An AI-Based Medical Chatbot Model for Infectious Disease Prediction

A
Project Report
Submitted in partial fulfilment of the
Requirements for the award of the Degree of

BACHELOR OF ENGINEERING
IN

INFORMATION TECHNOLOGY

By

1602-20-737-051 - B. Suresh Kumar

Under the guidance of

Dr. S. Sree Lakshmi

Assistant Professor (Sr.Scale)

Department of Information Technology
Vasavi College of Engineering (Autonomous)
ACCREDITED BY NAAC WITH 'A++' GRADE
(Affiliated to Osmania University)
Ibrahimbagh, Hyderabad-31 2024

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Information Technology

DECLARATION BY THE CANDIDATE

I, **B.Suresh Kumar** bearing hall ticket number 1602-20-737-51 hereby declare that the project report entitled "**An AI-Based Medical Chatbot Model for Infectious Disease Prediction"** under the guidance of **Dr. S. Sree Laksmi ,Assistant Professor(Sr.Scale),**Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by me and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

B. Suresh Kumar (1602-20-737-051)

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Information Technology

BONAFIDE CERTIFICATE

This is to certify that the project entitled "**An AI-Based Medical Chatbot Model for Infectious Disease Prediction"** being submitted by **B. Suresh Kumar** bearing **1602-20-737-051** in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Information Technology is a record of bonafide work carried out by them under my guidance.

Dr. S . Sree Lakshmi	**Dr. K . Ram Mohan Rao**
Assistant Professor (Sr. Scale)	**Professor,**
Internal Guide	**HOD,IT**

External Examiner

ACKNOWLEDGEMENT

The satisfaction that accompanies that the successful completion of the Main project would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

It is with immense pleasure that we would like to take the opportunity to express our humble gratitude to **Dr. S.Sree Lakshmi Assistant Professor (Sr.Scale), Information Technology** under whom we executed this project. We are also grateful to **S.Aruna Associate Professor, Information Technology** for her guidance. Their constant guidance and willingness to share their vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks.

We are very much thankful to **Dr. K. Ram Mohan Rao, Professor, HOD, Information Technology** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to **Dr. S.V. Ramana, Principal** of **Vasavi College of Engineering and Management** for providing facilities. Not to forget, we thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported me in completing my project in time.

Abstract

The purpose of this paper is to show concisely how we can promote chatbots in the medical sector and cure infectious diseases. We can create awareness through the users and the users can get proper medical solutions to prevent disease. We created a preliminary training model and a study report to improve human interaction in databases in 2021. Through natural language processing, we describe the human behaviors and characteristics of the chatbot. In this paper, we propose an AI Chatbot interaction and prediction model using a deep feedforward multilayer perceptron. Our analysis discovered a gap in knowledge about theoretical guidelines and practical recommendations for creating AI chatbots for lifestyle improvement programs. A brief comparison of our proposed model concerning the time complexity and accuracy of testing is also discussed in this paper. In our work, the loss is a minimum of 0.1232 and the highest accuracy is 94.32%.

This study describes the functionalities and possible applications of medical chatbots and explores the accompanying challenges posed by the use of these emerging technologies during such health crises mainly posed by pandemics. We believe that our findings will help researchers get a better understanding of the layout and applications of these revolutionary technologies, which will be required for continuous improvement in medical chatbot functionality and will be useful in avoiding COVID-19. INDEX TERMS Artificial intelligence, chatbot, LSTM algorithm, machine learning, natural language processing, query processing.

Table of Contents

List of Figures **7**

List of Abbreviations **8**

1 INTRODUCTION

A. Background of the Study 2

B. Problem Statement 3

C. Aims and Objectives 4-5

D. Scope and Limitations of the work 5-12

2 LITERATURE SURVEY 13-18

3 EXISTING METHOD 19-23

4 PROPOSED SYSTEM

4.1 Background LSTM

A.Working with LSTM 25

B. Recurrent Neural Network 26-28

C. Decision Tree Model 28-30

4.2 Proposed Model **30-32**

5 RESULT

A. Datasets 33

B. Results 34-41

6 CONCLUSION AND FUTURE SCOPE 42

REFERENCES 43

APPENDIX **44-56**

LIST OF FIGURES

Fig. No	Description	Page No
1	Functioning of Forget gate in LSTM algorithm	25
2	Functioning of the input gate in LSTM algorithm.	25
3	Functioning of output gate in LSTM algorithm	26
4	Functioning of the cell gate in the LSTM algorithm	27
5	Graph showing test set versus predicted values using LSTM in our model	27
6	Demonstration of the RNN algorithm as applied to chatbot model	28
7	Demonstration of Decision Tree Model as applied To our chatbot model	30
8	Flowchart of the overall chatbot system	42

LIST OF ABBREVIATIONS

HTML: Hyper Text Markup Language

CSS: Cascading Style Sheets

JS: JavaScript

CNN: Convolutional neural network

1. INTRODUCTION

Covid-19 is an ailment because of the SARS-CoV-2 virus. ‘World Health Organization (WHO)’ has been declared a pandemic on March 11, 2020. Around 15 million human beings have been affected worldwide by more than one million deaths by covid-19. From the start, the affordability and sustainability of oxygen have been a problem in poor and underdeveloped countries. Oxygen is a very important medicine for treating hospitalized patients with Covid-19. According to PATH, the organization that works with global institutions and businesses to tackle health problems, the demand for Oxygen cylinders has been growing between 6%8% eacdaily India. When someone gets severe Covid-19, the oxygen levels in the body get low.

Patients suffered from enormous fevers, coughs, and lost sense of taste and smell, all of which became problematic when the SARS-CoV-2 virus invaded them. As a result, to prevent such massive problems associated with COVID-19 and facilitate a quick cure method, we have developed a Chatbot that facility an interconnection between users and computers in a natural manner. In the twentyfirst century, artificial intelligence algorithms have been used to create a revolutionary medium with which users can interact with their needs to prevent and resolve acoustic problems easily; one of these is chatbots, a modern form of interaction. Chatbot systems use human interactions such as decision-making performing daily tasks, replying to users quickly, and solving problems like a human. Chatbots are also called communicational agents or answering engines.

Developers and programmers train these chatterbots with the help of artificial intelligence and machine learning systems to interact with users through voice commands, over communication, or text-based messages. We focused on developing the core of AI to interact more efficiently with users and get to understand the user’s queries, and provide the user with an appropriate solution. This application works in a very simple way because the data is already put in the system. A few things like matching patterns, NLP (Natural Language Processing), and data mining are used in the application to train the system. Chatbots match the

user input texts or voices with the previously included data and give replies according to the data.

