International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Cure AI - AI-Powered Care at Your Fingertips

Ashish Vishwakarma
Dept. of Information Technology
Inderprastha Engineering College
Ghaziabad, India

Swati Goel
Assistant Professor, Dept. of IT
Inderprastha Engineering College
Ghaziabad, India
swati.goel@ipec.org.in

Somay Sharma
Dept. of Information Technology
Inderprastha Engineering College
Ghaziabad, India

Mayank Kumar Jha
Dept. of Information Technology Inderprastha Engineering College Ghaziabad, India

Govind Sharma
Dept. of Information Technology Inderprastha Engineering College Ghaziabad, India

***Abstract*—CureAI is a machine learning-based disease prediction system designed to enhance early diagnosis and medical decision-making through the power of data-driven insights. The model utilizes a Random Forest classifier to predict potential diseases based on user-input symptoms, offering a reliable and scalable solution to support healthcare professionals and patients alike. In developing CureAI, a carefully curated dataset of symptoms and corresponding diseases was used, emphasizing data preprocessing, feature engineering, and model evaluation to ensure accuracy and generalizability.**

The model achieved high classification performance by leveraging the ensemble learning nature of Random Forest, which reduces overfitting and handles high-dimensional symptom data effectively. CureAI is deployed through a Flask-based REST API, enabling easy integration with web or mobile front-ends. This architecture ensures real-time predictions with low latency and broad accessibility.

The motivation behind CureAI lies in addressing the gap between the onset of symptoms and timely medical consultation, particularly in resource-limited settings. By offering immediate, preliminary diagnostic insights, CureAI empowers individuals to seek professional care proactively, potentially leading to earlier treatment and better outcomes.

The system is designed with scalability and interpretability in mind, making it suitable for extension into more complex health prediction frameworks or integration with electronic health records. With continued improvements in dataset quality and user interface design, CureAI has the potential to serve as a vital tool in preventive healthcare, disease surveillance, and health education.

I. INTRODUCTION

The increasing demand for accessible and efficient healthcare solutions has led to the integration of intelligent technologies in medical diagnostics. Among the common barriers to timely treatment are delayed diagnoses, lack of awareness about symptoms, and the unavailability of medical professionals in remote or underserved areas. To address these challenges, we propose CureAI [7] [11], a machine learning-based disease prediction system designed to assist individuals in identifying potential illnesses based on symptoms provided in natural language.

CureAI aims to empower users by offering a quick and user-friendly way to obtain probable disease predictions and precautionary advice. The system is built on a Random Forest Classifier, a robust machine learning algorithm known for its high accuracy in classification tasks [1]. To make the interaction more natural, Natural Language Processing (NLP) techniques using the spaCy library are incorporated to extract relevant symptoms from user-input text. These symptoms are then converted into a binary input vector and passed to the trained model for prediction.

One of the significant features of CureAI is its ability to handle unstructured inputs, making it more accessible to non-technical users. The backend is trained on curated datasets containing symptom-disease mappings and associated precautions, ensuring that the predictions are grounded in real medical knowledge. Once a disease is predicted, the system retrieves the recommended precautions, providing immediate and actionable health advice.

CureAI is envisioned not as a replacement for professional diagnosis but as a supportive tool that promotes health awareness and encourages timely medical consultations. Its lightweight implementation and web integration capabilities make it scalable and adaptable to various platforms, including mobile and web applications. With continued development [13], CureAI holds the potential to bridge gaps in primary healthcare access and serve as a valuable aid in digital health initiatives.

II. LITERATURE REVIEW

Machine learning, especially Random Forest, is widely used for disease prediction due to its accuracy and robustness. NLP techniques help extract symptoms from user input, improving usability. Deploying models via APIs enables real-time predic- tions, though challenges like data quality and ethics remain.
[3] [5]

A. Machine Learning in Healthcare

Recent advancements have demonstrated the efficacy of ma- chine learning in predicting diseases, identifying medical pat- terns, and supporting clinical decision-making. [7] Algorithms like decision trees, SVMs, and neural networks have been applied across various domains, but Random Forests remain a popular choice due to their robustness, interpretability, and resistance to overfitting.

