



Sanjeevini Bot Using Machine Learning

Abdul Azeem Sheikh¹, Mrs. Manasa S.M², Chinmay Hegde AJ³, B Aman Ulla Khan⁴, and Harshith DR⁵

Student, Department of Information Science and Engineering, Jawaharlal Nehru New College of
Engineering,
Shivamogga, India^{1,3,4,5}

Assistant professor, Department of Information Science and Engineering, Jawaharlal Nehru New College of
Engineering, Shivamogga, India²,

Abstract: The Sanjeevini Bot is an AI-driven virtual healthcare assistant designed to improve accessibility and efficiency in healthcare. It utilizes machine learning techniques, including a Random Forest classifier and TF-IDF vectorization, to predict medical conditions based on user-reported symptoms, recommend medicines, and connect users to healthcare professionals. Its features include natural language processing for seamless interaction, a user-friendly interface, and text-to-speech functionality for enhanced accessibility. With a robust medical database and personalized recommendation engine, the bot provides accurate, timely guidance, bridging gaps in healthcare delivery. Testing demonstrates its high reliability in diagnosing conditions and offering relevant medical advice.

Future advancements include IoT-based real-time health monitoring, multilingual support, expanded medical databases, and telemedicine capabilities for direct consultations. By integrating advanced technology with user-centric design, the Sanjeevini Bot addresses critical gaps in healthcare access, particularly in underserved areas, and sets the stage for scalable, inclusive healthcare solutions.

Index Terms - Sanjeevini Bot, Virtual Healthcare Assistant, Machine Learning, Artificial Intelligence, Random Forest Classifier, TF-IDF Vectorization, Natural Language Processing (NLP), Text-to-Speech (TTS), Disease Prediction, Personalized Medical Advice, Recommendation Engine, IoT Health Monitoring, Multilingual Support, Telemedicine, Healthcare Accessibility, Symptom Analysis, Medical Database, Healthcare Chatbot, Real-time Assistance, Data Privacy and Security, User-friendly Interface, Inclusive Healthcare, Wearable Integration, Diagnosis Optimization, Predictive Analytics.

I. INTRODUCTION

Sanjeevini Bot is a new concept of intelligent virtual medical assistant designed to increase the accessibility and efficiency of medical services. As technology changes the healthcare sector, the program aims to use artificial intelligence and machine learning to provide personalized medical advice. The robot predicts diseases based on the symptoms reported by the user, recommends the appropriate medicine, and connects the user with the doctor using the user's understanding. Integrating natural language processing (NLP) for seamless communication and text-to-speech (TTS) technology for easy access, the system allows users from different cultures to interact easily. A random forest classifier is a machine-learning algorithm designed to predict diseases accurately. The model is trained on rich data, analyses user input, and provides recommendations for diagnosis. The bot uses TF-IDF vectorization and text normalization to optimize the treatment message, ensuring that it can correctly interpret user queries.

A recommendation engine is incorporated into the system to provide personalized drug recommendations based on treatment from medical records. The bot also bridges the gap between patients and doctors by enabling doctors to refer customers to good doctors. A great way to solve real-world problems. From the user-friendly Graphical User Interface (GUI) to the AI-powered power, every part of the system is carefully designed to provide a good user experience. Interactive conversation increases accessibility for people who have difficulty with traditional texts, such as the visually impaired or users who prefer to chat. Make it an

essential tool in treatment. The main goal is to be a collaborative healthcare assistant who can understand customer needs, process medical information, and provide appropriate recommendations. The system can predict medical conditions, recommend appropriate medications, and provide information about doctors for consultation. These features, combined with a powerful interface, enable Sanjeevini Bot to meet a wide range of customer needs, making healthcare more inclusive and responsive. The management provides relevant recommendations.

