HEALTH CARE MEDIBOT IN MEDICAL FIELD USING MACHINE LEARNING

Thattikota Devi Prasanna[1], Anupoju charan[2], Chintu Raja Sekhar[3], Dunna Uday Kiran[4], Ganga Nikhil Yogeswarao[5]

[1]*Assistant Professor, Computer Science and Engineering, Raghu Engineering College, Visakhapatnam*
[2-5] *B.Tech Students, Computer Science and Engineering, Raghu Institute Of Technology, Visakhapatnam*

---***---

ABSTRACT

Health is considered an important part of human health, including physical, mental and social health, and plays an important role in our lives. Hospitals are the most common way for patients to receive physical examinations ,diagnosis , treatment recommendations. However, it Is very difficult to take every minor health problem to a doctor and get approval from the doctor. The idea is to use artificial intelligence to create a "medical chatbot" that can diagnose disease and provide useful information about the disease before going to the doctor. Chatbots are computers that use natural language to interact with users. The main goal of the project is to create a Medibot that can assist doctors and patients with tasks such as symptom assessment, diagnostic support, treatment recommendations and publication of medical information. Medibot will use machine learning algorithms that learn from big data of medical conversations to understand users' questions and ideas and respond accordingly. medical software or existing platforms and perform rigorous testing to ensure accuracy and reliability. A continuous evaluation and improvement process will be used to improve Medibot's performance over time. Through this project, the team is focusing on the development of AI driven solutions in healthcare while solving real-world problems faced by doctors and patients. Deploying Medibot has the potential to improve clinical processes, improve access to medical information, and ultimately improve patient care outcomes

Key Words:: Medibot, Rasa framework, Health, Artificial intelligence, Natural language processing, Queries, Deployment of ML algorithms, intent, entity

I.NTRODUCTION

The main purpose of creating health-related Chabot's is to provide patients with accurate information for their health-related questions before going to the doctor. "Medical Robotics Using Machine Learning and RASA Framework in Medicine" aims to create intelligent speech specifically for use in the medical field. We aim to create medical robots that can interact with users, understand medical questions, and provide accurate and appropriate answers by using the capabilities of artificial intelligence and machine learning. Leveraging the RASA framework, a well-known open source platform for creating interactive intelligence records, this work aims to create an advanced tool that can improve contact information and support in the healthcare industry.

Scope and Values: Health Domain Expertise: Understand a variety of health topics and questions users may have. This includes medical details, health issues, treatment options and more. 2. Conversational AI development: Use the RASA framework to build a conversational chatbot. This includes creating the conversation flow, processing user input, and generating appropriate responses. 3. Integration of machine learning: Use machine learning algorithms to improve the performance of the chatbot. This can include language understanding (NLU) models that interpret user messages, sentiment analysis to gauge user sentiment, and even personalized recommendations based on user data. 4. Information collection and disclosures: Collect information related to education and improving the quality of education. This can include medical information, health- related questions, or even user-generated chat scripts used to train the chatbot's ability to understand words. 5. User experience (UX) design: Create an intuitive and user friendly interface to interact with the chatbot. This includes considering factors such as ease of use, accessibility, and overall communication. 6. Test and evaluate the performance of the medical chat bot (By checking for bugs or defects) and user testing (for example, collecting feedback from real users to improve the chatbot's results). Communicate effectively and ask some questions and get the answers they want. and check for defects. Because it is a little difficult for patients to get consultation from the doctor for any minor problem. The idea is to create a chatbot that provides patients with some preliminary information and advice before going to the doctor. The motivation for developing a healthcare chatbot using machine learning and the RASA framework stems from the desire to solve important problems in healthcare and support. As the demand for medical services and information continues to increase, there is an urgent need for new solutions that will provide timely assistance and guidance to individuals seeking medical services