B. Symptom-Based Disease Prediction Models

Several studies have focused on mapping symptoms to prob- able diseases using supervised learning. Traditional systems relied on rule-based expert systems [12], but recent research

favors data-driven approaches. However, these models often struggle with real-world variability in symptom description and require effective feature extraction methods.

C. Natural Language Processing for Medical Texts

NLP has gained traction for processing unstructured medical data, including symptom descriptions, patient histories, and clinical notes. Techniques like tokenization, lemmatization, and named entity recognition (NER) have been employed to extract medical terms. [2] [6] CureAI builds on this by using a rule-based NLP approach to extract symptoms from plain English, enhancing accessibility.

D. Deployment of Predictive Models via APIs

Transforming machine learning models into real-world applications requires effective deployment. Literature highlights the importance of lightweight, scalable solutions using REST APIs. Frameworks like Flask and FastAPI have been widely adopted for integrating ML models with frontend systems for real-time interaction.

E. Challenges and Ethical Considerations

Key challenges include data quality, model generalization across populations, and the risk of over-reliance on AI without clinical oversight. Ethical concerns revolve around data privacy, patient consent, and model transparency, especially in critical domains like healthcare. In summary, the literature highlights the effectiveness of machine learning, especially Random Forest, in disease prediction due to its accuracy and robustness. Natural language processing techniques play a key role in extracting symptoms from unstructured user input, improving system usability. [10] Deploying models via REST APIs enables real-time predictions and easy integration, though challenges like data quality and ethical considerations remain critical.

III. METHODOLOGIES

A. Disease Prediction Model

The core of CureAI is a machine learning-based disease prediction model that analyzes user symptoms to predict likely diseases. We use the Random Forest classifier for its high accuracy, robustness against overfitting, and interpretability. This ensemble method builds multiple decision trees and aggregates their results to improve predictive performance.

- Uses Random Forest algorithm.
- Handles complex, non-linear symptom-disease relation- ships.
- Reduces overfitting through ensemble learning.
- Provides feature importance insights.

Fig. 1. Working of CureAi

B. Data Collection and Preparation

Accurate disease prediction requires high-quality data. CureAI's dataset consists of symptom-disease mappings cu- rated and cleaned for machine learning. Data preprocessing involves handling missing values, encoding categorical vari- ables, and feature selection to optimize model training [16].

- Data sourced from reliable medical datasets and medical literature.
- Data cleaning to handle inconsistencies and missing val- ues.
- Encoding categorical symptom data into numerical fea- tures.
- Splitting data into training and testing subsets.

C. Symptom Extraction using Natural Language Processing (NLP)

To interpret user input in plain English, CureAI employs a rule-based NLP approach that extracts relevant symptoms from free-text descriptions [6]. Techniques such as tokenization, lemmatization, and keyword matching against a predefined symptom list enable efficient symptom identification.

- Rule-based keyword matching for symptom extraction.
- Use of tokenization and lemmatization for text normal- ization.
- Predefined symptom vocabulary for matching.
- Designed for quick, accurate symptom detection without heavy models.

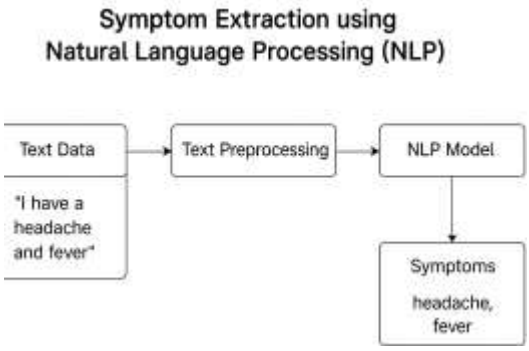

Fig. 2. Symptom Extraction

D. Model Training and Evaluation

CureAI's Random Forest model is trained on the processed dataset, with hyperparameter tuning to optimize performance. Evaluation metrics including accuracy, precision, recall, and

F1-score measure the model's effectiveness in disease prediction. Cross-validation ensures robustness [8].

- Training with labeled symptom-disease data.
- Hyperparameter tuning for number of trees, max depth, etc.
- Evaluation using accuracy, precision, recall, and F1-score.
- Cross-validation to prevent overfitting.