This knowledge or data has been taken from various sources. Everything in this generation is getting in touch with the web. It is effective to use a method to manage to advantage anything at your doorway. AI chatbots can be deployed on the web or in mobile applications as software or on computers. It works 24/7 without any delay, only a good internet connection and power have to provide to work smoothly. Chatbots have a significant role in various fields, especially in the medical field. The first established medical chatbot –ELIZA– was programmed in 1966 to simulate a text-based conversation with a Psychotherapist.

The top platform that lends invaluable support to that bot development, Amazon Alexa, has more than 0.1 million applications, many of which are for health. The 'World Health Organization (WHO) on the renowned social media Facebook Messenger to counter wrong information and provide accurate information about COVID-19 without any delay, launched a chatbot. As Chatbots are becoming communicational agent in the digital world, it opens many ways for demonstrating a change in behavior for disease prevention and promoting healthrelated actions on a large scale. The rest of the paper is organized as follows. Section ''Related Work'' defines some existing state-of-the-art methods for the AIassisted Chatbot in health care. Therefore, section-III describes the necessary background of the models that we have used in our research work. The section-IV discusses the proposed algorithm and overall workflow of the proposed AI chatbot system. Then a detailed result analysis and a brief comparison among some popular, highly efficient methods are described in section V. Then, a discussion on the applications and benefits of our proposed model is discussed in section VI and section VII, respectively. Eventually, we conclude the paper in section VIII by highlighting some of the future research scopes.

1.1 **Background of the Study**

Machine Learning is the domain that uses past data for predicting. Machine Learning is the understanding of computer system under which the Machine Learning model learn from data and experience. The machine learning algorithm has two phases: 1) Training & 2) Testing. To predict the disease from a patient's symptoms and from the history of the patient, machine learning technology is struggling from past decades. Healthcare issues can be solved efficiently by using Machine Learning Technology.

We are applying complete machine learning concepts to keep the track of patient's health. ML model allows us to build models to get quickly cleaned and processed data and deliver results faster. By using this system doctors will make good decisions related to patient diagnoses and according to that, good treatment will be given to the patient, which increases improvement in patient healthcare services.

To introduce machine learning in the medical field, healthcare is the prime example. For the prediction of diseases, the existing will be done on linear, KNN, Decision Tree algorithm. Specialists find it difficult to make decisions about the illnesses because they may not have skills in all areas. To address this issue, it is necessary to develop a disease prediction system that combines medical knowledge with an integrated system to produce the biggest results and can help society [1].

1.2 Problem Statement

- According to the World-data Info there are 0.09 doctors per 1000 Inhabitants in Cameroon.
- Specialists find it difficult to make decisions about the illnesses because they may not have skills in all areas.
- Inadequate Medical facilities distributed across various Hospitals make it more time-consuming for a Patient to know which disease they are suffering from. 13

1.3 Aims and Objectives

The project proposes to create a DISEASE PREDICTION WEB-APP USING MACHINE LEARNING that will provide the most accurate prediction of various patient illnesses. The prediction will be done thanks to collected datasets on Kaggle, and training will be done on those datasets, which will be mounted on a web app coded in Spyder and run using Streamlit.

This multiple disease prediction system will be collected on a sample hospital website and can only be accessed when given access (logging in). The prediction will be faster as compared to using laboratory equipment, and efficiency is greater than 94% for each trained and tested Model. I decided that the following modules could be included in the development process:

- ➔ Aim Rapidly determine if a patient is Infected with a Particular Disease or not, with a prediction Accuracy of up to 94% for each of the illnesses.
- ➔ Objectives

a) Assist the General Practitioner / Specialist in their day-to-day lifeSaving battle.

b) Educate the population with respect to the major illnesses faced by the population in our community (such as diabetes, breast cancer, Parkinson's, and Heart Disease), especially how they are transmitted, how they can be avoided, and the preventive measures put in place by the Government.

1.4 SCOPE AND LIMITATIONS OF THE WORK

1.4.1 SCOPE In the future, more models could be trained and used in various sectors, enhancing efficiency by considering more symptoms to predict disease.

1.4.2 LIMITATIONS OF THE WORK: Through the applications, it will be possible to predict if a User is sick of a Particular disease or not. The primary limitation of this application is that it can only be used by Medical doctors or high-class people who are able to afford medical equipment in their home for the collections of samples, which will be input into the applications.

1.4.3 ALGORITHM USE For 3 disease prediction, the SVM algorithm is used, and the Supervise algorithm is the Thus, in supervision learning, we feed our data to the

Machine Learning model, and the Machine Learning Model learns from the data and its respective labels. So in this case, we train our models with several pieces of medical information (Such as the blood glucose level and the insulin level of patients, along with whether the person has diabetes or not).

So this acts as a label to show whether the person is Diabetic or Not. So once we feed this data to our support vector machine, what happens is that it tries to plot this data. And once it plots, try to find hyperplane.

1.4.4 WORKFLOW OF MACHINE LEARNING:

1-From Kaggle.com I will download the datasets, and then try to train our data with the models and the respective levels and feed it into our Models.

2: I will pre-process the data, and we will try to analyze it. This data will be very suitable to feed the machine learning model, and we need to standardize this data. (Because there are a lot of attributes here, there is a lot of Medical Information.) So standardizing this data is important.

3: So once I pre-process the data, I will split the data into Train and Test. So I will train our machine Learning Model with training data.

4: So once I split the test data into Training data and testing data, we will feed this to our Support Vector Machine models. So we will be Using classification models, where these models will classify whether the patient is Diabetic or nondiabetic. (This is for the case of Diabetic prediction.) 27 5-So once I have a Train Vector Support Machine Classifier, when we give our new data, I can now predict whether the Person is Diabetic or non-diabetic.

All over the world, chronic diseases are a critical issue in the healthcare domain. According to the medical statement, due to chronic diseases, the death rate of humans increases. The treatments given for this disease consume over 70% of the patient's income. Hence, it is highly essential to minimize the patient's risk factor that leads to death. The advancement in medical research makes healthrelated data collection easier [1, 2]. The healthcare data includes the demographics, medical analysis reports, and the history of disease of the patient.

The diseases caused could be varied based on the regions and the living habitats in that region. Hence, along with the disease data, the environmental condition and the living habitat of the patient should also be recorded in the data set.

In recent years, the healthcare domain is evolving more due to the integration of information technology (IT) in it. The intention to integrate IT in healthcare is to make the life of an individual more affordable with comfort as smartphones made one's life easier [3]. This could be possible by making healthcare to be intelligent, for instance, the invention of the smart ambulance, smart hospital facilities, and so on, which helps the patients and doctors in several ways [4].

