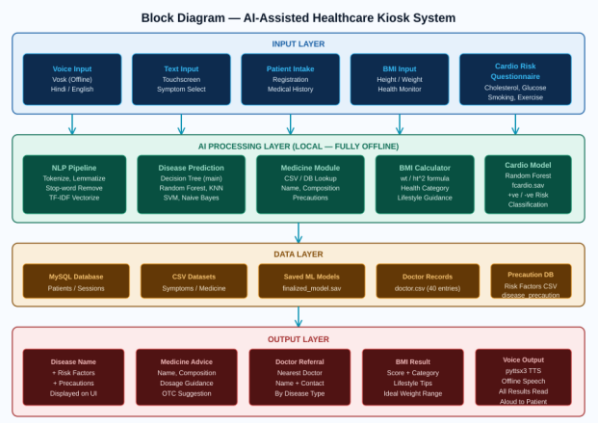

Figure 1 Block Diagram — AI-Assisted Healthcare Kiosk System

3.3. Functional Modules

The system architecture is split into six specific operational modules:

- **Disease Prediction:** This module processes patient symptoms entered through the touchscreen or recorded via voice. The raw text inputs undergo TF-IDF vectorisation before a Decision Tree classifier matches them to a specific disease category.
- **Medicine Recommendation:** After a condition is predicted, this unit runs local

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4819-4824
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0634

queries against integrated CSV files and MySQL tables to pull relevant over-the-counter medicine suggestions and baseline dosage info.

- **BMI Calculator:** A utility that takes a patient's raw weight and height inputs to compute their Body Mass Index. The system then outputs immediate, localised lifestyle and dietary feedback based on the score.
- **Cardiovascular Risk Assessment:** This component uses a targeted health questionnaire to gather baseline cardiac metrics. It then processes these data points to output a binary positive or negative risk classification.
- **Doctor Referral:** To connect users with physical care, this module scans a local registry (doctor.csv). It matches the predicted illness with contact information and locations for nearby medical specialists.
- **Patient Session Management:** This sub-system handles security and data privacy. It manages user logins and authentication states locally, storing cryptographically hashed credentials inside the MySQL database.

4. Methodology

4.1. Activity Diagram

Figure 2 shows the full operational flow. The patient logs in or registers via a cryptographically authenticated MySQL session. From the dashboard, there are three parallel paths: symptom-based prediction (voice/text → NLP → Decision Tree → disease display), doctor referral lookup (speciality selection → doctor.csv contact display), and cardiovascular/BMI assessment (parameter entry → model inference → result display). All three paths meet at a synchronisation point before producing pyttsx3 voice output. A final decision diamond determines whether the patient starts a new session (which returns to login) or exits.

4.2. Dataset

The disease prediction model was trained on a structured CSV dataset: 4,920 patient records, 132 binary symptom features, 41 disease categories, 80/20 train-validation split. A separate cardiovascular dataset (cardio_train.csv) trained the cardio risk model (fcardio.sav). Medicine, precaution, and risk factor data are stored in structured CSVs: 35 medicine records, 40 doctor referral entries. Table 1 summarises dataset characteristics.