E. Deployment via Flask API

To make the model accessible, CureAI is deployed through a Flask-based REST API. This allows real-time disease prediction by receiving symptoms via HTTP requests and returning predictions in JSON format. The API design supports easy integration with frontend applications.

- Flask REST API to serve the prediction model.
- Endpoint to accept symptoms and return disease predic- tions.
- Lightweight and scalable for quick responses.
- Designed for integration with web/mobile frontends.

F. System Scalability and Future Enhancements

CureAI is designed with scalability in mind, allowing future integration of larger datasets and advanced NLP techniques like Named Entity Recognition [4] [11]. Potential enhancements include a user-friendly frontend, database support, and expanded disease coverage.

- Modular architecture supports scaling.
- Future NLP improvements: custom NER and fuzzy matching.
- Plans for richer datasets and multi-language support.
- Frontend development for better user experience.

IV. IMPLEMENTATION

CureAI is implemented as a lightweight, modular system for predicting diseases based on user symptoms entered in plain English. The system is divided into clear components: user input handling, symptom extraction (NLP), machine learning prediction, and output delivery via an API. The end-to-end design ensures usability, accuracy, and future scalability.

A. System Architecture Overview

CureAI is built around a microservice architecture where the machine learning model is decoupled from the user interface. The backend is powered by Python and served via a Flask API, which connects the NLP module, ML model, and output logic.

- Lightweight, modular design.
- REST API enables platform-independent access.
- Components interact via clearly defined interfaces.

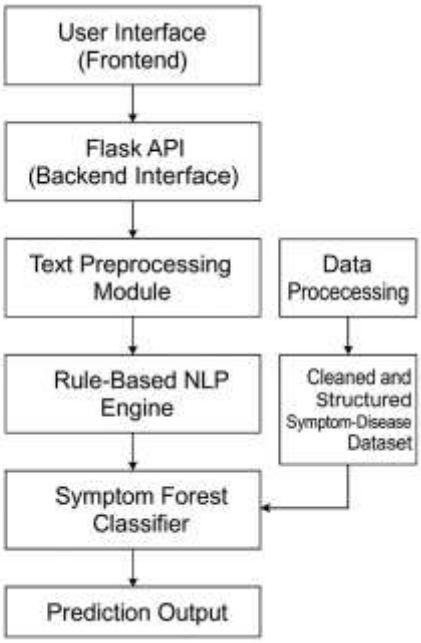

Fig. 3. CureAi Architecture

B. Data Processing Pipeline

The model uses a structured CSV dataset containing symptom-disease mappings. The data is preprocessed before being used to train the ML model.

- Lightweight, modular design.
- REST API enables platform-independent access.
- Components interact via clearly defined interfaces.

C. Symptom Extraction using NLP

To make the system user-friendly, CureAI accepts free-text descriptions like "I have a headache and sore throat." A rule- based NLP model extracts the relevant symptoms from this input.

- Tokenization and lemmatization using spaCy.
- Matching against a predefined symptom vocabulary.
- Custom logic to handle varied input phrases and syn- onyms.
- Output is a clean list of recognized symptoms, formatted for the model.

D. Machine Learning Model (Random Forest)

The cleaned symptom input is fed to a pre-trained Random Forest model for disease prediction [13]. This ensemble method builds multiple decision trees and uses majority voting for final output.

- Pre-trained using sklearn's RandomForestClassifier.
- Model is serialized using joblib or pickle.
- During prediction, symptom inputs are converted to the same format as training data (binary vector).
- The model returns the most probable disease based on learned patterns.

E. Flask API Integration

The entire model logic is wrapped in a Flask API for real-time interaction. The user sends a POST request with symptoms in plain English; the API processes the text, extracts symptoms, predicts the disease, and returns the result as JSON.

- /predict endpoint accepts user input.
- Internally calls NLP + model pipeline.
- Responds with predicted disease and optional confidence score.
- Lightweight and fast, suitable for deployment or local use.

V. EXPECTED OUTCOME

CureAI aims to revolutionize the early detection of diseases through a user-friendly, AI-powered prediction platform. By combining rule-based NLP with a robust Random Forest model, the system enables accessible, real-time diagnosis sup- port — especially for users with minimal medical knowledge.