The research on a specified region for patients affected by chronic diseases every year had been held and found that the difference between the patients in genderwise is very small, and it is found that the large number of patients were admitted in the year 2014 for treating chronic diseases.

The use of structured and unstructured data provides highly accurate results instead of using only structured data. Since the unstructured data includes the doctor's records on the patients related to diseases and the patient's symptoms and grievances faced by them, explained by themselves, which is an added advantage when used along with the structured data that consists of the patient demographics, disease details, living habitats, and laboratory test results [5, 6]. It is difficult to diagnose rare diseases. Hence, the use of self-reported behavioral data helps differentiate the individuals with rare diseases from the ones with common chronic diseases. By using machine learning approaches along with questionnaires, it is believed that the identification of rare diseases is highly possible [7].

In the last decade, some innovative technologies had been introduced to rapidly collect the data such as MRI (magnetic resonant imaging) readouts, ultrasonography, social media gained data, and electronically gained activity, behavioral, and clinical data. These big data sets of healthcare are highdimensional, which means the number of features recorder per observation might be greater than the total observations. They are noisy, sparse, cross-sectional, and lacks statistical power. By applying machine learning techniques, the issues in the high-dimensional data sets can be overcome [8].

Machine learning contributes more in several domains. Many of the complex models make use of exiting larger training data, simultaneously at the edge of a major shift in healthcare epidemiology [9]. These data can enhance the knowledge

gain in the risk factors of diseases to reduce healthcare-associated infections, improve patient risk stratification, and find the way of transmitting the infectious diseases [10].

Machine learning can facilitate the analysis of laboratory results and other details of patients for the early detection of diseases. The low-level data could be converted to high-level knowledge via knowledge discovery in the database so as to gain knowledge about the disease patterns to support early detection [11]. The data collected for creating a data set should be preprocessed for its missing values, and then only the important features needed for accurate disease prediction are selected so as to enhance the prediction accuracy and minimizing the time taken for model training [12].

In the era of the Internet and technologies, people are not concerned about their health and lives. As everyone is interested in surfing and social media activities, they ignore visiting hospitals for their health checkup. By taking this activity as an advantage, a machine learning model that takes the symptoms given as input and predicts the possibility and risk of the disease affected or the development of such diseases in an individual should be developed [13, 14]. The more common chronic diseases are diabetes, cardiovascular diseases, cancer, strokes, hepatitis C, and arthritis.

As these diseases persist for a long time and have a high mortality rate, the diagnosis of such diseases is highly important in the healthcare domain. Foreseeing the disease can help take preventive actions and avoid getting affected by it, and early detection of it can help provide better treatment [15]. There are various techniques in machine learning such as supervised, semisupervised, unsupervised, reinforcement, evolutionary, and deep learning.

The problem is associated with the processing of extracted features from real data and structured as vectors [16]. The processing quality is based on the proper combination of those vectors. But, most of the times, the high dimensionality of the vectors or the discrepancies in the data makes a big issue. Hence, it is important to reduce the dimensionality of the data set even if it leads to a little loss of details to make the data set a highly compatible dimension. This reduction in the

dimensionality of the data set improves the model performance [17].

The system of chronic diseases management is essential for those affected by such diseases and in need of proper medical assessment and treatment information [18]. Also, this system can be useful for individuals who are in need of self-care to improve their health condition, since it is proved that selfmanagement is the primary care of those with chronic diseases, and it is considered as the unavoidable part of treatment. With the use of mobile applications, the health information of patients can be recorded, and they serve as a better tool to enable self-management [19].

To effectively predict a disease, information such as narration about the symptoms felt by the patients, details of consultation with medical practitioners, lab examination results, and computed tomography and X-ray images [20]. Little research is performed in identifying the accuracy and predictive power for developing a machine learning model with only information from lab examination results for the diagnosis of diseases.

And, for performance enhancement, ensemble machine learning and deep learning model can be used [21, 22]. In the healthcare domain, artificial intelligence (AI) plays a major role in automating the roles involved in disease diagnosis and treatment suggestions and also schedules perfect timing by the medical practitioners to perform various obligations that cannot be automated [23].

The major objective of the proposed system is to identify and predict chronic disease in an individual using a machine learning approach [24, 25]. The data set comprises both the structured data that includes the patient's age, gender, height, weight, and so on, excluding the patient's personal information such as name and ID, and the unstructured data that includes the patient's symptoms, information related to consultation about the disease with the doctors, and the living habits of that individual [26]. These data are preprocessed for finding the missing values. They are then reconstructed to increase the quality of the model, thereby increasing the prediction accuracy. For prediction, the machine learning algorithms such as CNN and KNN are used [27, 28].

Creating a medical app that utilizes machine learning to predict seasonal diseases based on user symptoms and provides appropriate recommendations is a valuable initiative to assist users in managing their health effectively.

- Symptom Input: Users can input their symptoms into the app via a userfriendly interface. The symptoms can range from common cold symptoms like sneezing and coughing to more severe symptoms associated with diseases like malaria or cholera.

- Machine Learning Prediction: The app utilizes a trained machine learning model to analyze the input symptoms and predict the most likely seasonal disease. The model is trained on a dataset of symptoms and corresponding diseases, covering a wide range of seasonal illnesses.

- Disease Severity Classification: Based on the predicted disease, the app classifies the severity into categories such as mild, moderate, or severe. Diseases like common cold may be categorized as mild, while diseases like malaria or cholera are classified as severe.

- Recommendations and Advice:

- Mild Disease (e.g., Common Cold): If the predicted disease is mild, the app provides recommendations for over-the-counter medicines or home remedies to alleviate symptoms. It may suggest rest, hydration, and specific medications such as antihistamines or decongestants.

- Severe Disease (e.g., Malaria, Cholera): For severe diseases, the app advises users to seek immediate medical attention from a nearby doctor or healthcare facility. It provides information on the urgency of medical intervention and emphasizes the importance of timely treatment.

- Emergency Contacts: The app includes a feature to quickly access emergency contact numbers for local healthcare services, hospitals, or clinics. Users can easily find and contact healthcare providers in case of emergencies or severe symptoms.

- User Profile and History: Users can create profiles within the app to track their symptoms, receive personalized recommendations, and maintain a history of their health conditions and treatments. This feature enables users to monitor their health over time and share relevant information with healthcare providers if necessary.

- Privacy and Security: The app prioritizes user privacy and ensures the secure handling of sensitive health data. It complies with relevant regulations such as HIPAA (Health Insurance Portability and Accountability Act) to safeguard user information and maintain confidentiality.