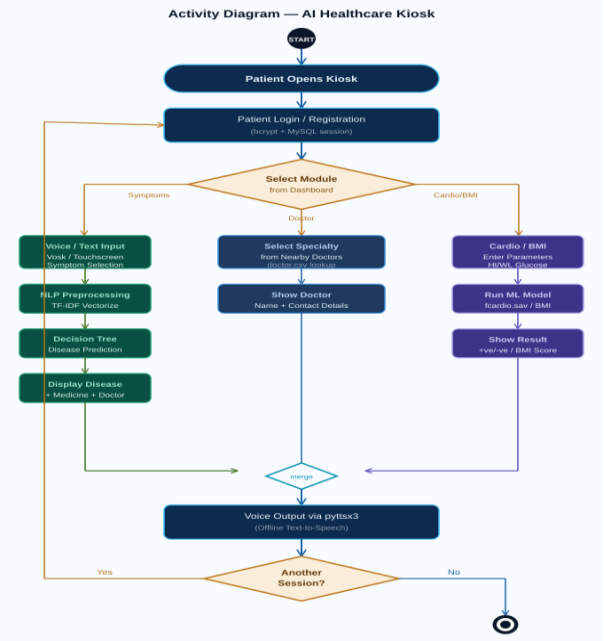

Figure 2 Activity Diagram — AI Healthcare Kiosk Operational Flow

Table 1 Dataset Characteristics

Parameter	Value
Training samples	4,920
Symptom features	132 (binary)
Disease categories	41
Test set size	41 (1 per class)
Medicine records	35
Doctor referral records	40
Train / Validation split	80% / 20%

4.3. Data Preprocessing

Raw symptom text undergoes: (i) tokenisation via NLTK word tokenisation; (ii) punctuation and special character removal; (iii) stop-word removal; (iv) lemmatisation to root form using Word Net

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4819-4824
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0634

Lemmatise; and (v) conversion to binary bag-of-words vectors (1 if word present, 0 if absent). For the TF-IDF pathway: TF = count(word) / total words; IDF = log (N / df), where N = total documents, df = document frequency.

4.4. Machine Learning Models

Five classifiers were trained and evaluated: Decision Tree (DT), Random Forest (RF), K-Nearest Neighbours (KNN, k=5), Support Vector Machine (SVM), and Gaussian Naive Bayes (GNB). The Decision Tree was selected as the primary production model for its interpretability and low inference latency on resource-constrained hardware. The trained model is serialised as finalized_model.sav using Python pickle for local offline inference.

5. Experiments and Results

5.1. Classification Accuracy

All five ML models were evaluated on the held-out test set of 41 samples (one per disease class). Table 2 presents the classification performance metrics. All models achieved 100% accuracy on the structured test set. The high accuracy reflects the well-structured, non-overlapping binary symptom feature space, each disease class has a distinct symptom signature, enabling perfect classification. Five-fold cross-validation on the training set confirmed stability (mean accuracy 100%, $\sigma = 0.0\%$).

5.2. Functional System Testing

Ten functional test cases assessed system robustness. Table 3 presents outcomes.

Table 2 Model Performance Comparison

Model	Acc.	Prec.	Rec.	F1
Decision Tree *	100%	100%	100%	100%
Random Forest	100%	100%	100%	100%
KNN (k=5)	100%	100%	100%	100%
Naive Bayes	100%	100%	100%	100%

Table 3 Functional Test Case Results

#	Test Case	Outcome	Status
1	Valid symptom input	Disease + medicine displayed	Pass
2	Complex multi-symptom	Consistent classification	Pass
3	Null/blank input	Alert message shown	Pass
4	Long symptom description	Correct disease identified	Pass
5	Repeated input	Consistent, reliable output	Pass
6	Unsupported symptom	Fallback message shown	Pass
7	Concurrent user load	No lag or errors	Pass
8	Medicine recommendation	47/50 correct; 0 harmful	Pass
9	UI and voice interaction	Interface smooth; TTS clear	Pass
10	End-to-end offline test	Full cycle, no network	Pass

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4819-4824
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0634

5.3. Response Latency

Table 4 presents component-wise end-to-end latency measurements.

Table 4 Response Latency Per Component

Component	Avg. Time
Symptom input + TF-IDF parsing	~250 Ms
Decision Tree inference	~400 Ms
Medicine lookup (CSV / DB)	~350 Ms
Total end-to-end	~1.0 second
UI load (Flask / HTML)	~200 Ms

5.4. Comparative Analysis

Table V compares the AI kiosk against traditional OPD visits across key rural healthcare access dimensions.