A. Symptom Extraction via NLP

CureAI offers plain-English input support, allowing users to describe their symptoms naturally. The system accurately maps this input to known medical symptoms using lightweight NLP techniques, making the platform highly accessible.

B. Accurate Disease Prediction

The Random Forest model ensures high prediction accu- racy across multiple diseases. It effectively handles symptom overlap, class imbalance, and irrelevant features, resulting in a more reliable prediction system.

C. Real-Time Response via Flask API

With a lightweight Flask-based REST API, CureAI delivers fast responses suitable for integration into apps, chatbots, or web platforms.

D. Scalability and Extension Possibilities

CureAI is designed with future scalability in mind. Its modular structure allows easy upgrades to NLP, model logic, or frontend interface without major rewrites.

E. Improved Health Awareness and Accessibility

By offering quick, AI-driven health insights, CureAI em- powers users to take early action, especially in remote or underserved areas.

VI. DISCUSSION

A. Interpretability and Trust in Predictions

CureAI is designed to assist users in identifying probable diseases based on their symptoms. While the Random Forest model provides high accuracy, it operates as a black-box algorithm, making it difficult for users to understand how specific symptoms contribute to the diagnosis. Enhancing the model with feature importance visualization or confidence scores could increase transparency and user trust, especially in critical use cases [10].

B. Limitations of Rule-Based NLP

The NLP component of CureAI uses a rule-based approach to extract symptoms from free-form user input. While this method is lightweight and effective for common phrases, it may face challenges in handling ambiguous language, slang, or uncommon symptom descriptions [13]. Although it performs well in structured contexts, integrating more advanced NLP techniques such as intent recognition or context-aware phrase detection could improve accuracy and flexibility.

C. Model Generalization and Dataset Scope

CureAI's model was trained on a specific dataset with predefined symptom-disease mappings. As a result, it may not perform optimally when exposed to diseases or symptom combinations outside the training data [7]. Future iterations of CureAI could incorporate broader datasets, including region- specific health data, to enhance generalization and relevance across diverse user populations.

D. System Performance and Scalability

CureAI's current implementation, powered by a Flask API, supports real-time disease predictions with minimal latency [1]. This setup is effective for development and personal deployment, but scaling to a larger user base would require optimization. Adopting a microservices architecture and de- ploying components to the cloud could significantly enhance performance, reliability, and scalability.

E. User Experience and Practical Use

The system allows users to input symptoms in plain English, making it accessible for non-expert users. However [3], the system currently returns a single predicted disease without pro- viding multiple possibilities or treatment suggestions. Adding multi-disease prediction [12], health tips, or links to trusted sources could make CureAI more practical and informative in real-world applications.

VII. FUTURE SCOPE

A. Enhanced Symptom Interpretation

Future improvements can involve integrating machine learning-based NLP models to handle more complex, vague, or misspelled symptom inputs, making CureAI more accurate and user-friendly.

B. Multi-Disease Prediction

Currently, CureAI outputs a single predicted disease. Adding support for predicting multiple possible diseases with associated confidence scores will provide users with broader diagnostic insights.

C. Dynamic Model Updating

Implementing mechanisms for continuous learning and au- tomatic updates based on new medical data can help keep the model relevant with emerging diseases and evolving symptom patterns.

D. Cloud-Based Scalability

Deploying CureAI on cloud platforms using containerization (Docker) and orchestration (Kubernetes) will support high availability, better load handling, and remote access across devices.

E. User Health History Integration

Incorporating user profiles and basic health history (with consent) could enable more personalized predictions, improving CureAI's relevance and usefulness in real-world scenarios.

VIII. LIMITATIONS

A. Basic Rule-Based NLP

The current rule-based approach is limited in its ability to understand nuanced or unstructured symptom descriptions, which may affect prediction accuracy for varied user inputs.

B. Restricted to Training Data

CureAI can only predict diseases it has been trained on. It cannot recognize or suggest rare, complex, or new illnesses that are absent in the current dataset.

C. No Medical Decision Support Certification

CureAI is a technical project for educational purposes and is not certified for clinical decision-making. Its outputs should not be used for self-medication or treatment without professional consultation.