The app can be deployed on mobile platforms such as iOS and Android, making it accessible to a wide range of users. It can be distributed through app stores or as a web-based application for seamless access across different devices.

HealthGuard provides users with a convenient and reliable tool for predicting seasonal diseases based on symptoms and receiving appropriate recommendations. By leveraging machine learning and personalized health advice, the app empowers users to take proactive steps in managing their health and seeking timely medical assistance when needed.

In recent years, the integration of machine learning (ML) technology into healthcare applications has revolutionized the delivery of medical services, making healthcare more accessible, efficient, and personalized. Online medical apps utilizing ML algorithms have emerged as invaluable tools for both healthcare providers and patients, offering a wide range of functionalities to enhance diagnosis, treatment, and healthcare management. This introduction will provide an overview of the concept of an online medical app powered by machine learning, highlighting its potential benefits and key features.

HealthLink is an innovative online medical app that harnesses the power of machine learning to provide personalized healthcare solutions to users. Whether it's symptom diagnosis, medication management, or health monitoring, HealthLink offers a comprehensive suite of features designed to streamline the healthcare experience and empower users to make informed decisions about their health.

Symptom Checker and Diagnosis:

- Users can input their symptoms into the app, which employs machine learning algorithms to analyze the symptoms and generate potential diagnoses.
- The app provides users with personalized recommendations based on the predicted conditions, including information on the likelihood of each diagnosis, common treatments, and when to seek medical attention.

Medication Management:

- HealthLink helps users manage their medications by providing reminders for dosage schedules, refill alerts, and medication interactions.
- Machine learning algorithms analyze user data to personalize medication reminders based on factors such as dosage frequency, medication type, and user preferences.

Health Monitoring and Predictive Analytics:

- The app enables users to track various health metrics such as blood pressure, blood glucose levels, heart rate, and sleep patterns.
- Machine learning models analyze historical health data to identify trends, detect anomalies, and provide personalized insights into users' health status and potential risk factors.

Telemedicine and Consultation Services:

- HealthLink facilitates virtual consultations with healthcare professionals through video calls, chat, or voice calls.
- Machine learning algorithms assist healthcare providers in triaging patients, prioritizing appointments, and identifying urgent cases based on symptom severity and medical history.

Health Education and Content Recommendation:

- The app offers educational resources, articles, and videos on various health topics, tailored to users' interests and medical conditions.
- Machine learning algorithms analyze user interactions and feedback to recommend relevant content, improving user engagement and knowledge dissemination.

Community Support and Peer Networking:

- HealthLink fosters a supportive online community where users can connect with peers, share experiences, and seek advice on health-related issues.
- Machine learning algorithms facilitate community moderation, content curation, and personalized recommendations for user interactions.

HealthLink represents a paradigm shift in healthcare delivery, leveraging machine learning technology to provide personalized, accessible, and efficient medical services to users. By combining advanced algorithms with user-centric design principles, HealthLink empowers individuals to take control of their health, make informed decisions, and access high-quality healthcare resources anytime, anywhere. As the field of machine learning in healthcare continues to evolve, HealthLink remains at the forefront of innovation, driving positive outcomes and improving the overall well-being of its users.

2. LITERATURE SURVEY

[1] M. Jiang, Y. Chen, M. Liu, S. T. Rosenbloom, S. Mani, J. C. Denny, and H. Xu, "A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries," J. Am Med Inform Assoc, vol. 18, no. 5, pp. 601–606, 2011.

Objective: The authors' goal was to develop and evaluate machine-learning-based approaches to extracting clinical entities-including medical problems, tests, and

treatments, as well as their asserted status-from hospital discharge summaries written using natural language. This project was part of the 2010 Center of Informatics for Integrating Biology and the Bedside/Veterans Affairs (VA) naturallanguage-processing challenge.

Design: The authors implemented a machine-learning-based named entity recognition system for clinical text and systematically evaluated the contributions of different types of features and ML algorithms, using a training corpus of 349 annotated notes. Based on the results from training data, the authors developed a novel hybrid clinical entity extraction system, which integrated heuristic rulebased modules with the ML-base named entity recognition module. The authors applied the hybrid system to the concept extraction and assertion classification tasks in the challenge and evaluated its performance using a test data set with 477 annotated notes.

Measurements: Standard measures including precision, recall, and F-measure were calculated using the evaluation script provided by the Center of Informatics for Integrating Biology and the Bedside/VA challenge organizers. The overall performance for all three types of clinical entities and all six types of assertions across 477 annotated notes were considered as the primary metric in the challenge.

Results and discussion: Systematic evaluation on the training set showed that Conditional Random Fields outperformed Support Vector Machines, and semantic information from existing natural-language-processing systems largely improved performance, although contributions from different types of features varied. The authors' hybrid entity extraction system achieved a maximum overall F-score of 0.8391 for concept extraction (ranked second) and 0.9313 for assertion classification (ranked fourth, but not statistically different than the first three systems) on the test data set in the challenge.

[2] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang,"Disease prediction by machine learning over big data from healthcare communities" IEEE Access, vol. 5, no.1, pp.8869–8879, 2017.

With big data growth in biomedical and healthcare communities, accurate analysis of medical data benefits early disease detection, patient care, and community services. However, the analysis accuracy is reduced when the quality of medical data is incomplete. Moreover, different regions exhibit unique characteristics of certain regional diseases, which may weaken the prediction of disease outbreaks. In this paper, we streamline machine learning algorithms for effective prediction of chronic disease outbreak in disease-frequent communities. We experiment the modified prediction models over real-life hospital data collected from central China in 2013–2015. To overcome the difficulty of incomplete data, we use a latent factor model to reconstruct the missing data. We experiment on a regional chronic disease of cerebral infarction.

We propose a new convolutional neural network (CNN)-based multimodal disease risk prediction algorithm using structured and unstructured data from hospital. To the best of our knowledge, none of the existing work focused on both data types in the area of medical big data analytics. Compared with several typical prediction algorithms, the prediction accuracy of our proposed algorithm reaches 94.8% with a convergence speed, which is faster than that of the CNN-based unimodal disease risk prediction algorithm.

[3] Sayali Ambekar, Rashmi Phalnikar, "Disease Risk Prediction by Using Convolutional Neural Network" IEEE, 978-1-5386-5257-2/18, 2018.

Data analysis plays a significant role in handling a large amount of data in the healthcare. The previous medical researches based on handling and assimilate a huge amount of hospital data instead of prediction. Due to an enormous amount of data growth in the biomedical and healthcare field the accurate analysis of medical data becomes propitious for earlier detection of disease and patient care. However, the accuracy decreases when the medical data is partially missing.

