Project Progress Report on

Artificial Narrow Intelligence based Healthcare Service Recommendation System

Submitted in partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by:

Samarth Tiwari	**2011099**
Prasoon Thapliyal	**2018559**

Under the Guidance of

Mr. Sushant Chamoli

Assistant Professor

Department of Computer Science and Engineering
Graphic Era Hill University
Dehradun, Uttarakhand

CANDIDATE'S DECLARATION

I/We hereby certify that the work is being presented in the Project Report entitled **"Artificial Narrow Intelligence based Healthcare Service Recommendation System"** in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering and submitted to the Department of Computer Science and Engineering of the Graphic Era (Deemed to be University), Dehradun is an authentic record of my work carried out during a period from **August-2023 to May-2024** under the supervision of **Mr. Sushant Chamoli, Assistant Professor**, Department of Computer Science and Engineering, Graphic Era Hill University

The matter presented in this dissertation has not been submitted by me/us for the award of any other degree of this or any other Institute/University.

Samarth Tiwari	2011099
Prasoon Thapliyal	2018559

The above-mentioned students shall be working under the supervision of the undersigned on the **"Artificial Narrow Intelligence Based Healthcare Service Recommendation System."**

29/05/24

Supervisor **Head of the Department**

Examination

Name of the Examiners: **Signature with Date**

1.

2.

Abstract

The healthcare sector confronts various challenges, including the imperative to manage extensive datasets, develop innovative healthcare applications, and optimize overall delivery efficiency. This research project proposes a unique system designed to enable users to input symptoms, receive rapid diagnosis, and access recommendations for medication and nearby hospitals and doctors, addressing the time-sensitive nature of medical diagnoses. The envisioned system capitalizes on the combined strengths of Artificial Narrow Intelligence (ANI) and cloud computing to provide swift and accurate diagnoses. An extensive literature survey has been conducted to understand the existing challenges in current approaches to healthcare delivery. ANI facilitates precise interpretation of symptoms, enhancing user experience, while cloud computing ensures seamless accessibility, allowing users to utilize the system at their convenience. Moreover, this integration aims to alleviate the workload on medical professionals, ultimately enhancing their workflow and efficiency. The disease prediction model utilizes the random forest model, boasting an accuracy of 97%, while the drug prediction model, which utilizes the Sequential model, boasts an accuracy of 93%. Recommendation system with an accuracy of 98%, along with the NLP is utilized for recommending hospitals and doctors. The proposed system not only tackles the issue of delayed diagnoses but also offers a user-friendly interface, empowering individuals to proactively manage their health. The collaborative use of these technologies promises to reshape healthcare delivery, providing an efficient and accessible solution tailored to the demanding nature of the medical field.

Keywords—healthcare, chatbot, artificial narrow intelligence (ANI), cloud computing, natural language processing (NLP), machine learning (ML), recommendation system, long short term memory (LSTM), cosine similarity, support vector classification (SVC)

ACKNOWLEDGEMENT

We would like to express our deepest gratitude to everyone who has contributed to the successful completion of this project. First and foremost, I am thankful to our supervisor, Mr. Sushant Chamoli Sir, for their continuous support, guidance, and invaluable insights throughout the research process. Their expertise and encouragement have been instrumental in shaping this work.

We thank the **management of Graphic Era Hill University** for the support throughout the course of our Bachelor's Degree and for all the facilities they have provided.

Last, but certainly not least we thank all teaching and non-teaching staff of Graphic Era Hill University for guiding us on the right path. Most importantly we wish to thank our parents for their support and encouragement.

Samarth Tiwari	**2011099**
Prasoon Thapliyal	**2018559**

Table of Contents

Contents	Page No.
Abstract	i
Acknowledgment	ii
Table of Contents	iii
List of Figures	iv
Chapter 1 Introduction	**1-3**
1.1 Project Introduction	1
1.2 Problem Statement	2
1.3 Objectives	2-3
Chapter 2 Literature Survey/ Background	**4-5**
Chapter 3 Software Design	**6-7**
Chapter 4 Requirements and Methodology	**8-11**
4.1 Requirements	
4.1.1 Hardware Requirements	
4.1.2 Software Requirements	
4.2 Libraries and Components Used	
4.3 Methodology	
Chapter 5 Coding /Code Templates	**13-14**
Chapter 6 Testing	**15**
Chapter 7 Result and Discussion	**16**
Chapter 8 Conclusion and Future Work	**17**
Details of Research Publication	**18**
References	**19-20**

List of Tables

Table No.	TITLE	PAGE No.
4.1	Hardware Requirements	8
4.2	Software Requirements	8
6.1	Unit Testing	15
6.2	Integrating Testing	15
6.3	System Testing	16

List of Figures

FIGURE No.	TITLE	PAGE No.
3.1	Connectivity Diagram	6
3.2	Use Case Diagram for NLP	7
3.3	Implementation of Disease Prediction	7
4.1	ChatBot Interface	11
4.2	Confusion Matrix for SVC (Disease Prediction)	11
5.1	Recommend Drug Function	13
5.2	Recommend Hospital Function	13
5.3	Recommend Doctor Function	14
5.4	Disease Prediction Function	14
6.1	Accuracy and Loss Function for NLP Model	15

Chapter 1

Introduction and Problem Statement

1.1 Introduction

In today's rapidly evolving healthcare landscape, the discovery of new technologies and convergence of old technologies has paved the way for groundbreaking innovations with the potential to enhance patient care and make medical operation efficient. ANI is a specific type of artificial intelligence in which a learning algorithm is designed to perform a single task, and any knowledge gained from performing that task will not automatically be applied to other tasks. It leverages AI technologies to establish a highly efficient system that mimics and potentially exceeds human intelligence. ANI facilitates faster decision-making since they can process and complete tasks significantly quicker than humans. As a result, it boosts productivity & efficiency of the task at hand. Cloud computing is on-demand access, via the internet, to computing resources—applications, servers (physical servers and virtual servers), data storage, development tools, networking capabilities, and more—hosted at a remote data center managed by a cloud services provider. Cloud computing has greatly changed how healthcare works. It's made it easier to get and share medical information, adjust resources as needed, save money, keep data safe, and work together. It helps doctors use data analysis, AI, and telehealth to give better care, do more research, and make healthcare more efficient overall. In simple terms, cloud computing is like a powerful tool that's making healthcare better and smarter. The fusion of cloud computing and artificial narrow intelligence (ANI) is one such groundbreaking fusion. In today's world, where data is everywhere and information keeps coming in non-stop, combining cloud computing and ANI is a powerful combination that's ready to completely change how healthcare works in many ways. One of its main advantages is that it helps healthcare providers use their resources like doctors, equipment, and time more effectively, making sure patients get the right treatment when they need it. Additionally, it can improve how well treatments work by using data to give healthcare professionals better information for making decisions about patient care. During our exploration, we will see how technology can drastically change healthcare for the better, making it more personalized and efficient. This will lead to a brighter future for medicine. We hope to create something which makes the irksome task of going to the hospital and waiting hours in line simpler and less time-consuming.