The engine not only provides personalized medical advice but also provides easy access to doctors categorized by specialty. Ability to identify symptoms and provide customized solutions to enhance the bot's ability to solve complex questions. The benefits of robots can be further enhanced by integrating features such as real-time health monitoring through IoT integration, multimodal support messaging for a wider audience, and telemedicine capabilities for virtual consultation. These improvements will make the robot more versatile, flexible, and usable by a wide range of users. By combining technology with a user-centric interface, Sanjeevini Bot provides a solid foundation for building smart machines that adapt to be user-friendly. It demonstrates the potential of AI-powered tools to address healthcare inequality, especially for underserved and remote populations. Medical assistance. Its accurate predictions, personalized recommendations, and user-friendliness make it useful in improving healthcare. By addressing current challenges and embracing future success, Sanjeevini Bot aims to revolutionize healthcare and make healthcare accessible to all.

II. BACKGROUND

The Sanjeevini Bot project emerges as a response to the growing need for accessible and efficient healthcare solutions, especially in underserved regions. Advances in artificial intelligence (AI) and machine learning (ML) have paved the way for intelligent virtual assistants capable of addressing critical challenges in medical service delivery. Sanjeevini Bot aims to leverage these technologies to offer personalized medical guidance, ensuring that users can access essential healthcare information anytime, anywhere.

Central to the system is the integration of the **Random Forest classifier**, which accurately predicts diseases based on user-reported symptoms. Complemented by text pre-processing methods like tokenization, TF-IDF vectorization, and symptom standardization, the bot ensures that user inputs are interpreted accurately. A robust database storing information on medical conditions, medicines, and doctors enhances the bot's ability to provide relevant recommendations.

The project prioritizes inclusivity by incorporating a user-friendly graphical interface and text-to-speech functionality, making it accessible to individuals with varying technical proficiency or disabilities. It addresses key challenges such as the handling of ambiguous user inputs and the secure management of sensitive health data, ensuring reliability and user trust.

Sanjeevini Bot exemplifies how AI-driven healthcare assistants can bridge gaps in medical access by providing real-time assistance for symptom analysis, medication suggestions, and referrals to healthcare professionals. This innovation not only improves healthcare delivery but also highlights the potential for scalable, adaptive solutions in addressing the global demand for equitable and efficient healthcare services.

III. PROBLEM DESCRIPTION

In today's fast-paced environment, businesses and organizations are increasingly challenged with managing effective communication. Sectors such as healthcare, customer support, and education are particularly burdened by a high volume of inquiries and tasks, consuming substantial time and resources. Human operators often find it difficult to keep pace, leading to delays, diminished customer satisfaction, and escalated operating costs. The absence of 24/7 support further complicates matters, as users now expect prompt access to information and assistance.

Current chatbot solutions frequently fall short of expectations. They often necessitate significant customization, which restricts flexibility and adaptability across different systems. Furthermore, concerns surrounding user data security persist, particularly with breaches that threaten trust and credibility. This situation underscores the urgent need for intelligent, customizable, and secure chatbot solutions that effectively address these challenges while promoting user satisfaction and boosting productivity.

IV. LITERATURE SURVEY

[1] “Personal Healthcare Chatbot for Medical Suggestions Using Artificial Intelligence and Machine Learning”

Authors: R. Jegadeesan et al. (2023)

This study focuses on developing a healthcare chatbot that uses AI and ML, particularly NLP, to analyze user-reported symptoms and provide medical suggestions. The chatbot classifies ailments as severe or minor, offering appropriate advice such as prescriptions, Ayurvedic remedies, and homeopathy treatments for minor conditions while recommending doctors for severe cases. It uses a retrieval-based algorithm with a disease prediction accuracy of 82%. Key features include a user-friendly text-based interface, instant medical guidance, and secure login. Operating 24/7, the chatbot enhances user experience by reducing healthcare facility visits for minor issues.

The system offers high prediction accuracy, ensures reliable guidance, and operates round-the-clock to provide immediate assistance. It supports diverse medical preferences, including alternative remedies, enhancing user convenience. However, it is limited to predefined datasets, reducing adaptability for complex cases. The chatbot relies heavily on user-reported symptoms, which may lack accuracy and does not include real-time health monitoring or integration with wearable devices.