2. PROPOSED SYSTEM

The proposed system “Using Medibot in Healthcare Using Machine Learning and RASA Framework” offers a new approach to healthcare delivery through the integration of intelligent chatbots into the existing treatment ecosystem. Powered by machine learning algorithms and built on the RASA framework, Medibot provides personalized medical advice, assistance and support to doctors and patients. The main elements of the proposed system are: Medibot interface: The Medibot interface plays a key role between the user and the system. The interface is designed to be intuitive and easy to use, allowing users to communicate with Medibot via text or speech. This session provides an interactive interface that allows users to ask questions, seek medical advice, and obtain relevant information in real time. Natural Language Understanding (NLU) module: Medibot's NLU module is responsible for understanding user queries and providing relevant information to determine the user's intent and context. Leveraging machine learning technology, the NLU module evaluates text or speech, identifies key tasks and goals, and processes user requests to generate appropriate responses Medibot relies on a comprehensive repository of knowledge and content that includes medical information, recommendations, treatment options, and other resources. This knowledge base is compiled and updated regularly, allowing Medibot to provide users with accurate and up-to-date information to support a wide range of medical questions and conditions. Machine learning models: Trained on large datasets of medical conversations, machine learning models form the backbone of Medibot's intelligence. Built using the RASA framework, this model understands medical terminology, identifies the user's intent, and generates contextually appropriate responses. Continuous training

and improvement of these models over time increases Medibot's accuracy, performance, and adaptability. Patient registration: Patient registration is an important step for users to access the medical chatbot. Users need to provide certain details such as username, email address, location and password. It allows them to receive personalized medical care and interact effectively with the robot. An efficient and user-friendly registration process ensures patients collect accurate information while maintaining security and privacy. User Authentication: Patient authentication is an important part of healthcare chatbot systems to ensure bots interact securely and appropriately with authorized users. Effective patient authentication 7 procedures not only protect patient privacy but also help protect the integrity and confidentiality of sensitive medical information. There are many methods that can be used for patient diagnosis in medical chatbot systems. Users or patients must grant access to Medibot Administration. Answer: Responsive user interface (UI) in medical chatbot projects refers to the design that allows the chatbot interface to adapt seamlessly to various devices and screen sizes, providing users with the best experience across desktops, laptops, and computers, tablets, and mobile devices.

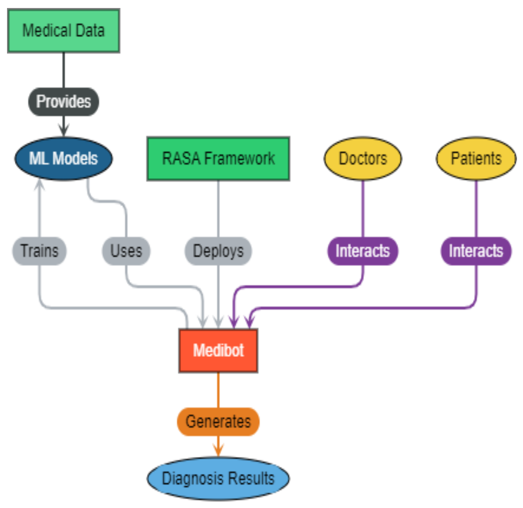


Fig -1: Architecture Design

2.1 ALGORITHMS USED

Natural Language Understanding (NLU):

Algorithm: Natural Language Processing (NLP) techniques combined with machine learning algorithms, such as recurrent neural networks (RNNs) or transformer models like BERT (Bidirectional Encoder Representations from Transformers), are commonly used for NLU tasks.

Explanation: NLU algorithms analyze user queries, tokenize input text, and extract relevant intents and entities. These algorithms utilize pre-trained language models to understand the semantics and context of user input, enabling the system to interpret user queries accurately and determine the user's intent

Dialogue Management (DM):

Algorithm: Rule-based systems, finite-state machines, or reinforcement learning algorithms are often used for dialogue management .Explanation: DM algorithms manage the flow of conversation, maintain context, and select appropriate responses based on the current dialogue state and user input. Rule-based systems use predefined rules and logic to determine system actions, while reinforcement learning algorithms learn optimal dialogue strategies through interaction with users .