To overcome the problem of missing medical data, we perform data cleaning and imputation to transform the incomplete data to complete data. We are working

on heart disease prediction on the basis of the dataset with help of Naïve bayes and KNN algorithm. To extend this work, we propose the disease risk prediction using structured data. We use convolutional neural network based unimodel disease risk prediction algorithm. The prediction accuracy of CNNUDRP algorithm reaches more than 65%. Moreover, this system answers the question related to disease which people face in their life.

[4] Naganna Chetty, Kunwar Singh Vaisla and Nagamma Patil, "An Improved Method for Disease Predictionusing Fuzzy Approach" IEEE, DOI 10.1109/ICACCE.2015.67, pp. 569-572, 2015.

Data mining is a process of extracting useful information from the huge amount of data. Data Mining has great scope in the field of medicine. This article deals with the working on PIMA and Liver-disorder datasets. Many researchers have proposed the use of K-nearest neighbor (KNN) algorithm for diabetes disease prediction. Some researchers have proposed a different approach by using K-means clustering for preprocessing and then using KNN for classification. These approaches resulted in poor classification accuracy or prediction. In our work we proposed and developed two different methods first one is fuzzy c-means clustering algorithm followed by a KNN classifier and second one is fuzzy cmeans clustering algorithm followed by fuzzy KNN classifier to improve the accuracy of classification.

We are successful in obtaining the better results than the existing methods for the given datasets. Our second approach produced better result than the first one. Classification is carried out using ten folds cross-validation technique.

[5] Dhiraj Dahiwade, Gajanan Patle and Ektaa Meshram, "Designing Disease Prediction Model Using Machine Learning Approach" IEEE Xplore Part Number: CFP19K25-ART; ISBN: 978-1-5386-7808-4, pp. 1211-1215, 2019.

Now-a-days, people face various diseases due to the environmental condition and their living habits. So the prediction of disease at earlier stage becomes important task. But the accurate prediction on the basis of symptoms

becomes too difficult for doctor. The correct prediction of disease is the most challenging task. To overcome this problem data mining plays an important role to predict the disease. Medical science has large amount of data growth per year.

Due to increase amount of data growth in medical and healthcare field the accurate analysis on medical data which has been benefits from early patient care. With the help of disease data, data mining finds hidden pattern information in the huge amount of medical data. We proposed general disease prediction based on symptoms of the patient. For the disease prediction, we use K-Nearest Neighbor (KNN) and Convolutional neural network (CNN) machine learning algorithm for accurate prediction of disease.

For disease prediction required disease symptoms dataset. In this general disease prediction the living habits of person and checkup information consider for the accurate prediction. The accuracy of general disease prediction by using CNN is 84.5% which is more than KNN algorithm. And the time and the memory requirement is also more in KNN than CNN. After general disease prediction, this system able to gives the risk associated with general disease which is lower risk of general disease or higher.

[6] Lambodar Jena and Ramakrushna Swain, "ChronicDisease Risk Prediction using Distributed Machine Learning Classifiers" IEEE, 978-153862924-6/17, pp. 170-173, 2017.

The prime use of the classification technique is to predict the target class accurately for each case in the dataset. The recent study is focused on the usage of classification techniques in the field of medical science and bioinformatics. The main focus of this paper is to predict Chronic- Kidney-Disease and its usage for classification in the field of medical bioinformatics. It firstly classifies dataset and then determines which algorithm performs better for diagnosis and prediction of Chronic- Kidney-Disease.

Thus, the prime objective of this paper is to analyze the data from a chronic-kidney-disease (CKD) dataset using classification technique to predict class accurately in each case. Many researchers have compared the performance of

different classifiers applied on various datasets. But, none of the author worked on prediction of accuracy for chronic-kidney-disease dataset. Here, we have considered two critical classifiers to study their performance based on various parameters obtained by applying them in the dataset.

[7] Dhomse Kanchan B. and Mahale Kishor M., "Study of Machine Learning Algorithms for Special Disease Prediction using Principal of Component Analysis" IEEE, 978-1-5090-0467-6/16, pp. 5-10, 2016.

The worldwide study on causes of death due to heart disease/syndrome has been observed that it is the major cause of death. If recent trends are allowed to continue, 23.6 million people will die from heart disease in coming 2030. The healthcare industry collects large amounts of heart disease data which unfortunately are not "mined" to discover hidden information for effective decision making. In this paper, study of PCA has been done which finds the minimum number of attributes required to enhance the precision of various supervised machine learning algorithms.

The purpose of this research is to study supervised machine learning algorithms to predict heart disease. Data mining has number of important techniques like categorization, preprocessing. Diabetic is a life threatening disease which prevent in several urbanized as well as emergent countries like India. The data categorization is diabetic patients datasets which is developed by collecting data from hospital repository consists of 1865 instances with dissimilar attributes. The examples in the dataset are two categories of blood tests, urine tests. In this research paper we discuss a variety of algorithm approaches of data mining that have been utilized for diabetic disease prediction. Data mining is a well known practice used by health organizations for classification of diseases such as diabetes and cancer in bioinformatics research.

[8] Ankita Dewan and Meghna Sharma, "Prediction of Heart Disease Using a Hybrid Technique in Data Mining Classification" IEEE, 978-9- 3805-44168/15, pp. 704-706, 2015.

Heart disease prediction is treated as most complicated task in the field of medical sciences. Thus there arises a need to develop a decision support system for detecting heart disease of a patient. In this paper, we propose efficient genetic algorithm hybrid with the back propagation technique approach for heart disease prediction. Today medical field have come a long way to treat patients with various kind of diseases. Among the most threatening one is the Heart disease which cannot be observed with a naked eye and comes instantly when its limitations are reached. Bad clinical decisions would cause death of a patient which cannot be afforded by any hospital.

To achieve a correct and cost effective treatment computer-based and support Systems can be developed to make good decision. Many hospitals use hospital information systems to manage their healthcare or patient data. These systems produce huge amounts of data in the form of images, text, charts and numbers. Sadly, this data is rarely used to support the medical decision making. There is a bulk of hidden information in this data that is not yet explored which give rise to an important query of how to make useful information out of the data. So there is necessity of creating an excellent project which will help practitioners predict the heart disease before it occurs. The main objective of this paper is to develop a prototype which can determine and extract unknown knowledge (patterns and relations) related with heart disease from a past heart disease database record. It can solve complicated queries for detecting heart disease and thus assist medical practitioners to make smart clinical decisions which traditional decision support systems were not able to. By providing efficient treatments, it can help to reduce costs of treatment.