[2] “Medical Information Extraction with NLP-Powered QA Bots”

Authors: Claudio Crema et al. (2024)

This paper introduces the NLP Extraction and Management Tool (NEMT), which combines rule-based regex methods with a BERT-based Question Answering Bot (QABot) to automate clinical data extraction from unstructured documents. Deployed locally in hospitals, the system ensures compliance with privacy regulations like GDPR and HIPAA. It is tailored for clinical research, reducing data extraction time by 86%. By organizing extracted information into a centralized database, the tool supports collaboration and improves research efficiency.

The tool achieves high accuracy in data extraction, significantly reducing manual workload and enabling collaboration. Its compliance with GDPR and HIPAA ensures data privacy. However, it is limited to dementia-related topics and requires adaptation for other domains. The deployment demands advanced infrastructure and trained personnel, which may not be available universally, and struggles with handling unstructured or unconventional text formats.

[3] “Healthcare Chatbot”

Authors: Athulya N et al. (2021)

This paper explores a chatbot for predicting diseases and offering basic medical advice using AI and ML techniques. The Decision Tree algorithm predicts diseases based on user-reported symptoms, responding to text and voice formats. Designed for accessibility and cost-effectiveness, it enables instant assistance, particularly in underserved areas. Achieving a 78.24% accuracy rate, the chatbot minimizes dependence on doctors for minor consultations, thereby saving time and resources.

The chatbot delivers reliable disease predictions, supports cost-effective healthcare delivery, and ensures 24/7 availability through text and voice interaction. However, its limited dataset restricts the diagnosis of rare conditions. It depends solely on user-reported symptoms, increasing the risk of diagnostic errors, and lacks real-time health monitoring or wearable integration.

[4] “Healthcare Chatbot System”

Authors: Mark Lawrence et al. (2024)

This system provides accessible medical assistance using Python-based technologies. Employing algorithms like Decision Trees and SVM, it predicts diseases, offers tailored medical solutions, and recommends doctors. The chatbot prioritizes ethical considerations like data security and user trust, achieving a prediction accuracy of 78%. Its scalable design and consistent performance ensure usability in underserved areas.

The system combines reliability with privacy compliance, providing tailored advice and doctor recommendations. Its round-the-clock operation ensures uninterrupted service. However, dataset limitations restrict the diagnosis of rare conditions, and it struggles with maintaining conversational flow in multi-turn interactions. Additionally, it lacks integration with wearable health devices.

[5] "AI-Based Healthcare Chatbot System"

Authors: M.V. Patil et al. (2021)

This chatbot addresses healthcare accessibility challenges in rural areas, using NLP and decision tree algorithms for disease diagnosis and medical advice. It reduces dependency on doctors for minor issues and provides users with comprehensive medical information. Its ability to gather data from external sources like search engines expands its knowledge base.

The chatbot enhances healthcare accessibility in underserved regions and provides comprehensive symptom information to improve user understanding. It is customizable for extended functionality. However, it is limited in diagnosing rare conditions, requires internet connectivity for optimal operation, and lacks real-time health monitoring capabilities.

[6] "AI Healthcare Chatbot"

Authors: Manali Jain et al. (2024)

This paper presents an ANN-powered chatbot designed to diagnose health conditions using user-reported symptoms. Leveraging deep learning, NLP, and frameworks like Flask, the system facilitates real-time interaction and offers potential integration with Android applications. It continuously improves through data updates and incorporates advanced features like emergency services.

The chatbot reduces dependency on doctors for basic queries, facilitates real-time conversations, and allows for future integration with advanced applications like emergency response systems. However, it demands substantial computational resources for training and maintenance, faces risks of shallow responses compared to professional consultations, and requires robust data protection to ensure security.