The healthcare sector confronts various challenges such as the requirement to gather and manage extensive data, the necessity to create and implement novel healthcare applications, and the imperative to enhance the efficiency and quality of healthcare services. The field of medicine is intricate, leading to delays in diagnosis, which can be crucial for certain patients. In situations where visiting a hospital is impractical, virtual diagnosis may prove challenging and less accurate, particularly in critical conditions. Addressing this issue is of utmost importance.

The current healthcare industry faces a number of challenges, including collection and storage of large amounts of data, and the suboptimal efficiency and effectiveness of present healthcare delivery systems. The practice of medicine demands precision and care, which requires doctors to often invest considerable time in diagnosing patients. These assessments are essential for accurately identifying and treating complex medical conditions. However, it's important to note that not all medical issues require such extensive diagnosis; for minor ailments and routine health concerns, simpler and quicker assessments are sufficient. In response to these pressing issues, we propose an integration of Natural Language Processing (NLP), Machine Learning (ML) and Cloud Computing technologies to create a narrow Artificial Intelligence (AI) chatbot, allowing users to seamlessly enter their symptoms and receive comprehensive diagnoses, coupled with tailored medication recommendations. Additionally that chatbot will also provide the users with information regarding the nearest healthcare service providers, including doctors and hospitals, thereby facilitating swift and informed healthcare decisions.

1.2 Problem Statement

The healthcare industry is facing a number of challenges, including the need to collect and store large amounts of data, the need to develop and deploy new healthcare applications, and the need to improve the efficiency and effectiveness of healthcare delivery. Medicine is a very delicate profession; hence it takes time for the doctors to diagnose people which can create delays. This lost time can be very critical for some patients.

Sometimes going to the hospital is not an option and virtual diagnosis is hard and likely to be not accurate when the patient is in a serious condition. This is a huge concern which needs to be dealt with

We are creating a system which lets users enter their symptoms and gives them the diagnosis while providing them with the option for medication, recommended hospital in their area etc.

We will make this system with the help of artificial narrow intelligence (ANI) and cloud computing. The fusion of both these technologies will make the system efficient and quick, as well as easily accessible. It will also make the highly hectic life of medical workers easier and even improve their efficiency.

1.3 Objectives

The Healthcare Service Recommendation System is a sophisticated cloud-based solution that leverages Artificial Narrow Intelligence (ANI) to offer patients highly personalized healthcare recommendations. The system's core objectives encompass a wide range of healthcare-related functions, designed to enhance patient care, streamline healthcare decision-making, and improve overall healthcare experiences.

Main Objectives:

1) Disease Prediction:

The primary goal of the system is to predict potential health issues using advanced AI algorithms. These algorithms analyze patient data, including medical history, demographics, and lifestyle factors, to identify early signs of diseases or health risks.

Disease prediction serves as a proactive approach to healthcare, allowing patients and healthcare providers to take preventive measures and initiate timely interventions.

2) Hospital Recommendation:

Another critical objective is to suggest suitable healthcare facilities to patients based on their geographical location, personal preferences, and medical diagnosis.

The system ensures that patients are directed to hospitals and medical centers that specialize in their healthcare needs, optimizing treatment outcomes.

3) Medicine Recommendation:

Providing personalized medication recommendations is a key function of the system. It takes into account factors such as patient allergies, medical history, current conditions, and potential drug interactions.

Tailored medication recommendations not only improve patient safety but also enhance medication adherence, leading to more effective treatments.

4) NLP-guided User Interaction:

To enhance user experience, the system implements Natural Language Processing (NLP) techniques. These techniques enable the system to understand and respond effectively to natural language queries from patients. Patients can engage with the system using voice commands or text queries, making it user-friendly and accessible.

5) Patient-Centric Design:

The system has been meticulously designed with a patient-centric approach, prioritizing accessibility and ease of use. Patient interfaces, including mobile applications and web portals, are intuitive and user-friendly, ensuring that patients can easily access and utilize the system's recommendations.

Improving Healthcare Quality and Patient Experience:

Ultimately, the system aims to enhance the quality of healthcare services and elevate the overall patient experience. Timely recommendations facilitate faster diagnoses, treatments, and interventions, potentially reducing healthcare costs and improving health outcomes.

6)Doctor Recommendation:

Ethical considerations, such as transparency, fairness, and patient consent, are integral to the system's operations. Robust data privacy measures and compliance with healthcare regulations are in place to safeguard patient information and build trust.

7) Cloud deployment:

All the Components of the system that have been implemented have to be integrated and deployed in a cloud platform so that the users can easily utilize the system from any part of the world.

The Healthcare Service Recommendation System, with its multifaceted objectives and patient-focused approach, represents a significant step towards revolutionizing healthcare services and ensuring that patients receive the best possible care tailored to their unique needs and circumstances.

Chapter 2

Literature Survey

Cloud computing and artificial narrow intelligence-based healthcare service recommendation systems is a novel approach to provide personalized and optimal health care solutions to patients. The system leverages the power of cloud computing to store and process large amounts of health data, and the capabilities of artificial narrow intelligence to analyse and learn from the data. The system can recommend the best health care services for each patient based on their medical history, symptoms, preferences, and budget. The system can also monitor the patient's health status and provide feedback and alerts if needed. The system aims to improve the quality and efficiency of health care delivery, reduce costs, and enhance patient satisfaction.

S. Baker and W. Xiang, [1] in their paper provides a comprehensive overview of the Artificial Intelligence of Things (AIoT) for healthcare applications, which is a novel paradigm that integrates artificial intelligence (AI) and the Internet of Things (IoT) to enable smart, connected, and intelligent healthcare systems. The paper reviews the main concepts, architectures, applications, and challenges of AIoT for healthcare, as well as the state-of-the art techniques and technologies for implementing AIoT solutions. The paper also discusses the future research directions and opportunities for AIoT for healthcare, with a focus on addressing the key challenges of data quality, security, privacy, interoperability, scalability, and ethics.

Mu-Hsing Kuo [2] in his article explained Opportunities and Challenges of Cloud Computing to Improve Health Care Services. Cloud computing is a technology that allows users to access and process data over the internet, without requiring local storage or computing resources. Cloud computing has many potential benefits for health care services, such as improving efficiency, scalability, security, and interoperability. However, cloud computing also poses some challenges, such as privacy, legal, ethical, and technical issues. This article by Mu-Hsing Kuo provides an overview of the opportunities and challenges of cloud computing in health care, and discusses some possible solutions and future directions.