3. EXISTING METHOD

In the paper [1] the author described the whole procedure of developing a chatbot by dividing the process into segments such as speech-to-text conversion, natural language processing, response generation, knowledge base creation, dialogue management, text-to-speech, etc. The author has also included security

considerations such as security flaws in chatbot platforms and malicious chatbots. In the paper [3] the authors reviewed topics surrounding the chatbot's knowledge domain, response generation, text processing, machine learning model, and the dataset usage and evaluation strategy topics.

In the paper [4] the authors have described the building of a chatbot that can provide an authentic and accurate answer for any type of query using Artificial Intelligence Markup Language (AIML) and Latent Semantic Analysis (LSA) by using the application of the python platform. In the paper [5] the author covers different approaches to chatbot development, including key points on chatbot integration and deployment, and employs machine learning to configure a chatbot. In the paper [2] the author discussed the method of data analysis, which allows the analytical system to learn through the way of solving problems and applying similar methods in its working process. In the paper [6] the authors have analyzed how artificial intelligence and machine learning are implemented in popular use to make advancements in chatbot services specifically in helping users to access college websites.

The problem selects the closest matching response from the closest matching statement that matches the input and then chooses a response from an available selection of statements for that response. In the paper, [7] the author has elaborated on the use of diverse neural framework exhibits as the learning technique for setting up the chatbot to make it continuously like human enlistment authority. NLP techniques such as NLTK (python) can be applied for speech analysis and intelligent responses can be generated by designing a model to provide appropriate human-like responses. In the paper [8] the authors have discussed their research on how to design, develop and evaluate a health assistant chatbot application that helps users to ask any personal query related to healthcare without physically availing any hospital facilities.

In the paper by [9] the authors have described an AI chatbot whose work depends on Natural language processing. The chatbot users can upload their queries related to healthcare without physically availing of any healthcare facilities. It uses

Google API for voice-text and text-voice conversations. The query is sent to the chatbot and the related answer is displayed on an android app.

The system's main concern behind developing this web-based platform is analyzing customers' sentiments. In the paper [10] the authors have described a proposed idea to create a system with AI which meets the user's requirements. The AI can predict diseases based on symptoms and give a list of available treatments. The system can also give the composition of the medicines and their prescribed uses. In our chatbot model, we have incorporated the symptoms of Covid 19 since it is based on only this one disease currently. It also shows the list of medications and precautions that the users might take if they are infected.

In the paper [11] the authors have focused to show the implementation of a retrieval-based chatbot with voice support. They have investigated other standing chatbots and how it is useful in helping the patients to fetch all the necessary details about Covid 19. In the paper [12] the authors have described the modern chatbot functioning and incorporating institution-specific responses in chatbots related to Covid 19 related queries. The main necessity is for unique response mapping, complex contextualization, and dynamic validations which are led by human resources of content-led industry leaders to develop a chatbot through collaborative communication with companies that are experienced in machine learning and natural language processing.

In the paper [13] the authors have discussed the tests they conducted on 701 French participants. They found that interacting with their chatbot for a few minutes significantly increases people's intention to get vaccinated and have a positive attitude towards Covid 19 vaccination. The results suggest that a properly scripted and regularly updated chatbot could offer a powerful resource to help fight hesitancy toward COVID-19 vaccines. In the paper [14] the author has described how to identify chatbot use cases deployed for public health response activities during the Covid 19 pandemic. The authors filtered articles basing them on the abstracts and keywords in their texts and made their assessment.

Chatbots, their applications of usage, and chatbot design techniques were extracted from these articles. In the papers [12], [15] the authors have discussed his

study to review the current status of Covid 19 related chatbots in the healthcare sector, identify and categorize the upcoming and new technologies and their applications for Covid 19 and find solutions to related challenges.

In the paper [16] the authors have discussed their research where they took an interview study with 29 participants to study the daily positive and negative aspects that are experienced with CAs. By assessing how users presently think about CAs, the authors have identified one of the best criteria that could transform their future design of the model. This can contribute to the end user's perspective by evaluating these functionalities for existing research topics about the guidelines for efficient and seamless user experience for CAs.

In the paper [12] the author has outlined the creation of a Penn Medicine chatbot collaboratively created with Verily, Google Cloud, and Quantiphi, a Google Cloud strategic partner. The author has described how interactions with users that can be updated and transformed, such as checkers for disease symptoms, must be consistently made with the capacity, capabilities, and different types of pathways of the existing health system using it to communicate important information to patient actions required by the user while efficiently managing constrained and contained resources.

In the paper [17] the authors have proposed a conversational chatbot on Google Cloud Platform (GCP) to deliver telehealth in India to increase the user's access to knowledge related to healthcare and be able to levy the potentials of Artificial Intelligence to bridge the currently existing gap of demand and supply of human healthcare providers. In the paper [18] the authors have presented the design of a highly efficient Artificial Intelligence Chatbot for evaluation based on diagnostic technologies and recommended efficient and quick measures when patients are exposed to the deadly Covid-19. Along with this, creating a virtual assistant can also help in the measurement of the severity of the infection and connect with registered doctors when the given symptoms become serious.

In the paper [19] the author provides chatbot structures ALICE and Elizabeth, illustrating the speak information illustration and sample matching gadget of each. It discusses the problems, which can arise when Dialogue Diversity

Corpus is used to retain a functional chatbot system with examples of dialogues spoken by general human beings. A basic implementation of corpusbased chatbot training can be found when a Java program is used to convert from dialogue transcript to AIML.

In the paper [20] the authors have provided statistics concerning epidemiology, serological and molecular diagnosis, the starting place of SARSCoV2 and its capacity to contaminate human cells and protection issues. Then it focuses on the available therapies to combat Covid 19, the development of different vaccines, and the role of AI in managing the pandemic and limiting the spread of the virus. In the paper [21] the author discusses the challenges posed by Covid 19 to the education system. In the paper [22] the authors used Chabot technology to implement the medical consultant system service. It was implemented using information from the DoctorMe application's symptoms and treatment records. The test results demonstrate the proposed system's capability.

In the paper [23] the authors have introduced a sketch for a clinical chatbot that gives diagnoses and Treatments based totally on signs provided to the system. The device will be capable to measure the seriousness of the analysis and if needed, it will connect the user to a doctor available online. In the paper [24] the authors have brought to light the usefulness of chatbots in human resource management systems. They have illustrated a detailed analysis of chatbots in HRM, which are also known as HR-bots. It has been studied to emphasize its usefulness in real-time considering the different relevant challenges such as cost factors, complex business domains, limited responsiveness, etc.