[7] "Good Fellow: A Healthcare Chatbot System"

Authors: Palak Dohare et al. (2023)

This chatbot integrates disease diagnosis with symptom management using voice and text interaction, session history storage, and GPS-based doctor recommendations. Its design enhances healthcare accessibility and efficiency, catering to underserved areas. Additional features include multimedia inputs and tools for medical professionals, such as session summaries.

The system enhances user convenience with multimedia input support and GPS-based doctor recommendations. It provides session history for both users and healthcare providers, improving interaction continuity. However, it struggles with diagnostic accuracy for complex cases, relies heavily on user input precision, and does not support real-time interaction with medical professionals.

[8] "Healthcare Chatbot System Using Artificial Intelligence"

Authors: Nikita Vijay Shinde et al. (2021)

This chatbot uses advanced algorithms like TF-IDF and cosine similarity to predict diseases and provide personalized health advice. It includes medication reminders and supports minor condition management, offering a lightweight design suitable for emergencies. The system ensures rapid response and reduces dependency on healthcare facilities for non-critical issues.

The chatbot enables instant assistance for minor ailments, reduces healthcare costs, and supports personalized treatment suggestions. However, it has limited diagnostic capabilities for complex diseases, relies heavily on English for communication, and lacks interaction with healthcare professionals or integration with wearable health devices.

[9] "Machine Learning-Based Healthcare Chatbot"

Authors: Rhythm Goel et al. (2022)

This chatbot employs ML and NLP techniques to predict diseases based on user-reported symptoms. It leverages a database of common symptoms and conditions to provide tailored recommendations based on severity, offering a conversational interface and real-time guidance.

The system enhances healthcare accessibility for underserved areas, provides private and instant assistance, and reduces dependency on human operators for minor health concerns. However, it is limited to predefined datasets for diagnosis, relies on accurate user input, and cannot manage emergency medical scenarios effectively.

[10] "Healthcare Chatbot for Personal Healthcare Assistance"

Authors: Garima Chandel et al. (2024)

This chatbot uses algorithms like Random Forest and SVM for disease prediction, addressing gaps in healthcare access by providing real-time assistance, tailored advice, and doctor referrals. It aims to be inclusive and reliable, with features that cater to diverse user needs.

The chatbot ensures accessibility for remote and underserved users, offers cost-effective solutions for non-critical concerns, and provides round-the-clock support. However, it lacks advanced diagnostic capabilities for complex diseases, depends on user-reported symptoms for accuracy, and cannot handle life-threatening medical scenarios.