Chen Gao [3] in the paper provides a comprehensive overview of the recent advances in graph neural networks (GNNs) for recommender systems, which are systems that suggest items or services to users based on their preferences and behavior. The paper discusses the challenges and opportunities of applying GNNs to recommender systems, such as modeling complex useritem interactions, incorporating side information, and handling dynamic and heterogeneous graphs. The paper also reviews the existing methods of GNN-based recommender systems, which can be categorized into three types: embedding-based, generation-based, and reasoning based methods. The paper further identifies the future research directions of GNN-based recommender systems, such as developing more efficient and scalable algorithms, designing more expressive and interpretable models, and exploring more realistic and challenging scenarios.

The book Artificial Intelligence for Smart Healthcare [4] is a collection of chapters that explore various applications of AI techniques in the domain of health and medicine. The editors have compiled the work of researchers from different countries and backgrounds, who present their insights and experiences on topics such as natural language processing, computer vision, machine learning, deep learning, data mining, and optimization. The book covers a wide range of problems and solutions, such as diagnosis, prediction, analysis, decision support, recommendation, and personalization. The book aims to provide a comprehensive overview of the current state-of-the-art and future directions of AI for smart healthcare.

Romany Fouad Mansour [5] proposes a novel disease diagnosis model that leverages artificial intelligence and internet of things technologies to provide smart healthcare services. The model consists of three main components: a wearable device that collects vital signs and symptoms from patients, a cloud server that analyses the data using deep learning algorithms, and a mobile application that displays the diagnosis results and recommendations. The model is evaluated on four common diseases: diabetes, hypertension, heart failure, and asthma. The results show that the model achieves high accuracy, precision, recall, and F1-score for each disease, outperforming existing methods. The model also provides personalized and proactive healthcare solutions, such as medication reminders, lifestyle suggestions, and emergency alerts. The paper demonstrates the potential of artificial intelligence and the internet of things to improve the quality and efficiency of healthcare systems.

Chapter 3

Software Design

The healthcare recommendation system is designed to provide comprehensive assistance to users by predicting diseases and recommending appropriate drugs, hospitals, and doctors based on user input. The system architecture and its components are illustrated through three detailed diagrams.

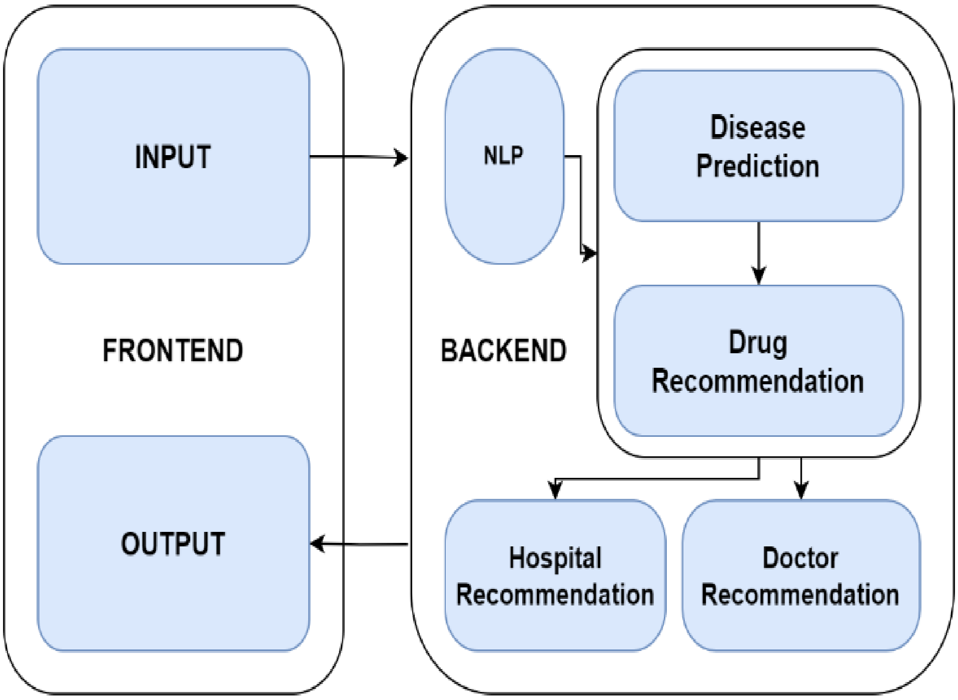

Fig. 3.1 Connectivity Diagram

The first diagram presents a high-level architectural overview of the system, divided into frontend and backend components. The frontend consists of two primary modules: the Input and Output interfaces. The Input module captures user data, which is then processed by the backend. The Output module displays the recommendations and predictions generated by the backend.

The backend is responsible for the core functionalities of the system. It includes an NLP (Natural Language Processing) module, which interprets and processes the user input to extract meaningful information. This processed information is then utilized by the Disease Prediction module, which employs machine learning techniques to predict potential diseases based on the input data. Subsequently, the Drug Recommendation module provides suggestions for medications relevant to the predicted diseases. The backend also encompasses Hospital Recommendation and Doctor Recommendation modules, which recommend suitable healthcare providers and professionals, ensuring a comprehensive support system for the user.