D. Single Disease Output

The system currently provides only the most probable disease prediction without listing alternatives, which may be limiting in cases where symptoms overlap across conditions.

E. Limited Deployment Readiness

The model and API are suitable for demonstration but are not yet optimized for high-scale production use. Performance may degrade under high concurrent user loads without further optimization.

IX. RESULTS AND ANALYSIS

The CureAI system was evaluated using standard classification metrics to assess the effectiveness of the Random Forest algorithm. The model achieved high accuracy and strong performance across all evaluation parameters. The following sections summarize the results and visual representations of the model's evaluation.

Model Evaluation Table (Performance Metrics)

Metric	Value
Accuracy	98% (0.98)
Precision	0.89
Recall	0.92
F1-Score	0.94
Support	Varies

A. Algorithm Usage Overview

During model development, various machine learning algo- rithms were explored. Random Forest was selected based on its consistent and high performance, handling of imbalanced data, and interpretability for disease prediction tasks.

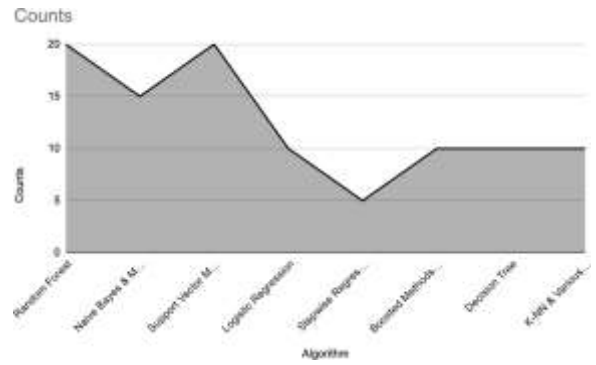

Fig. 4. Comparison of commonly used ML algorithms for disease prediction

B. Confusion Matrix

The confusion matrix below shows the classification results, giving insight into the model's ability to correctly predict disease classes and identify misclassifications.

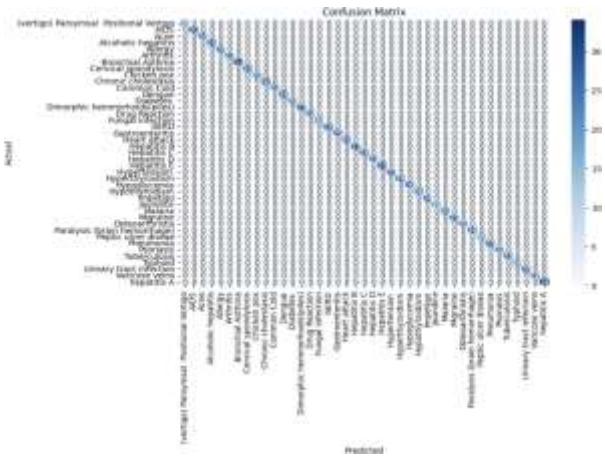

Fig. 5. Confusion matrix of CureAI model predictions

The evaluation results demonstrate that CureAI's Random Forest model performs with high accuracy and reliability in disease prediction. The metrics indicate a well-balanced model with strong generalization capability, suitable for real-world symptom-based diagnostics.

X. CONCLUSION

CureAI presents a robust and efficient solution for early-stage disease prediction using symptom-based inputs. By leveraging a Random Forest classification model and integrating a rule-based NLP system, CureAI can process natural language symptoms and map them accurately to medical

conditions. The system offers an intuitive user interface backed by a Flask API, ensuring real-time prediction and accessibility. The high accuracy and balanced performance across metrics demonstrate the model's reliability. CureAI is designed for educational and research purposes and shows great potential for real-world applications with further enhancement. Future integration with health history and cloud deployment can further extend its practical utility.