V. RESEARCH METHODOLOGY AND SYSTEM ARCHITECTURE

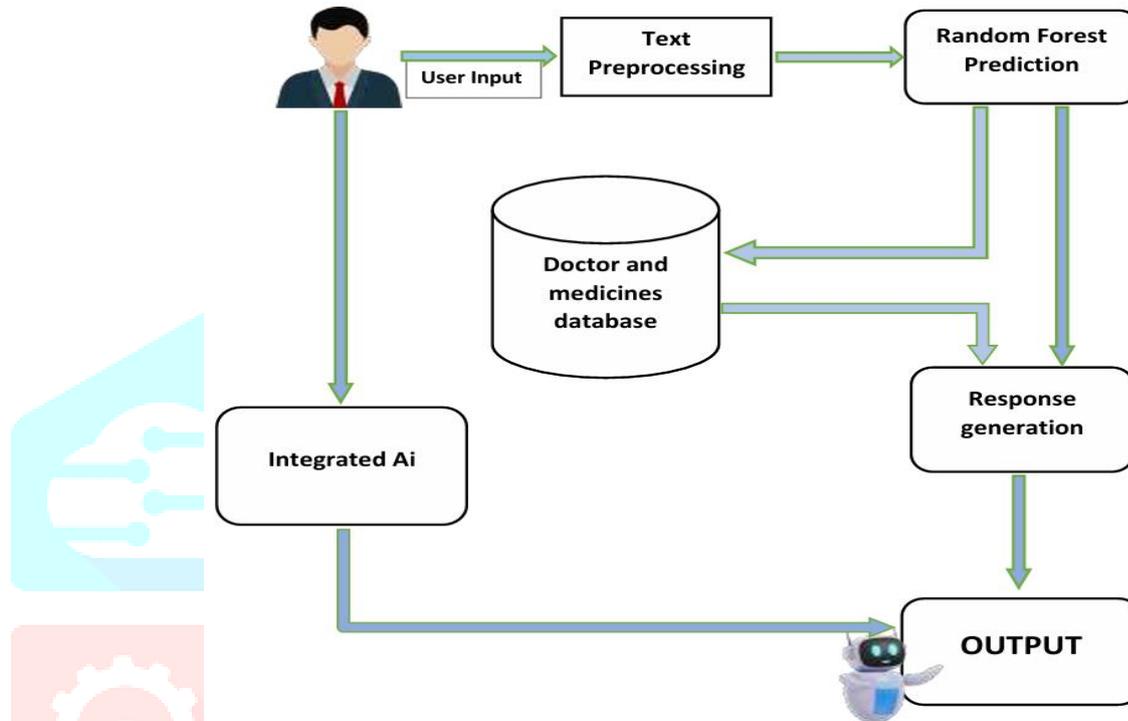


Figure 5.1: System Architecture

Sanjeevini Bot is an intelligent healthcare assistant designed to provide medical guidance by analyzing user inputs. It integrates machine learning, a medical database, and AI-based response generation to offer recommendations, including doctor suggestions and medicine details.

User Input Layer acts as input to the system, allowing users to provide text-based descriptions of symptoms, diseases, or messages asked. This system can provide a relationship that guides users to formulate the right questions, while also using techniques to eliminate meaningless or incomplete information. Features such as easy access, multi-language support for easy access, and error management ensure user interaction.

Once the input is received, the Text Preprocessing Module will convert the raw data into a format suitable for machine analysis. These methods include tokenization to split the input word into smaller pieces, word stop to remove non-uniform words, lemmatization to standardize the data of a word, and noise filter to clean the information. This step allows the system to extract useful patterns that increase the prediction accuracy of subsequent machine-learning models.

The main prediction engine is the Random Forest Classifier, which first processes the data to classify the input according to clinical scenarios. This learning model is trained from specific clinical data and features to identify the most common diseases with high accuracy. The output includes prediction scores and text, which ensures reliability even in the case of noise or minimal overlap.

To support its recommendations, the Medical Database is used to store important information using a lightweight and portable CSV file. This includes doctor information by specialty (e.g. cardiology, pediatrics) with general information on a wide range of drugs including drug names, indications, dosages, and precautions along with details such as eligibility, availability, and contact information. The system predictively queries the data to get relevant and relevant recommendations, enabling users to get actionable guidance.

The Professional Production Module processes end-user feedback and combines predictions and input data into consistent and actionable messages. It states that following a practice such as consulting a doctor or taking medication is clear, accurate, and important.

Integrated AI Coordinator acts as a central controller that helps communicate between products. It handles exceptions, and recovery situations (like uncertain predictions), and optimizes user queries. This module prioritizes feedback to improve overall performance and user satisfaction.

Finally, the output process presents the results in an understandable and usable format. This includes medical diagnoses, recommendations for doctors or medications, and actionable advice. Sanjeevini Bot provides accurate, safe, and user-friendly healthcare services with advanced machine learning, data enhancement, and AI-driven response systems.

The methodology section outlines the plan and method that how the study is conducted. This includes Universe of the study, sample of the study, Data and Sources of Data, study's variables and analytical framework. The details are as follows;

5.1 Prediction of Symptoms Using Random Forest:

Random Forest is a robust machine learning algorithm that excels in classification tasks, particularly for scenarios with high-dimensional data, such as predicting diseases based on symptoms. In the context of the Sanjeevini Bot, this algorithm plays a central role in diagnosing medical conditions by analyzing user-reported symptoms, represented as feature vectors. Below, we delve into the functioning of Random Forest for symptom-based disease prediction, detailing its algorithmic workflow and computational intricacies.