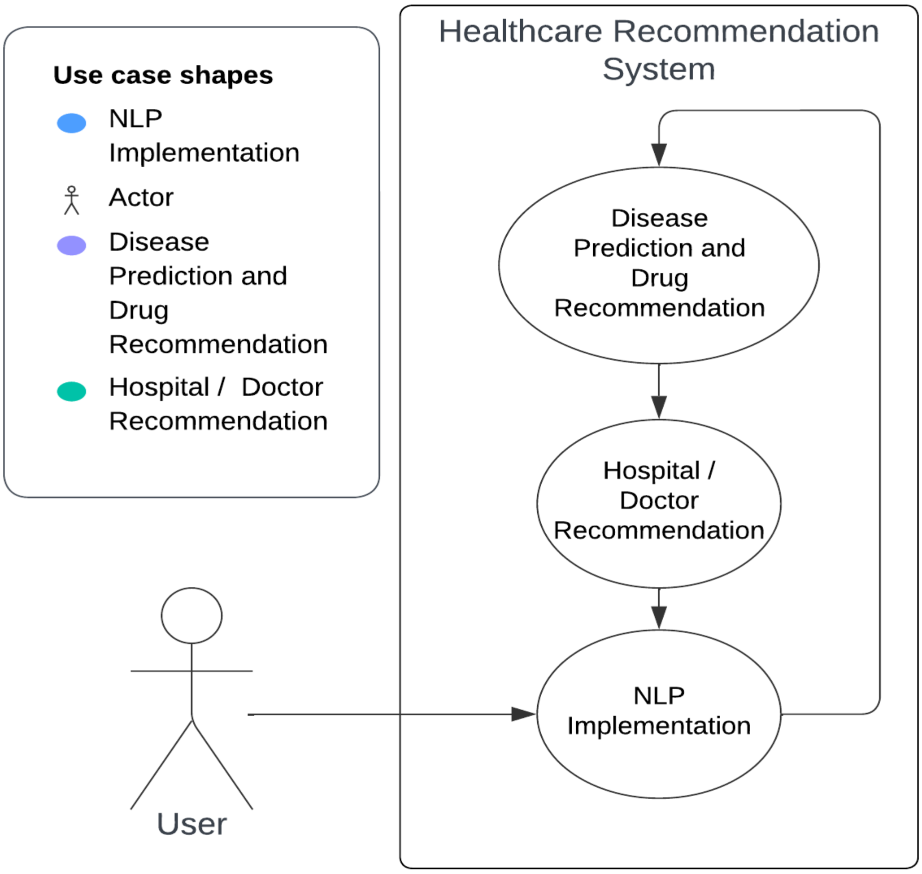

Fig. 3.2 Use Case Diagram for NLP

The second diagram, a use case diagram, elaborates on the interaction between the user and the system. The User interacts with the system by providing input, which is processed by the NLP Implementation. This implementation is crucial as it bridges the user's natural language input with the system's analytical capabilities. The processed data then flows through the Disease Prediction and Drug Recommendation module, followed by the Hospital/Doctor Recommendation module. This sequential flow ensures that the user receives a holistic set of recommendations, starting from disease prediction to finding the appropriate healthcare services.

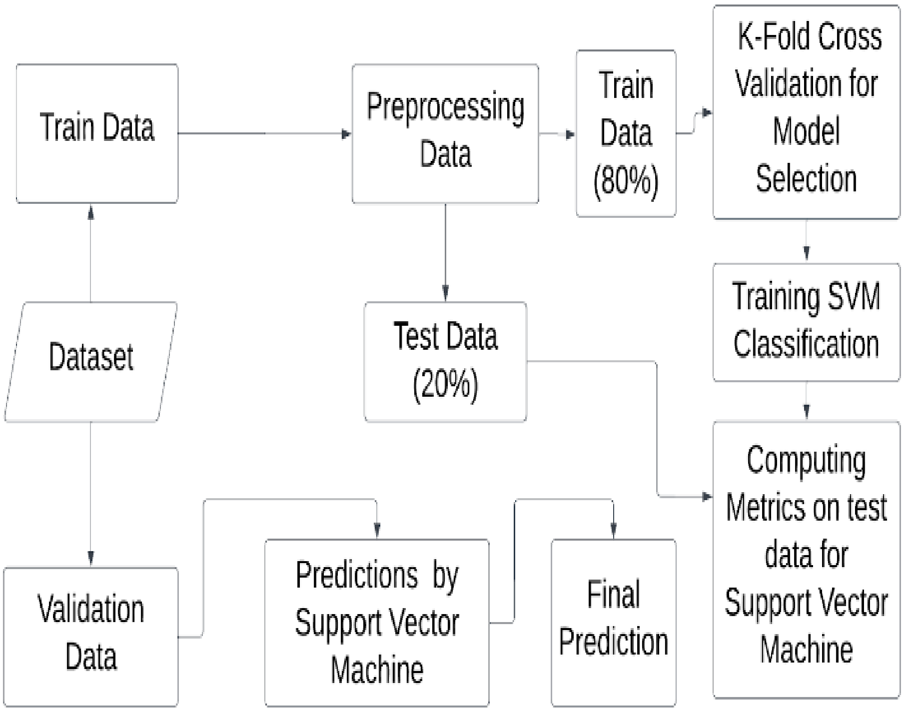

Fig. 3.3 Implementation of Disease Prediction

The third diagram provides a detailed flowchart of the Disease Prediction implementation process. This process begins with the Dataset acquisition, which is split into Train Data and Validation Data. The Train Data undergoes Preprocessing to clean and prepare it for the machine learning model. The data is then divided into Train Data (80%) and Test Data (20%) to facilitate model training and evaluation. The system employs K-Fold Cross Validation to ensure robust model selection, followed by training a Support Vector Machine (SVM) classifier. The model's performance is evaluated using metrics computed on the test data, leading to the Final Prediction. This rigorous process ensures that the disease prediction module is accurate and reliable.

Chapter 4

Requirements and Methodology

4.1 Requirements:

4.1.1 Hardware Requirements

Table. 4.1.1 Hardware Requirements

HARDWARE	MINIMUM REQUIREMENTS
System	A system with a minimum of 16GB RAM and a high end GPU will be required for development and testing
Network Connection	A stable and fast network connection will be required for the development, testing, and deployment of the application
Storage	An extensive storage will be required to store datasets and video taken for training and testing

4.1.2 Software Requirements

Table. 4.1.2 Software Requirements

SOFTWARE	MINIMUM REQUIREMENTS
Development Environment	An Integrated Development Environment (IDE), we are using Google Colab to develop the software
Data Sources	Annotated data sources of vehicle images, helmet images are required. Video frames are taken and annotated manually.
Cloud Infrastructure	Google Colaboratory is used for development and training of the models.

4.2 Libraries and Components Used:

4.2.1 Flask==3.0.3

- **4.2.1.1 Description**: Flask is a lightweight and flexible web framework for Python. It is designed to make getting started quick and easy, with the ability to scale up to complex applications.
- **4.2.1.2 Key Features**:

- **Simplicity**: Flask has a simple and easy-to-understand API, making it a good choice for beginners.
- **Flexibility**: It provides the flexibility to use different components and libraries as needed.
- **Extensibility**: There are numerous extensions available to add more functionality to Flask applications.
- **Jinja2 Templating**: Integrated support for Jinja2 templating to render dynamic web pages.

4.2.2 flask-cors==4.0.0

- **4.2.2.1 Description**: Flask-CORS is an extension for handling Cross-Origin Resource Sharing (CORS) in Flask applications. CORS is a mechanism that allows web applications to request resources from different domains.
- **4.2.2.2 Key Features**:
    - **Easy Integration**: Simple to integrate with Flask applications to enable CORS.
    - **Configurable**: Allows configuration of CORS policies to specify which domains are permitted.
    - **Security**: Helps in securely managing cross-origin requests.

4.2.3 nltk==3.6.5

- **4.2.3.1 Description**: The Natural Language Toolkit (NLTK) is a leading platform for building Python programs to work with human language data.
- **4.2.3.2 Key Features**:
    - **Comprehensive Library**: Provides tools for various tasks such as classification, tokenization, stemming, tagging, parsing, and more.
    - **Educational Resources**: Extensive documentation and resources for learning and teaching natural language processing (NLP).
    - **Corpora**: Includes a large collection of text corpora for experimentation and training.

4.2.4 numpy==1.26.4

- **4.2.4.1 Description**: NumPy is the fundamental package for numerical computing in Python. It provides support for arrays, matrices, and many mathematical functions.
- **4.2.4.2 Key Features**:
    - **Efficient Array Operations**: Fast operations on arrays and matrices of arbitrary size.
    - **Mathematical Functions**: Extensive library of mathematical functions to operate on these arrays.
    - **Integration**: Can be integrated with a wide variety of databases and other scientific computing libraries.