In the paper [25] the authors have demonstrated their deep learning model, which is named the Long Short-Term Memory (LSTM) network-based patientdependent model that is adopted for FOG detection. In the paper [26] the authors discuss the sudden impact and severity of Covid 19 around the world and how to fight it by enabling the following possibilities: autonomous everything, pervasive knowledge, assistive technology, and rational decision support. In the

paper [27] the authors have discussed the essential roles of some AI-driven techniques (machine learning, deep learning, etc.) and AI-empowered imaging techniques to analyze, predict, and diagnose COVID-19 disease.

In the paper [28] the authors have demonstrated the various machine learning models which have been built to predict the PPIs between the virus and human proteins that are further validated using biological experiments. A special chatbot with the ability of visual question answering with the integration of scenetext using PHOCs and fisher vectors is introduced in the paper [29], [30], [31].

The paper [32] discusses on the impact of algorithmic information processing have on users' attitudes and actions while using artificial intelligence (AI). A paper [33] creates a cognitive model to describe user interactions with conversational journalism (CJ) in the setting of chatbot news using the anthropomorphism and explainability constructs. In a study, an AI-based machine learning model was developed to forecast the effects of interactions between Paget's disease treatment and pharmaceuticals used to treat osteoporosis. This model reduces the cost and time required to apply the most effective medication combination in medical practice [34]. A deep learning model that could locate FMN interacting residues using a 2D convolutional neural network and positionspecific score matrices [35].

4.PROPOSED METHOD

4.1 BACKGROUND LSTM

Long-Short Term Memory (LSTM) is an artificial neural network famous for its applications in Artificial Intelligence and deep learning. Using its four main gates, it is easy to solve complex problems in the fields of machine translation, speech recognition, input-output mapping, and neural networks. When it comes to learning specific patterns, LSTMs perform better than other types of neural networks. It is a sort of RNN (Recurrent Neural Network) that is commonly used to

learn sequential data and mapping issues. As far as the LSTM gates are concerned, they fall under four main categories - forget gate, input gate, output gate, and cell gate. The purpose of every gate is to perform a specific function that has to be achieved.

- Forget gate: It is answerable for deciding that info is unbroken for scheming the cell state and that isn't relevant and might be discarded. The information from the previous hidden state or previous cell can be stated as ht-1 and the information from the current cell can be stated as xt . Two inputs are given in the forget gate.

- Input gate: Input Gate comes in available in updating the cell state and decides which records are necessary and which are not. It helps to discard the data, and the input gate helps to discover necessary information and store certain data in the relevant memory. ht-1 and xt are the inputs that are each passed via sigmoid and tanh functions respectively. 'tanh' regulates the network.

- Output gate: The output gate decides what the next hidden state should be. It is the last gate. ht-1 and xt are exceeded to a sigmoid function. The most current modified state is passed via the tanh function and is multiplied with the sigmoid output to determine the information of hidden state has to carry.

- Cell gate: First, all knowledge gained is accustomed to calculating new cell states. Firstly, it increased with the output. This has a chance of dropping values within this state if it is increased by close values of zero. Our chatbot's interface is made with the Tkinter library.

For loading the library and making it a perfect layout for the bot, it takes O(n) time complexity. However, for the training model and to predict the correct reply to the user's question, the algorithm's running time cost is O(n), and as we know for making the training model and putting different layers on it to make a

perfect neural network, the time complexity is O(4h(3d+h+d)) where the neural network layers d, h and it is the proposed time complexity for the LSTM algorithm.

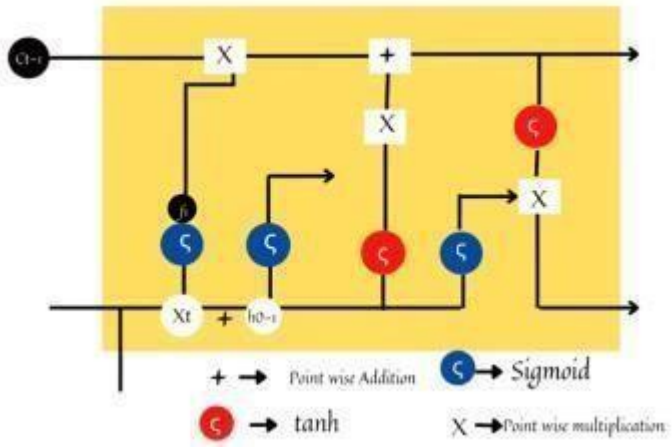


FIGURE 1. Functioning of Forget gate in LSTM algorithm.

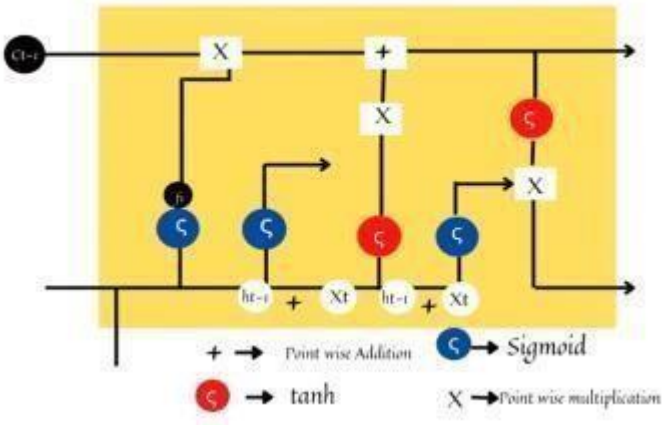


FIGURE 2. Functioning of the input gate in LSTM algorithm.

A. WORKING WITH LSTM

LSTM functions in the same manner as we have discussed in the given figures. The inputs given to our model are stored in the memory of the neural network and during the training of the model, it cross-checks the present information with all the accumulated ones and then finally provides the required output. The algorithm has acted as a good classifier for the model inputs as it filters data from previous timestamps and sends the refurbished data to its memory. The collected data is sent to the output gate after it has passed through the forget gate and the prediction for

the next output gets ready in time. This helps in the precision of the model and reduces the space complexity.

Fig.1 to Fig4 describe the functionality of components of LSTM method. In Fig 5, the graph demonstrates the trend that we can see during the testing phase of our model. The model is trained with different types of input values. When a test set is given as an input, it predicts the upcoming inputs and shows the output after analyzing them with the recent past values. As the graph shows, the predicted values according to the corresponding test set are quite accurate, and the model can maintain consistency in its outputs with sensational precision.