Random Forest is an ensemble learning method that builds multiple decision trees during training and aggregates their predictions to deliver a final result. By combining the outputs of diverse decision trees, Random Forest minimizes the risk of overfitting, increases prediction accuracy, and enhances the model's generalization. Each tree in the forest operates independently, using a random subset of features and training data, which introduces diversity and reduces correlation among trees.

In the Sanjeevini Bot, the input symptoms are preprocessed and converted into a numerical feature vector using techniques like TF-IDF vectorization. This feature vector becomes the input for the Random Forest model, which predicts the most probable disease.

5.2 Algorithm Workflow:

1. Input Feature Vector Generation

The user-reported symptoms are preprocessed, and a numerical representation of the symptoms is created using Term Frequency-Inverse Document Frequency (TF-IDF). For example, an input like "wheezing frequent sneezing clear breathing" might result in a vector [0.748, 0.878, 0.575], where each value corresponds to the weighted importance of a symptom. This feature vector serves as the input to the Random Forest algorithm.

2. Building the Decision Trees

Each decision tree in the Random Forest is constructed using a unique random subset of the training dataset (bootstrapping) and a random subset of features. The steps for building a single tree are as follows:

1. Split the Training Data: A bootstrap sample is drawn, creating a training subset for the tree.
2. Select Random Features: From the total feature set, a smaller random subset of features is chosen to determine the splits at each node.
3. Tree Construction: The tree is grown by splitting the data at each node based on the feature and threshold that best reduces impurity (e.g., using Gini impurity or entropy). This process continues until the tree reaches a maximum depth or contains too few samples to split further.
4. Output at Leaf Nodes: Each leaf node stores the most frequent class label among the samples it contains.

3. Prediction by Individual Trees

Once the decision trees are built, the input feature vector is passed through each tree for prediction. Each tree evaluates the vector by traversing its nodes based on decision rules. For example:

- Root Node Decision: If the "wheezing" weight > 0.7, go left; otherwise, go right.
- Leaf Node Output: The tree outputs a disease prediction, such as "Allergy" or "Asthma."

Each tree generates its prediction along with a confidence score, which reflects the proportion of similar cases in its training subset that led to the same decision.

4. Aggregation of Predictions

The Random Forest combines the predictions of all trees using a majority-voting mechanism. The steps for this aggregation are:

1. Count Votes: Each tree votes for its predicted class. For example:
 - Tree 1: Allergy (confidence: 52%)
 - Tree 2: Allergy (confidence: 48%)
 - Tree 3: Asthma (confidence: 35%)
2. Weighted Aggregation: The confidence scores of each tree are used to calculate the average confidence for each class. For instance:
 - Allergy Confidence: $(52\% + 48\%) / 3 = 47.78\%$
 - Asthma Confidence: $35\% / 3 = 11.67\%$
3. Final Prediction: The class with the highest aggregated confidence score is selected as the final output. In this example, "Allergy" would be the diagnosis with a confidence of 47.78%.

5. Output Interpretation

The model outputs the predicted disease along with associated confidence scores and highlights key symptoms that influenced the decision. This transparency allows users and healthcare providers to understand the reasoning behind the prediction. For example:

- Prediction: "You may have Allergy (47.78% confidence)."
- Key Symptoms: "Wheezing, frequent sneezing, absence of cough".

5.3 Mathematical Foundations of Random Forest:

1. Gini Impurity for Splitting Nodes:

Gini impurity measures the probability of misclassifying a randomly chosen sample from a node. For a node containing k classes, the Gini impurity is:

$$G = 1 - \sum_{i=1}^k p_i^2$$

where p_i is the proportion of samples belonging to class i .