4.2.5 tensorflow==2.16.1

- **4.2.5.1 Description**: TensorFlow is an open-source platform for machine learning and artificial intelligence. It is widely used for developing and training neural networks.
- **4.2.5.2 Key Features**:
    - **Deep Learning**: Extensive support for deep learning models and architectures.
    - **TensorFlow Hub**: Access to a repository of reusable machine learning models.
    - **Scalability**: Designed to scale from simple prototypes to large-scale machine learning applications.

4.2.6 scikit-learn==1.4.2

- **4.2.6.1 Description**: Scikit-learn is a robust library for machine learning in Python. It provides simple and efficient tools for data mining and data analysis.
- **4.2.6.2 Key Features**:
    - **Algorithms**: Wide range of supervised and unsupervised learning algorithms.
    - **Model Evaluation**: Tools for model selection, validation, and evaluation.
    - **Integration**: Works seamlessly with other scientific libraries like NumPy and Pandas.

4.2.7 pandas==2.2.2

- **4.2.7.1 Description**: Pandas is a powerful data analysis and manipulation library for Python. It provides data structures like DataFrame, which are ideal for handling and analyzing structured data.
- **4.2.7.2 Key Features**:
    - **DataFrames**: Easy handling of tabular data with DataFrame objects.
    - **Data Manipulation**: Tools for reshaping, merging, and aggregating data.
    - **Time Series**: Robust support for working with time series data.

4.2.8 React==18.2.0

- **4.2.8.1 Description**: React is a popular JavaScript library for building user interfaces, particularly for single-page applications where you need a fast, interactive experience. Developed and maintained by Facebook, React enables developers to create large web applications that can update and render efficiently in response to data changes.
- **4.2.8.2 Key Features**:
    - **Virtual DOM**: Efficiently update and render just the right components when your data changes, leading to high performance.
    - **Component-Based:** Build encapsulated components that manage their own state, then compose them to make complex UIs.
    - **Strong Ecosystem**: Rich ecosystem with extensive tools, libraries, and community support.

4.3Methodology

4.3.1 User Interface:

The user interface of the healthcare chatbot, developed using React, is designed to be intuitive and user-friendly, ensuring a seamless experience for users seeking medical assistance. Key features include:

- **Clean and Simple Design:** A minimalist layout with clear navigation to enhance usability and accessibility.
- **Responsive Layout:** Ensures compatibility across various devices, providing a consistent experience on desktops, tablets, and smartphones.
- **Interactive Chat Window:** An easy-to-use chat interface where users can type in their symptoms or questions and receive instant responses.
- **Accessible Features:** Includes options for voice input and support for screen readers to cater to users with different needs.
- **Dynamic Content:** Real-time updates and personalized responses based on user input, leveraging React's efficient state management and rendering capabilities.

This UI design prioritizes ease of use, ensuring that users can quickly and efficiently access the healthcare information they need.

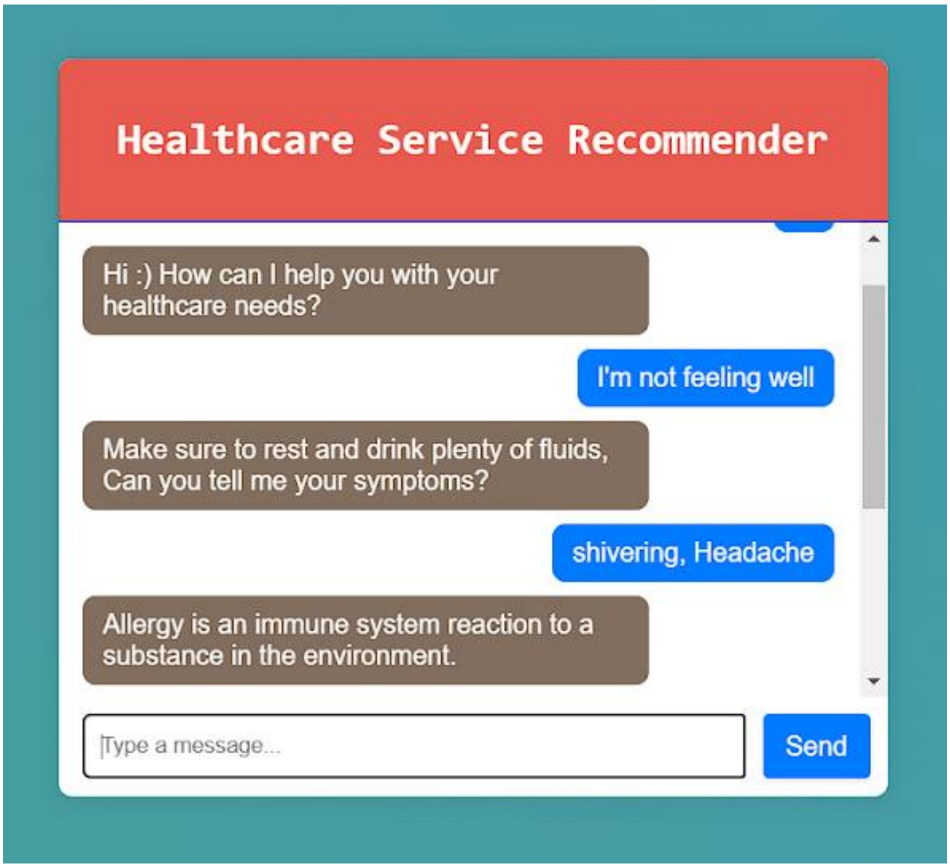

Fig.4.1 Chatbot Interface

4.3.2 Disease prediction model:

4.3.2.1 Imported Libraries: The code uses the Tkinter library for creating the GUI.

Other libraries such as NumPy and Pandas are used for data manipulation.

Scikit-learn is used for implementing machine learning algorithm, Random Forest.

4.3.2.2 Data Loading: The training and testing datasets are loaded from CSV files ("Training.csv" and "Testing.csv"). The datasets contain symptoms and corresponding disease labels.

4.3.2.3 Data Preprocessing: The code replaces disease labels with numerical values for both training and testing datasets.