Machine Learning Models:

Algorithms: Various supervised and unsupervised machine learning algorithms, including logistic regression, support vector machines (SVM), decision trees, random forests, and deep learning models like convolutional neural networks (CNNs) and long short-term memory (LSTM) networks, are employed for training models in different components of the system .Explanation: Machine learning models are trained on annotated datasets to perform tasks such as intent classification, entity recognition, sentiment analysis, and response generation. These models learn patterns and relationships from input data and make predictions or generate outputs based on learned representations, enabling the system to understand user queries, generate contextually relevant responses, and improve over time through iterative training.

Rule-based Fallback Mechanism:

Algorithm: Rule-based systems or heuristics are used to define fallback strategies when machine learning models fail to provide accurate responses. Explanation: Rule-based fallback mechanisms define rules or thresholds for identifying situations where the confidence of machine learning models is low or where no suitable response is available. In such cases, predefined rules or default responses are triggered to handle user queries and maintain the conversation flow, ensuring robustness and reliability in system interactions.

Diet Classifier:

Implementing a diet classifier within the Rasa framework allows for the integration of conversational AI capabilities, enabling users to receive personalized nutrition guidance through natural language interactions.

Here's a concise guide on how to achieve this: Data Collection and Preparation: Gather a dataset containing information about foods, their nutritional content, and dietary classifications. Preprocess the data, ensuring consistency and encoding categorical variables like dietary classifications .Intent Classification: Define intents in Rasa representing user requests for dietary advice or classification. Train the intent classifier using the dataset to recognize user intents accurately. Entity Recognition: Identify entities such as food items and dietary preferences mentioned by users in their messages. Utilize Rasa's entity recognition capabilities to extract relevant information from user input. Dialogue Management:Design a dialogue flow that guides users through the process of providing dietary preferences and receiving recommendations. Implement custom actions in Rasa to handle user requests, query the diet classifier, and generate responses. Integration with Diet Classifier: Integrate the trained diet classifier model into Rasa as a custom action or external service. Utilize the model to predict the most suitable dietary classification based on user input and dietary preferences. Response Generation
:Generate personalized diet recommendations based on the predicted classification and user preferences. Craft informative and user-friendly responses to provide users with actionable nutrition guidance.

2.2 LIBRARIES USED

NLTK:

NLTK (Natural Language Toolkit) is the foundation of natural language processing (NLP) and speech computing, providing rich functionality and resources for text analysis and word understanding. Its extensive collection includes corpora, linguistic material, and pre-trained models that allow users to easily solve various NLP tasks. NLTK provides a broad set of methods and tools, from simple tokenization and rooting to more advanced tasks such as sentiment analysis, name recognition, and syntax analysis. Moreover, its user-friendly interface and rich information make it accessible to both novice and experienced developers, thus encouraging innovation and discovery in the field. Over the years, NLTK has become the solution of choice for academics, businesses, and hobbyists, pioneering advances in data transmission for chatbots and in areas such as machine translation and sociological analysis. With its continued development and community support, NLTK continues to be a valuable resource for anyone entering the fascinating world of natural language understanding and processing. NLTK's versatility extends beyond traditional NLP tasks, offering support for linguistic research, educational purposes, and industry applications. Its modular design allows for easy integration with other Python libraries and frameworks, amplifying its utility in various projects. With ongoing updates and a vibrant community, NLTK continues to shape the landscape of NLP, empowering users worldwide to unlock the complexities of human language through computational analysis and interpretation**.**

Flask:

Flask is a lightweight and flexible Python web framework that allows developers to create fast and efficient web applications. With its simplicity and simplicity, Flask provides the flexibility to create applications of any size, from simple models to complex systems. The design structure encourages the use of extensions to increase efficiency, allowing developers to tailor applications to their specific needs. The design structure encourages the use of extensions to increase efficiency allowing developers to tailor applications to their specific needs. Flask's built-in developer server and debugger simplify the development.. Flask's built-in developer server and debugger simplify the development process, while its rich documentation and community provide broad support for developers of all levels. Additionally, Flask's Jinja2 template engine helps create dynamic and interactive pages to enhance user experience. Flask's support for RESTful APIs allows integration with front-end systems and third-party services, making it ideal for modern web development. Strong security features such as CSRF protection and secure cookie management keep applications and user data safe. Thanks to its scalability and versatility, Flask remains a popular choice for developers to build web applications that deliver better performance and user satisfaction.

RASA:

Rasa is an open-source framework for building conversational AI applications. It provides tools and libraries for developing chatbots and virtual assistants capable of understanding natural language input and responding appropriately key features include in RASA: Natural Language Understanding (NLU): Rasa NLU allows you to train models to extract intents and entities from user messages, enabling your chatbot to understand what the user is asking or saying. Dialogue Management: With Rasa Core, you can create conversational flows and manage dialogue states to provide contextually relevant responses to users. Rasa Core uses machine learning algorithms to predict the next best action based on the conversation history. Customization and Flexibility: Rasa offers flexibility for customizing and extending its components to suit your specific use case. You can integrate custom machine learning models, define complex conversational logic, and integrate with external APIs and services. Open- Source Community: Rasa has a vibrant open-source community that contributes to its development, shares best practices, and provides support through forums and

resources. Scalability and Deployment: Rasa is designed to be scalable and can be deployed in various environments, including on-premises servers, cloud platforms, and containerized environments. This makes it suitable for both small-scale prototypes and large-scale production deployments

PyTorch:

The Python function library Torch is an evolution of the original Lau-based Torch framework and is a versatile tool for deep learning. Torch leverages the power of high-performance GPU acceleration to speed up the training process of complex neural network models. The built-in auto-differentiation functionality makes it easy to apply custom loss and optimization techniques. Torch has a broad ecosystem of pre-existing learning models, various databases, and critical tools to enable efficiency and innovation in the development process. Torch integrates seamlessly with PyTorch to support collaborative and social code. Torch's adaptability and high-performance capabilities are widely recognized by researchers and practitioners, ensuring its position as the foundation of deep learning libraries.

2.3 TECHNOLOGIES USED

Python:

Python as the primary programming language for developing the Medibot system components, leveraging its extensive libraries and frameworks for natural language processing, machine learning, web development, and system integration and ideally hands on coding experience on the Python programming language plays a important role in developing the medical chatbot.

Development Frameworks and Libraries:

RASA Framework for building and deploying conversational AI applications, providing tools and libraries for natural language understanding, dialogue management, and machine learning model training and inference. PyTorch for implementing machine learning algorithms and models, enabling tasks such as natural language processing, sentiment analysis, and intent classification within the Medibot system. Flask or Django as web application frameworks for backend development, facilitating the creation of RESTful APIs, request handling, data processing, and business logic implementation

Database Management System (DBMS):

MySQL, PostgreSQL database management system to store and manage user data, system configurations, interaction logs, and patient feedback and other persistent data, ensuring data integrity, reliability, and scalability and also plays a major part for user authentication that enables only authorized users can handle the bot. Patient Data Management: A DBMS would be used to store and manage patient data securely. This includes information such as patient demographics, medical history, medications, allergies, and treatment plans. The DBMS ensures that patient data is organized, accessible, and remains confidential.