2. Weighted Confidence Scores:

The confidence score for a class is calculated as:

$$C_{\text{Class}} = \frac{\sum_{i=1}^k w_i \delta(t, \text{class})}{T}$$

where w_i is the confidence score of tree t , and $\delta(t, \text{class})$ is 1 if the tree predicts the class and 0 otherwise.

3. Ensemble Accuracy:

The overall accuracy of the Random Forest depends on the diversity and accuracy of individual trees. The ensemble reduces variance and bias, achieving better performance than single classifiers.

5.3.1 Use Case Example: Predicting Allergy:

Let's consider a user reporting symptoms: "I don't have a cough, but I feel wheezing and frequent sneezing." The process unfolds as follows:

1. Feature Vector:

The input symptoms are converted into [0.575, 0.473, 0.748, 0.878].

2. Tree Predictions:

- Tree 1 predicts "Allergy" with 52%.
- Tree 2 predicts "Allergy" with 48%.
- Tree 3 predicts "Asthma" with 35%.

3. Aggregation:

Allergy scores are aggregated to 47.78%, higher than any other class.

4. Output:

"You may have an Allergy with 47.78% confidence. Key symptoms include wheezing and frequent sneezing, with no cough observed."

Random Forest is a powerful algorithm for disease prediction, combining multiple decision trees to deliver accurate, reliable, and interpretable results. Its ability to handle noisy and high-dimensional data, coupled with transparency in feature importance, makes it an ideal choice for symptom-based medical diagnosis. In the Sanjeevini Bot, Random Forest provides actionable insights, bridging the gap between symptom reporting and healthcare assistance.

5.4 Classification and Model Optimization

The Sanjeevini Bot utilizes advanced machine-learning techniques for accurate medical diagnosis and optimization of its predictive model. The system employs **Random Forest classifiers**, which aggregate multiple decision trees to improve reliability. Input preprocessing includes negation detection, symptom standardization, and TF-IDF vectorization to handle user-reported symptoms effectively. Each term's importance is calculated using the TF-IDF mechanism, enabling the model to prioritize rare and significant symptoms like "frequent sneezing" over more common terms such as "clear breathing".

Optimization focuses on enhancing performance through hyperparameter tuning, feature selection, and efficient vectorization. For example, TF-IDF weights help identify the most critical features for prediction, and batch processing ensures resource efficiency. Model accuracy is improved by addressing user input ambiguities, incorporating negation handling, and retraining with updated datasets. Outputs include confidence scores to ensure transparency and reliability in predictions.

This integrated approach ensures that the bot delivers fast, accurate, and user-specific responses, demonstrating the significance of combining robust classification algorithms with continuous optimization in healthcare AI solutions.

5.5 Model Evaluation and Performance Metrics

The Sanjeevini Bot evaluates its machine learning models using key performance metrics to ensure accuracy and reliability. Metrics include **accuracy**, **precision**, **recall**, and the **F1-score**, each offering insights into the model's predictive quality. The Random Forest classifier, central to the system, undergoes rigorous testing through cross-validation and dataset splitting (80% training, 10% validation, 10% testing).

Accuracy measures the correct predictions, while precision assesses the relevance of predicted conditions. Recall evaluates sensitivity by identifying true positives among actual conditions. The F1 score balances precision and recall for comprehensive performance analysis. Confidence scores are also provided to enhance user trust in predictions.

5.6 Deployment and Integration into Healthcare Systems

The Sanjeevini Bot is designed for easy deployment and integration into healthcare environments. It has cross-platform compatibility, supporting various operating systems such as Windows, macOS, and Linux. The system features a modular architecture that includes components like a graphical user interface (GUI), machine learning models, and a structured medical database to provide real-time assistance.

Deployment utilizes virtual environments to manage dependencies and ensure consistency across installations. Lightweight databases based on CSV format simplify integration with existing healthcare systems, while APIs enable interoperability with third-party tools, such as electronic health records (EHRs). Regular updates to datasets and models keep the system relevant and adaptable to changing medical knowledge.