4.3.2.4 Machine Learning Algorithms: SVM is implemented using the Support Vector Classifier class from scikit-learn.

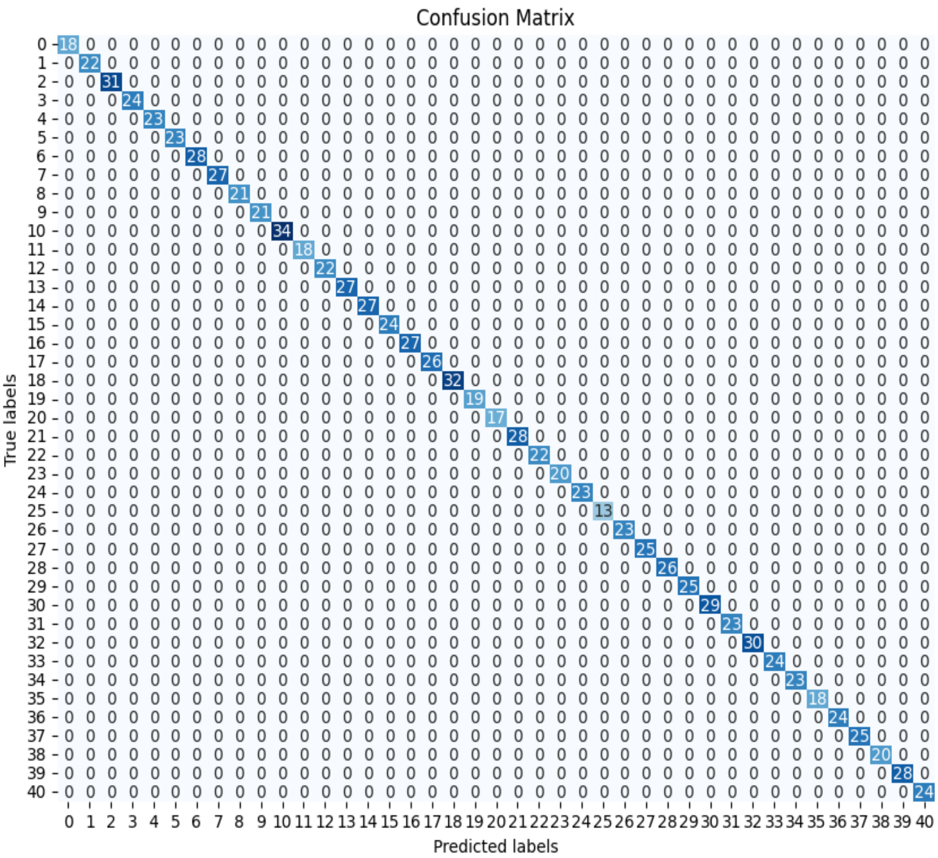

Fig.4.2 Confusion Matrix for SVC (Disease Prediction)

4.3.3 Drug recommendation model:

This module implements a deep learning model to classify drugs efficiently, based on patient attributes and drug descriptions. The dataset, consisting of 5220 entries, encapsulates crucial information including patient demographics (gender and age) and textual descriptions of drugs alongside associated diseases. Initial preprocessing involves encoding categorical variables and tokenizing drug descriptions to facilitate further analysis. Dataset is cleansed to ensure the quality of input data, thereby optimizing model performance. After data preprocessing, a Keras sequential model is used to train the dataset. The model architecture encompasses an embedding layer to convert drug descriptions into dense vector representations to capture temporal dependencies effectively. Hyperparameters like epochs and batch size are optimized.

4.3.4 NLP implementation:

Natural Language Processing has been utilized to create a dynamic and responsive chatbot. By leveraging NLP techniques, the chatbot can interpret and understand user input more accurately. NLP enables the chatbot system to extract key information and detect the context of the user input, allowing for more contextually relevant responses.

The dataset utilized in this model consists of intents which in turn comprise patterns of questions, tags and responses to the same. Each intent undergoes tokenization, normalization, and conversion into a "bag-of-words" representation. Unique words, classes, and documents are identified for training.

A Keras Sequential model is trained on the dataset using appropriate activation functions for each layer. The model is trained using bag-of-words representations and encoded labels. Hyperparameters like epochs and batch size are optimized. When the chatbot is provided with a user input, preprocessing, tokenization and classification of the input is performed. Based on classification, the function retrieves or generates a response. The chatbot interacts with users in a continuous loop. Error handling and user feedback are incorporated for improvement.

Chapter 5

Code Templates

Here are some important code snippets demonstrate key functionalities.

This Code Snippet Depicts the Recommend Drug Function:

```python
def recommend_drug(disease, gender, age):
    gender = gender.capitalize()
    new_data = pd.DataFrame({'Disease': [disease], 'Gender': [gender], 'Age': [age]})
    new_data['Disease'] = label_encoders['Disease'].transform(new_data['Disease'])
    new_data['Gender'] = label_encoders['Gender'].transform(new_data['Gender'])
    new_data['Age'] = scaler.transform(new_data[['Age']])
    predicted_drug_prob = best_model.predict(new_data)
    predicted_drug_index = np.argmax(predicted_drug_prob, axis=1)
    predicted_drug = label_encoder_drug.inverse_transform(predicted_drug_index)
    drug_dict = {
        "drug": predicted_drug[0]
    }
```

Fig. 5.1 Recommend Drug Function

This Code Snippet Depicts the Recommend Hospital Function:

```python
def recommend_hospital(state, city, pincode):
    input_location = state + ' ' + city + ' ' + str(pincode)

    input_tfidf = tfidf_vectorizer.transform([input_location])

    similarity_scores = cosine_similarity(input_tfidf, tfidf_matrix)

    most_similar_index = similarity_scores.argmax()

    recommended_hospital = hospital_data.iloc[most_similar_index]

    hospital_dict = {
        'Hospital': recommended_hospital['Hospital'],
        'State': recommended_hospital['State'],
        'City': recommended_hospital['City'],
        'Pincode': recommended_hospital['Pincode']
    }

    return hospital_dict
```

Fig. 5.2 Recommend Hospital Function

This Code Snippet Depicts the Recommend Doctor Function:

```python
def recommend_doctor(specialization, city, state):
    input_search_key = specialization.lower() + ' ' + city.lower() + ' ' + state.lower()

    input_tfidf = tfidf_vectorizer.transform([input_search_key])

    similarity_scores = cosine_similarity(input_tfidf, tfidf_matrix)

    # Handling the case where no doctors match the input search key
    if similarity_scores.max() == 0:
        return {"message": "Sorry, no doctors found matching the provided criteria."}

    most_similar_index = similarity_scores.argmax()

    recommended_doctor = doctors_data.iloc[most_similar_index]

    doctor_dict = {
        'Doctor': recommended_doctor['doctor'],
        'Specialization': recommended_doctor['specialization'],
        'City': recommended_doctor['city'],
        'State': recommended_doctor['state'],
        'Address': recommended_doctor['address'],
        'Link': recommended_doctor['link']
    }
    print(doctor_dict)
    return {"Recommended_Doctors": doctor_dict}
```

Fig. 5.3 Recommend Doctor Function

This Code Snippet Depicts the Disease Prediction Function:

```python
# Initialize feature vector with zeros
num_features = 132  # Adjust this based on the actual number of features expected by svc_model
feature_vector = [0] * num_features

# Encode symptoms in the feature vector
for symptom in symptoms:
    if symptom in symptoms_dict:
        feature_vector[symptoms_dict[symptom]] = 1  # Set the corresponding feature to 1 if the symptom is present

# Predict the disease
x = loaded_svc_model.predict([feature_vector])

# Map the predicted value to the corresponding disease
x_disease = diseases_list[x[0]]

# Save the predicted disease to a text file
with open("predicted_disease.txt", "w") as f:
    f.write(x_disease)

return x_disease, x[0]
```

Fig. 5.4 Disease Prediction Function

Chapter 6

Testing

6.1 Chatbot Functionality:

A function cleans, tokenizes, and classifies user input. Based on classification, the function retrieves or generates a response.