HTML CSS AND Bootstrap:

The user interface (UI) of a bot, implemented with HTML, CSS, and potentially enhanced with frameworks like Bootstrap, plays a crucial role in the overall user experience. Here's how: Presentation of the Bot: HTML and CSS are used to structure and style the visual elements of the bot's interface. This includes designing the layout, formatting text, styling buttons, and creating any visual elements such as avatars or icons. A well-designed UI can make the bot more appealing and engaging to users. User Interaction: HTML provides the structure for interactive elements like input fields and buttons, while CSS defines their appearance and behavior.. This ensures that the bot's interface is accessible and usable across desktops, tablets, and mobile devices, providing a consistent experience for users regardless of the device they're using.

2.4 RESULTS

The below images depicts different Machine learning models to simulate the results for the "Health care medibot in medical field project. Which includes different cases like disease predictions based on the symptoms given and food recommendations and remedies that offers a vast knowledge to the users.

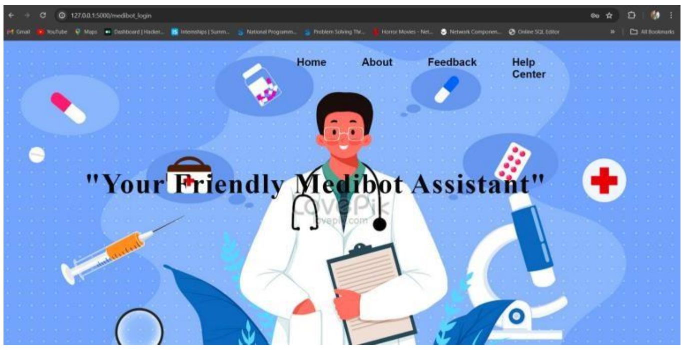


Fig -2: Home page

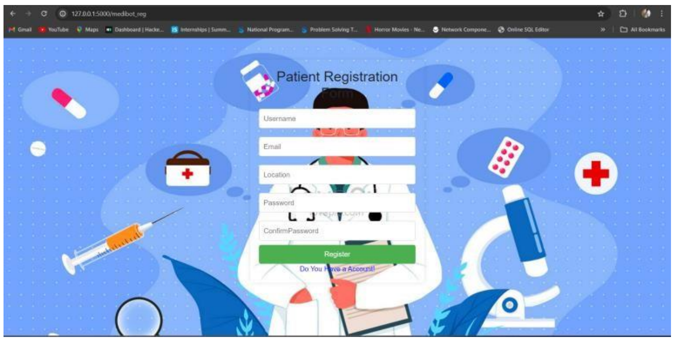


Fig -3: Patient Registration

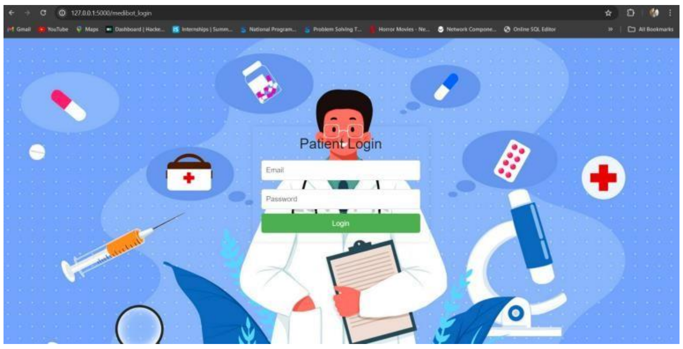
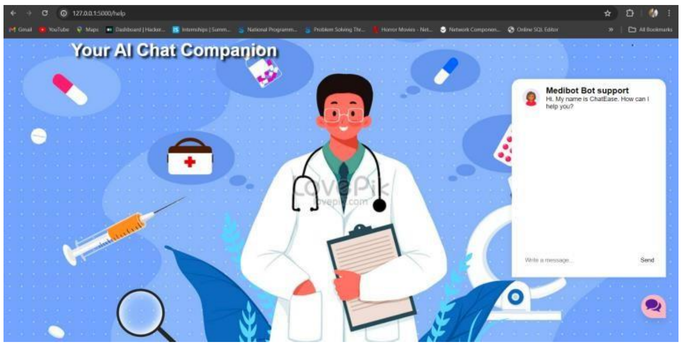
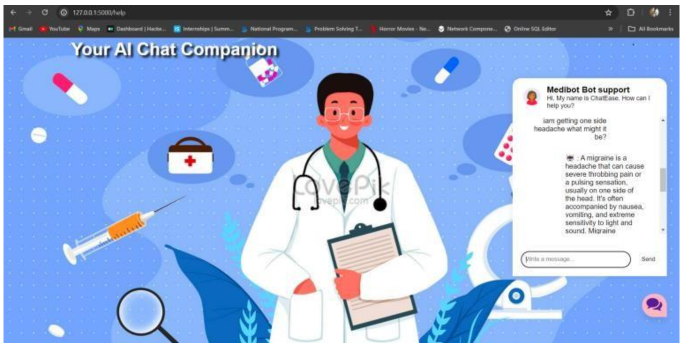