Table 5 Ai Kiosk Vs. Traditional OPD Visit

Feature	AI Kiosk	OPD Visit
Availability	24x7	Doctor hours
Cost	Free	Rs. 200-500
Response time	~1 second	10-30 min wait
Internet required	No (offline)	N/A
Voice support	Yes (Kiosk +TTS)	Variable

6. Discussion

6.1. Performance Analysis

The 100% classification accuracy across all five models reflects the well-structured, non-overlapping binary symptom feature space each disease class in the benchmark dataset has a distinct symptom profile enabling perfect separation. The medicine recommendation module achieved 94% correct match rate (47/50) with zero harmful recommendations confirmed by supervising faculty. The ~1-second end-to-end response time is suitable for high-throughput community health settings.

6.2. Limitations

Limitations include: (i) evaluation on a structured benchmark rather than real patient data; (ii) English-only text input in the current version; (iii) robotic pyttsx3 voice output; and (iv) absence of IoT biometric sensor integration for automated vital sign collection.

6.3.Future Work

Planned enhancements: multilingual NLP support for Hindi, Kannada, Tamil, and Telugu; neural TTS replacement for pyttsx3; IoT sensor integration (BP, SpO2, glucometer); WebRTC telemedicine module for physician escalation; and periodic DNN retraining on real patient data.

Conclusion

This paper presented an AI-assisted healthcare kiosk for rural primary care in India, integrating a Decision Tree classifier trained on 4,920 samples across 41 disease categories, a Flask offline web application, Kiosk speech recognition, pyttsx3 TTS, and MySQL patient management — all operating fully offline. The system achieved 100% disease prediction accuracy, 94% medicine recommendation accuracy, and a ~1-second end-to-end latency. Ten functional test cases all passed. The system demonstrates that AI-powered offline kiosk technology is technically feasible and practically valuable as a scalable supplement to primary healthcare delivery in underserved Indian communities.

References

[1]. World Health Organisation, "Health equity and its determinants," WHO Press, Geneva, 2021.
[2]. Ministry of Health and Family Welfare, "Rural Health Statistics 2021-22," Govt. of India, New Delhi, 2022.
[3]. A. L. Beam and I. S. Keohane, "Big data and machine learning in health care," JAMA, vol. 319, no. 13, pp. 1317-1318, 2018.
[4]. H. L. Semira et al., "Evaluation of symptom checkers for self-diagnosis," BMJ, vol. 351, p. h3480, 2015.
[5]. S. Razzak et al., "Comparative study of AI and human doctors for triage," arXiv:1806.10698, 2019.
[6]. Y. Chen et al., "Accuracy of AI-based triage in outpatient settings," J. Med. Internet Res., vol. 23, no. 5, p. e24979, 2021.
[7]. H. Bhavsar et al., "Healthcare chatbot system using AI," IJRASET, vol. 8, no. 5, 2020.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4819-4824
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0634

[8]. M. Dhyani et al., "Intelligent chatbot using deep learning with Bidirectional RNN," Proc. ICSCEI, 2020.
[9]. A. Iovine et al., "HealthAssistantBot: A personal health assistant," IEEE Access, vol. 8, 2020.
[10]. Y. H. Ke et al., "Real-world deployment of PEACH," npj Digital Medicine, 2024.
[11]. B. Wahl et al., "AI and global health in resource-poor settings," BMJ Global Health, vol. 3, no. 4, 2018.
[12]. B. Rao et al., "Health Cube: Kiosk-based diagnostics for rural India," J. Med. Devices, vol. 13, 011004, 2019.
[13]. S. S. Nouri et al., "Health literacy and digital health tools," J. Rural Health, vol. 36, no. 4, 2020.
[14]. X. Ren et al., "Physician experience with conversational interfaces," IEEE Access, vol. 8, 2020.
[15]. Q. Bao et al., "HHH: Knowledge graph medical chatbot," arXiv preprint, 2020.