A. CureAI User Interface

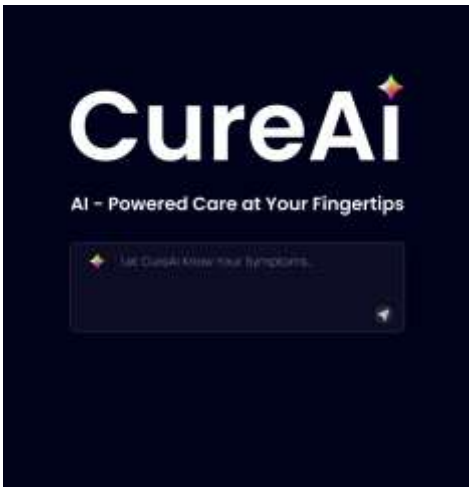

Fig. 6. Graphical User Interface of CureAI

B. Disease Prediction Output

Fig. 7. Sample output of predicted disease from user symptoms

REFERENCES

[1] V. Jackins, S. Vimal, M. Kaliappan, and M. Y. Lee, "AI-based smart prediction of clinical disease using random forest classifier and Naive Bayes."
[2] S. Biradar and Dr. S. Shastri, "Medical Chatbot: AI Based Infectious Disease Prediction Model."
[3] S. Chakraborty, H. Paul, S. Ghatak, S. K. Pandey, A. Kumar, K. U. Singh, and M. A. Shah, "An AI-Based Medical Chatbot Model for Infectious Disease Prediction."
[4] K. Kabore, O. Sawadogo, Y. Traore, and J. Thiombiano, "Towards a Chatbot for Medical Diagnosis Based on Patient Symptoms."
[5] J. V. Anchitadagammai, S. Kavitha, S. Murali, L. Gururaj, and J.
P. Pranav Kiruthik, "Predictive Health Assistant: AI-Driven Disease Projection Tool."
[6] G. S. Sannala, K. V. G. Rohith, A. G. Vyas, and C. R. Kavitha, "Explain- able Artificial Intelligence-Based Disease Prediction with Symptoms Using Machine Learning Models."
[7] F. Sogandi, "Identifying diseases symptoms and general rules using supervised and unsupervised machine learning."
[8] T. Bhanuteja, K. V. N. Kumar, K. S. Poornachand, C. Ashish, and P. Anudeep, "Symptoms Based Multiple Disease Prediction Model using Machine Learning Approach."
[9] S. Dhanka and S. Maini, "Random Forest for Heart Disease Detection: A Classification Approach," in *Proc. IEEE Int. Conf. on Power, Energy, Environment and Research (ICEPEER)*, Dec. 2021. DOI: 10.1109/ICEPES52894.2021.9699506
[10] M. M. Ahsan, S. A. Luna, and Z. Siddique, "Machine-Learning-Based Disease Diagnosis: A Comprehensive Review," *Healthcare*, vol. 10, no. 3, p. 541, Mar. 2022. DOI: 10.3390/healthcare10030541
[11] X. Luo *et al.*, "A Deep Language Model for Symptom Extraction from Clinical Text and Its Application to Extract COVID-19 Symptoms from Social Media," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 4, pp. 1737–1748, Apr. 2022. DOI: 10.1109/JBHI.2021.3123192
[12] E. Sezgin *et al.*, "Extracting Medical Information From Free-Text and Unstructured Patient-Generated Health Data Using Natural Language Processing Methods: Feasibility Study With Real-World Data," *JMIR Formative Res.*, vol. 7, 2023. DOI: 10.2196/43014
[13] K. V. Rao *et al.*, "Leveraging Flask API and Machine Learning to Forecast Multiple Diseases," *Commun. Appl. Nonlinear Anal.*, vol. 32, no. 1S, Oct. 2024. DOI: 10.52783/cana.v32.2145
[14] L. Tang, J. Li, and S. Fantus, "Medical Artificial Intelligence Ethics: A Systematic Review of Empirical Studies," *Digit. Health*, vol. 9, 2023.
DOI: 10.1177/20552076231186064
[15] I. D. Mienye, "A Survey of Bias and Fairness in Healthcare AI," in *Proc. IEEE Int. Conf. Healthcare Informatics (ICHI)*, Jun. 2024. DOI: 10.1109/ICHI61247.2024.00103
[16] N. Lalwani *et al.*, "Survey on Ethical Challenges of Implement- ing AI in Healthcare," in *Proc. IEEE Int. Conf. on Pervasive Ar- tificial Intelligence and Radio Intelligence (PraI)*, Aug. 2024. DOI: 10.1109/PRAI62207.2024.10827739