Fig -4: Patient login



Hi. My name is ChatEase. How can I help you?

Fig-5: Medibot interface

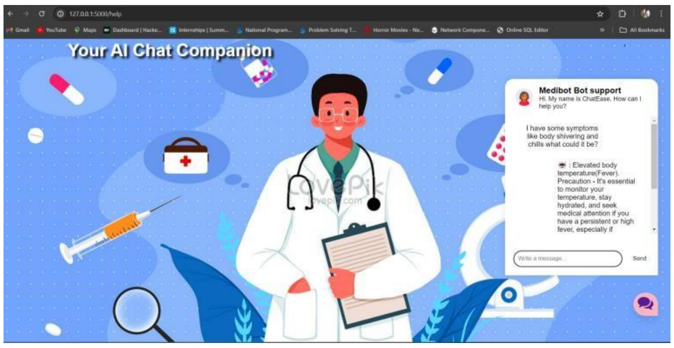


Hi. My name is ChatEase. How can I help you?

iam getting one side headache what might it be?

: A migraine is a headache that can cause severe throbbing pain or a pulsing sensation, usually on one side of the head. It's often accompanied by nausea, vomiting, and extreme sensitivity to light and sound. Migraine

Fig-6 :Different Responses by the Medibot

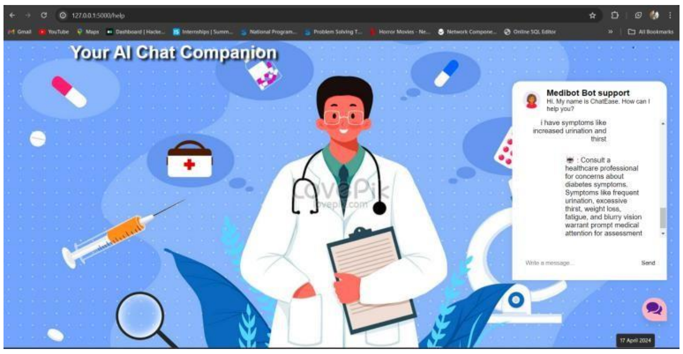


Hi. My name is ChatEase. How can I help you?

I have some symptoms like body shivering and chills what could it be?

: Elevated body temperature(Fever). Precaution - It's essential to monitor your temperature, stay hydrated, and seek medical attention if you have a persistent or high fever, especially if

Fig-7: Disease prediction based on symptoms

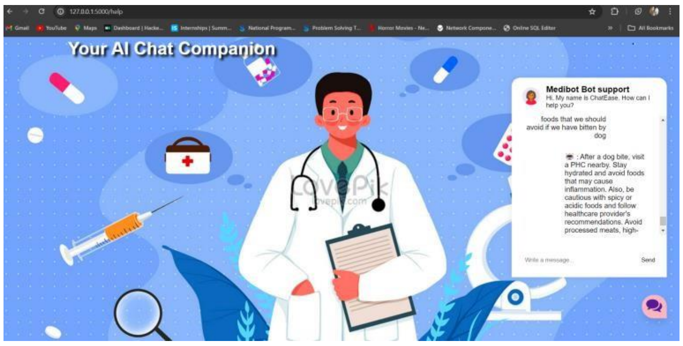


Hi. My name is ChatEase. How can I help you?