6.2 Interaction and Evaluation:

The chatbot interacts with users in a continuous loop. Error handling and user feedback are incorporated for improvement.

6.3 Unit Testing

Unit testing verified the functionality of individual components such as text processing, classification, and response generation.

Table No. 6.1 Unit Testing

Test Case	Description
Input Sanitization	Ensured correct handling of special characters and case variations.
Tokenization	Verified accurate splitting of sentences into tokens.
Classification	Tested accurate classification of user inputs.

6.4 Integration Testing

Integration testing validated the interaction between modules like NLP, disease prediction (SVM), drug recommendation (CNN), and hospital/doctor recommendation (Cosine Similarity).

Table No. 6.2 Integration Testing

Test Case	Description
NLP to SVM	Checked symptom classification and forwarding to the SVM model.
SVM to Drug Recommendation	Verified that disease predictions triggered appropriate drug recommendations.
NLP to Hospital/Doctor Recommendation	Ensured accurate recommendations for healthcare facilities.

6.5 System Testing

System testing evaluated the entire system's ability to handle real-time data and provide solutions.**Table No. 6.3 System Testing**

Test Case	Description
End-to-End Disease Prediction	Simulated real-world scenarios for symptom input and disease prediction.
Drug Recommendation Accuracy	Tested drug suggestions using the CNN model.
Frontend Responsiveness	Evaluated the usability and interaction quality of the frontend.

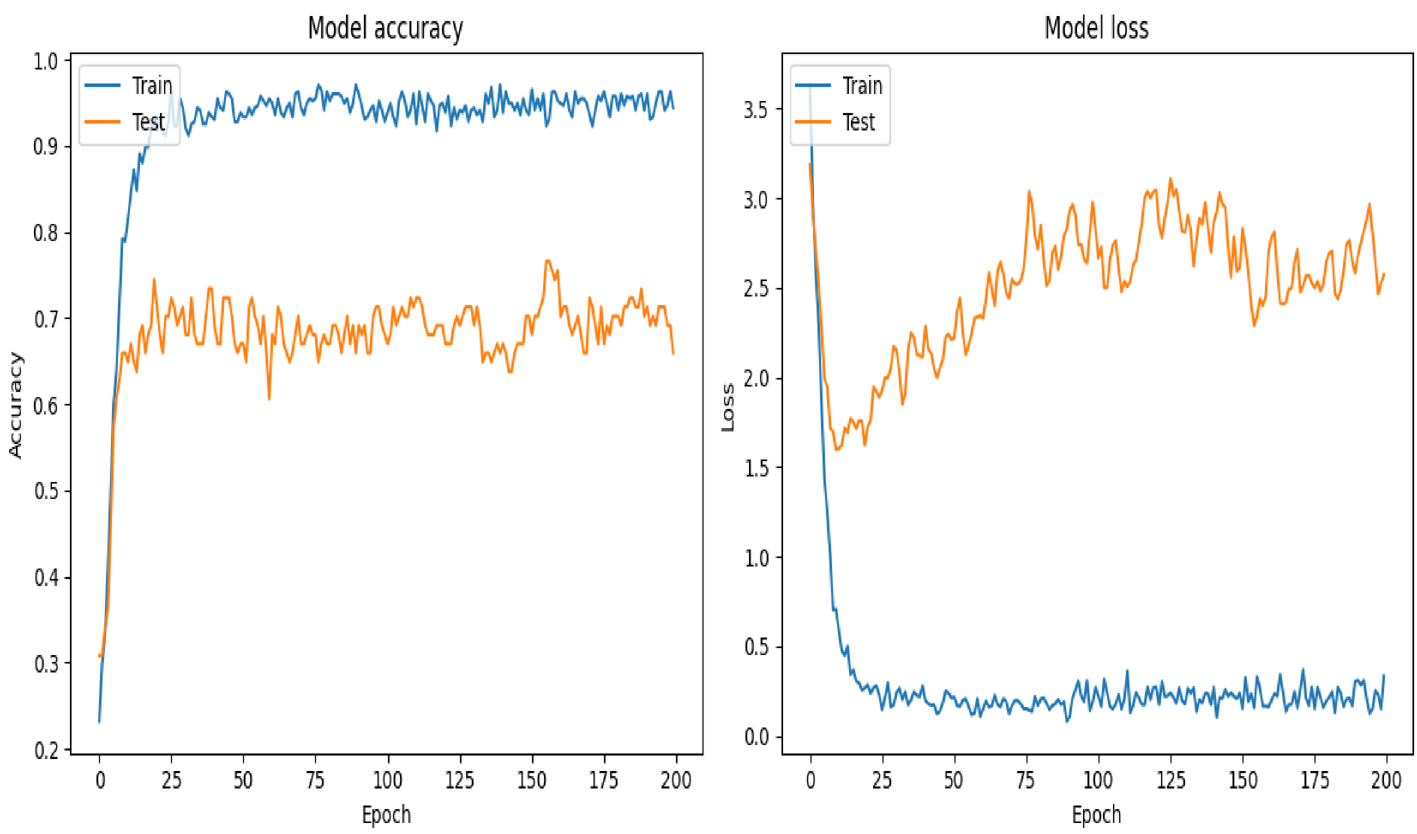

Fig. 6.1 Accuracy and Loss Function for NLP model

Chapter 7

Results and Discussions

The proposed Healthcare Service Recommendation System, integrating advanced technologies such as Natural Language Processing (NLP), Machine Learning (ML), and Cloud Computing, has shown promising results in enhancing current medical systems. The system aims to provide precise and rapid solutions for disease prediction, drug recommendation, and facilitating prompt access to nearby hospitals and qualified doctors. The results demonstrate the feasibility and reliability of this approach for predicting diseases and medications based on symptoms and offering valuable information regarding healthcare providers.

Key results include:

- **High Accuracy in Disease Prediction and Drug Recommendation**: The disease prediction model, utilizing a Support Vector Machine (SVM) classifier, achieved an impressive accuracy of 97%, as depicted by the confusion matrix in Figure 6. The NLP model used for handling diverse medical inquiries showcased an accuracy of 98%, as shown in Figure 5.
- **User Interface and Accessibility:** The system features a dynamic and responsive frontend developed using the React framework, enhancing user interaction and accessibility (Figure 4). Deploying the project on a cloud platform further broadens its accessibility, ensuring that users can access the system from various locations and devices.
- **Robust Methodology:** The integration of SVM, Convolutional Neural Networks (CNN), and cosine similarity models, along with rigorous data preprocessing and hyperparameter tuning, contributes to the system's notable accuracy and adaptability.