VI. CONCLUSION

The Sanjeevini Bot powerfully demonstrates the transformative capabilities of AI in healthcare, offering personalized, accessible, and efficient medical assistance. By harnessing advanced machine learning techniques, such as the Random Forest classifier and TF-IDF vectorization, the system effectively predicts medical conditions, delivers customized medication recommendations, and connects users with the right healthcare professionals. With its intuitive graphical interface and robust text-to-speech functionality, the bot champions inclusivity, serving a diverse range of users, including those with accessibility challenges. Its modular architecture allows for seamless integration into existing healthcare systems, and continuous updates and optimizations ensure that it remains at the forefront of addressing evolving medical needs. While navigating challenges like data privacy and complex query handling, the Sanjeevini Bot decisively bridges critical gaps in healthcare access, especially in underserved communities. This project underscores the vital role of AI-driven, user-centric solutions in enhancing the efficiency, scalability, and universal accessibility of healthcare services.

REFERENCES

- [1] P. Kandpal, K. Jasnani, ... R. R.-2020 F. W., and undefined 2020, "Contextual chatbot for healthcare purposes (using deep learning)," *ieeexplore.ieee.org* P Kandpal, K Jasnani, R Raut, S Bhorge 2020 Fourth World Conference on Smart Trends in Systems, Security, 2020 • *ieeexplore.ieee.org*, Accessed: Feb. 23, 2024. [Online]. Available <https://ieeexplore.ieee.org/abstract/document/9210351/>
- [2] R. Solanki, A. Rajawat, ... A. G.-H. of R., and undefined 2023, "Building a Conversational Chatbot Using Machine Learning: Towards a More Intelligent Healthcare Application," *igi-global.com* RK Solanki, AS Rajawat, AR Gadekar, ME Patil Handbook of Research on Instructional Technologies in Health Education, 2023 *igi-global.com*, Accessed: Feb. 23, 2024. [Online].
- [3] Sagar Badlani, Tanvi Aditya, Meet Dave, Sheetal Chaudhari, "Multilingual Healthcare Chatbot Using Machine Learning", 2021 2nd International Conference for Emerging Technology (INCET) Belgaum, India. May 21-23, 2021.
- [4] Yamin, F. M., Yusof, N. M., et al. (2019). "A Review of Chatbot in Healthcare: Promises and Challenges." 2019 IEEE Conference on Open Systems (ICOS).
- [5] Joel, J. I., Premalatha, K., et al. (2021). "Design and Implementation of a Health Care Chatbot with Deep Learning Techniques." 2021 International Conference on Communication and Signal Processing (ICCSP)
- [6] S. Ayanouz, B. Abdelhakim, M. B.-P. of the 3rd, and undefined 2020, "A smart chatbot architecture based NLP and machine learning for health care assistance," *dl. ACM.org* S Ayanouz, BA Abdelhakim, M Benhmed Proceedings of the 3rd international conference on networking, information, 2020 *dl.acm.org*, Mar. 2020, doi: 10.1145/3386723.3387897.
- [7] Megarajan G, "Mega Bot - The Healthcare Chatbot", Second International Conference on Electronics and Sustainable Communication Systems, 2021.
- [8] Atulya M. Nayar, Zulfa Attar, Shabbir Kachwala, Tanaya Biswas, Sharmila K. Wagh, "Dost-Mental Health Assistant Chatbot, 5th International Conference on Advances in Science and Technology, 2022.
- [9] Vaibhav Tode, Himanshu Gadge, Sudarshan Madane, Prateek Kachare, Anuradha Deokar, "A Chatbot for Medical Purpose using Deep Learning", International Journal of Engineering Research & Technology, 2021.
- [10] Biju, A., Menon, N.S., Joseph, S.L., Lopez, D. and PJ, S., 2021. Medical Chatbot (Medibot). International Journal, 10(3).