i have symptoms like increased urination and thirst

: Consult a healthcare professional for concerns about diabetes symptoms. Symptoms like frequent urination, excessive thirst, weight loss, fatigue, and blurry vision warrant prompt medical attention for assessment

Fig-8: Disease prediction based on symptoms



Hi. My name is ChatEase. How can I help you?

foods that we should avoid if we have bitten by dog

: After a dog bite, visit a PHC nearby. Stay hydrated and avoid foods that may cause inflammation. Also, be cautious with spicy or acidic foods and follow healthcare provider's recommendations. Avoid processed meats, high-

Fig-9: Food precautions for particular disease

3. CONCLUSIONS

Overall, Project Medibot represents a promising start in medical technology that has the potential to revolutionize the way patients, caregivers, and doctors access, present, and use medical records. Through continuous innovation and development, Medibot systems can play an important role in improving health, promoting preventive care and ultimately improving the health of people around the world. It paves the way for innovations in medical technology. Some future possibilities for the program include: Additionally, the Medibot program has the potential to improve access to healthcare, improve patient experience, collaborate, and support physicians in clinical decision-making. The system provides 24/7 medical information, personalized guidance and telemedicine services, allowing users to make decisions about their health and well-being, resulting in better health benefits and better Quality of Life. In conclusion, the Medibot project represents a promising initiative in the realm of healthcare technology, with the potential to transform the way medical information is accessed, delivered, and utilized by patients, caregivers, and healthcare professionals. By continuing to innovate and evolve, the Medibot system can play a pivotal role in advancing healthcare delivery, promoting preventive care, and ultimately improving the health and well-being of individuals worldwide.

ACKNOWLEDGEMENT

We are grateful to the Department of Computer Science and Engineering, Raghu Educational Institutions, Visakhapatnam for helping us in our work and supporting us always.

REFERENCES

1. Bates, D. W., & Gawande, A. A. (2003). Improving safety with information technology. New England Journal of Medicine, 348(25), 2526-2534.
2. Boissel, J. P., Auffray, C., & Noble, D. (Eds.). (2016). Systems medicine: Integrative, qualitative and computational approaches. Academic Press.
3. Chui, K. K. H., Wenger, D., Qiu, R., & Cheng, Y. (2018). Artificial intelligence and robotics. In Handbook of Medical Image Computing and Computer Assisted Intervention (pp. 925-942). Springer.
4. Davenport, T. H., & Kalakota, R. (2019). The potential for artificial intelligence in healthcare. Future Healthcare Journal, 6(2), 94-98.
5. Dilsizian, S. E., Siegel, E. L., & Greenes, R. A. (2018). Clinical decision support systems for improving diagnostic accuracy and achieving precision medicine. Journal of Nuclear Medicine, 59(3), 350-353
6. Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115-118.
7. Ficarra, B. J., Desai, A., & Narayanan, R. (Eds.). (2016). The role of artificial intelligence in advancing diabetes management. Springer International Publishing
8. Patel, V. L., Arocha, J. F., & Kaufman, D. R. (2002). Diagnostic reasoning and medical expertise. In International Handbook of Research in Medical Education (pp. 695-726). Springer
9. Roberts, K. J. (2012). Patient-generated health data, electronic health records, and clinical decision support: Survey of health care providers. JMIR Medical Informatics, 1(1), e5.
10. Shortliffe, E. H., & Cimino, J. J. (Eds.). (2014). Biomedical informatics: Computer applications in health care and biomedicine. Springer.
11. Topol, E. J. (2019). Deep medicine: How artificial intelligence can make healthcare human again. Basic Books