Despite these promising results, several challenges are directly linked to real-world deployment:

- **Data Quality:** Ensuring the accuracy and comprehensiveness of the data is critical for reliable recommendations.
- **Model Complexity and Interpretability:** Balancing model complexity with interpretability is essential to ensure that healthcare professionals can understand and trust the system's recommendations.
- **Computational Resources and Hardware Limitations:** The system requires substantial computational resources, and hardware limitations could impact its performance.
- **Continuous Model Updates:** The need for continuous updates to adapt to evolving medical scenarios is vital for maintaining the system's relevance and effectiveness.
- **Privacy Concerns:** Implementing robust data privacy measures is crucial, especially when dealing with sensitive medical information.

Future endeavours could explore integrating additional data sources and advanced deep learning techniques to further enhance the system's accuracy in disease detection and drug recommendations. Such advancements could streamline medical diagnosis processes, alleviate the burden on healthcare professionals, and improve patient care. Ongoing research efforts should also focus on optimizing the system's scalability and resilience, ensuring consistent performance across various user demands and data scenarios.

Chapter 8

Conclusion and Future Work

In this study, advanced technologies such as NLP, ML and Cloud Computing are integrated for creation of the proposed Healthcare Service Recommendation System. This system aims to improve the present medical systems by providing precise and rapid solutions for disease prediction, drug recommendation, and facilitating prompt access to nearby hospitals and qualified doctors. Through seamless integration of advanced techniques and real-time data, it provides healthcare professionals and patients alike with invaluable insights, allowing early response actions for health conditions and optimizing treatment outcomes. The results show that the method is feasible and reliable for predicting diseases and medications based on symptoms provided and providing valuable information regarding doctors and hospitals. Fig. 4 shows a dynamic and responsive frontend created in React framework while Fig. 5 showcases the accuracy and loss functions of the model used for NLP showing the impressive accuracy of 98%. The integration of NLP enriches the project's capabilities, allowing it to handle a diverse range of medical inquiries. The disease prediction model, utilizing the SVM, boasts an accuracy of 97%. Fig. 6 shows the confusion matrix for the SVM model. This methodology illustrates the application of Support Vector Machine Classifier, CNN, and Cosine Similarity models for disease detection, drug recommendation, NLP, and Hospital and doctor recommendation, respectively. Through rigorous data preprocessing and hyperparameter tuning, these models achieve notable accuracy and adaptability. Furthermore, deploying the project on a cloud platform broadens its accessibility to users, thereby extending its potential impact.

It is important to understand and acknowledge the deep-rooted challenges that are directly linked with the real-world deployment, such as data quality, model complexity, interpretability, computational resources, hardware limitations and the need for continuous model updates to adapt to evolving medical scenarios. Moreover, there is significant need to consider privacy concerns when implementing such systems at a broader level.

Looking forward, future endeavours could explore the integration of additional data sources and advanced deep learning techniques to further enhance disease detection and drug recommendation accuracy. Such advancements could streamline medical diagnosis processes, alleviating the burden on healthcare professionals and enhancing patient care. In addition, ongoing research efforts could focus on optimizing the system's scalability and resilience, ensuring consistent performance across various user demands and data scenarios. Collaborative partnerships with medical institutions and experts could provide valuable insights for refining and expanding the system's capabilities to address emerging healthcare challenges effectively. Through continuous innovation and refinement, this project has the potential to revolutionize medical diagnosis and treatment practices, benefiting healthcare providers and patients alike.

References

[1] S. Baker and W. Xiang, "Artificial Intelligence of Things for Smarter Healthcare: A Survey of Advancements, Challenges, and Opportunities," IEEE Access, vol. 9, pp. 101522-101544, 2021.

[2] M.H. Kuo, "Opportunities and Challenges of Cloud Computing to Improve Health Care Services," Journal of Medical Internet Research, vol. 13, no. 3, p. e67, 2011.

[3] C. Gao et al., "A Survey of Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions," ACM Transactions on Recommender Systems, vol. 15, no. 2, pp. 1-38, 2021.

[4] P. Agarwal et al., Eds., Artificial Intelligence for Smart Healthcare. Cham: Springer International Publishing, 2020.

[5] R.F Mansour "Artificial Intelligence and Internet of Things Enabled Disease Diagnosis Model for Smart Healthcare Systems", IEEE Access, vol 8, pp 211948-211961, 2020.

[6] Y. Cai, F. Yu, M. Kumar, R. Gladney, and J. Mostafa, "Health Recommender Systems Development, Usage, and Evaluation from 2010 to 2022: A Scoping Review," International Journal of Environmental Research and Public Health, vol. 19, no. 22, p. 15115, Nov. 2022.

[7] M. Etemadi et al., "A systematic review of healthcare recommender systems: Open issues, challenges, and techniques," Expert Systems with Applications, vol. 213, p. 118823, 2022.

[8] R. De Croon et al., "Health Recommender Systems: Systematic Review," Journal of Medical Internet Research, vol. 23, p. e18035, 2021.

[9] P. Kumar and A. Kumar, "A Review of Healthcare Recommendation Systems Using Several Categories of Filtering and Machine Learning-Based Methods," in 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2022, pp. 762-768.

[10] P. Howell, A. Aryal, and C. Wu, "Web-Based Patient Recommender Systems for Preventive Care: Protocol for Empirical Research Propositions," JMIR Research Protocols, vol. 12, p. e43316, Mar. 2023.

[11] Y. Sun et al., "Development and Evaluation of Health Recommender Systems: Systematic Scoping Review and Evidence Mapping," Journal of Medical Internet Research, vol. 25, p. e38184, 2023.

[12] P. Lops, M. Gemmis, and G. Semeraro, "Content-based Recommender systems: State of the Art and Trends," in Boston: Springer, 2011, pp. 73–105.

[13] P. Howell, A. Aryal, and C. Wu, "Web-Based Patient Recommender Systems for Preventive Care: Protocol for Empirical Research Proposition," JMIR Research Protocols, vol. 12, p. e43316, 2023.

[14] T. N. T. Tran, A. Felfernig, C. Trattner, and A. Holzinger, "Recommender systems in the healthcare domain: state-of-the-art and research issues," Journal of Intelligent Information Systems, vol. 57, pp. 171-201, 2020.

[15] J. G. D. Ochoa, O. Csiszár, and T. Schimper, "Medical recommender systems based on continuous-valued logic and multi-criteria decision operators, using interpretable neural networks," BMC Medical Informatics and Decision Making, vol. 21, p. 186, 2021.