B. RECURRENT NEURAL NETWORK

RNN stands for Recurrent Neural Network. RNN contains internal memory, which makes it one of a kind since there is

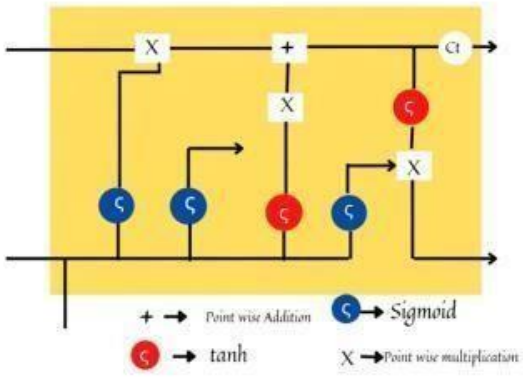


FIGURE 3. Functioning of output gate in LSTM algorithm

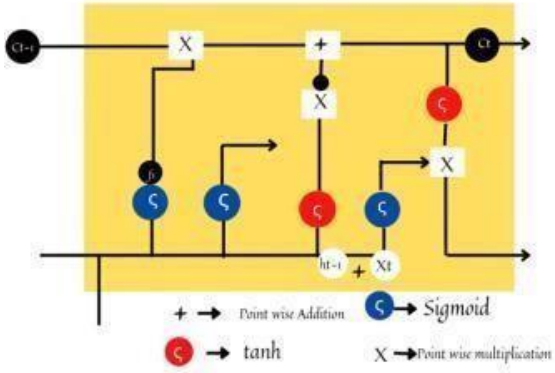


FIGURE 4. Functioning of the cell gate in the LSTM algorithm

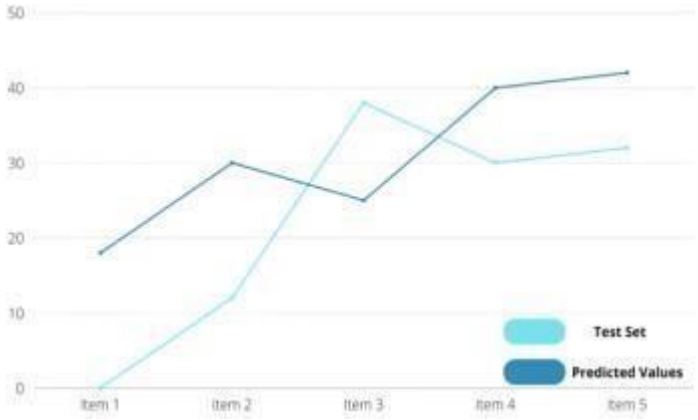


FIGURE 5. Graph showing test set versus predicted values using LSTM in our model.

no other neural network possessing such functionality. RNN is a robust and one of the most promising algorithms. Due to the presence of internal memory, RNNs can remember and reconstruct the inputs given to them, and they can predict the upcoming inputs with great precision and efficiency. The use of RNNs in popular usage has been described by Lex Fridman as "Whenever there is a sequence of data and that the temporal dynamics that connect the data is more important than the spatial content of each frame." The working of RNN resembles that of a human brain.

They use their predictive output with seamless precision and provide the exact required information. To simply state the working of RNN, we can say that recurrent neural networks. It works on two inputs, one is the recent past, and another is the present. It is the main aspect because the sequence of data contains details that are crucial to what will be coming up next. In our model, the RNN algorithm comes with this uniqueness since it analyzes the recent past and present inputs according to the queries given by the user. Every time, the accuracy of the model maintains the saturation level as the correct information is given as output and the model can detect what information the user is likely to require next. the time complexity at O(n), and the accuracy reaches as high as 9.5 approximately. LSTM comes in handy in maintaining a steady gradient with all the relevant and recurring values held together, and it maintains the accuracy level at a constant.

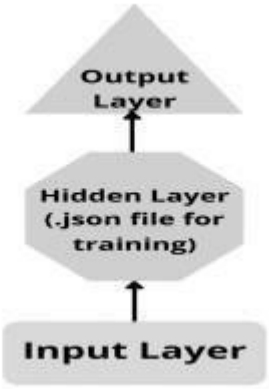


FIGURE 7. Demonstration of the RNN algorithm as applied to our chatbot model

In Fig 7, the demonstration of RNN is shown as it is applied to our chatbot model. During working with several inputs, there might exist a deeper network with a certain input layer, while others might have more than one hidden layer. All of these are trained together inside the .json file and are trained in such a way that each hidden layer has its own set of weights and biases. This gives independence to each layer, so, it does not have to remember all the inputs. This increases the efficiency of the model and the accuracy of the output given will have significant growth.

C. **DECISION TREE MODEL**

The decision tree is used for categorization and regression problems, but it is mainly used to solve classification problems. Where internal nodes show the dataset features, branch shows the rules of decision, and leaf node shows the outcome. There are two nodes present in the decision tree. They are the Leaf node and the Decision node. Decision nodes decide how the algorithm works and Leaf nodes show the output of that decision and are connected with Decision nodes. We can say that the decision tree is the graphical representation of all the solutions for all the problems
under given conditions.

It looks like a tree structure. It has a root node, all the branches connected with it, and leaf nodes connected with branches. We used the CART algorithm to make the Decision tree. Decision trees commonly impersonate human wondering capacity

while making decisions. From the very beginning, we have to choose the root node that fully contains the datasets. Then we can locate the satisfactory attribute dataset with the use of Attribute Selection Measurement (ASM). Next, we have to split the best attribute dataset, which contains possible values for the best attribute dataset. We have to generate the decision tree node and finally, we have to search the best dataset for a solution until we reach the final possible solution.

We used the following two mathematical equations to solve the determinant of the decision tree. Information Gain = Entropy(S) − [(Weighted Average) ∗ Entropy (each feature)] Gini Index = 1 − 6jP 2 j (1) From the above equation we get 0.6753 accuracies and Time Complexity of O(nkd). In Fig 8, a demonstration of the decision tree model as it is applied to our chatbot model. Suppose any user of our chatbot, is unaware about the Covid-19 and wants to decide whether he needs to know about the symptoms, doctors, or the oxygen help for the patient. For solving the problem, the decision tree begins with the root node (Corona Virus by ASM).

The root node splits similarly into the subsequent decision node (Symptoms, Doctor, Oxygen, Emergency Bed) and different leaf nodes primarily based on the corresponding labels. The subsequent decision node is in addition split into three decision nodes, and these are the ultimate leaf nodes of the previous decision node. Finally, the machine learning training module finds the final leaf node (Helpers) However, the user of the chatbot wants to know about the oxygen helpers. From the upper diagram, we can see how the Decision Tree Model solves a problem.

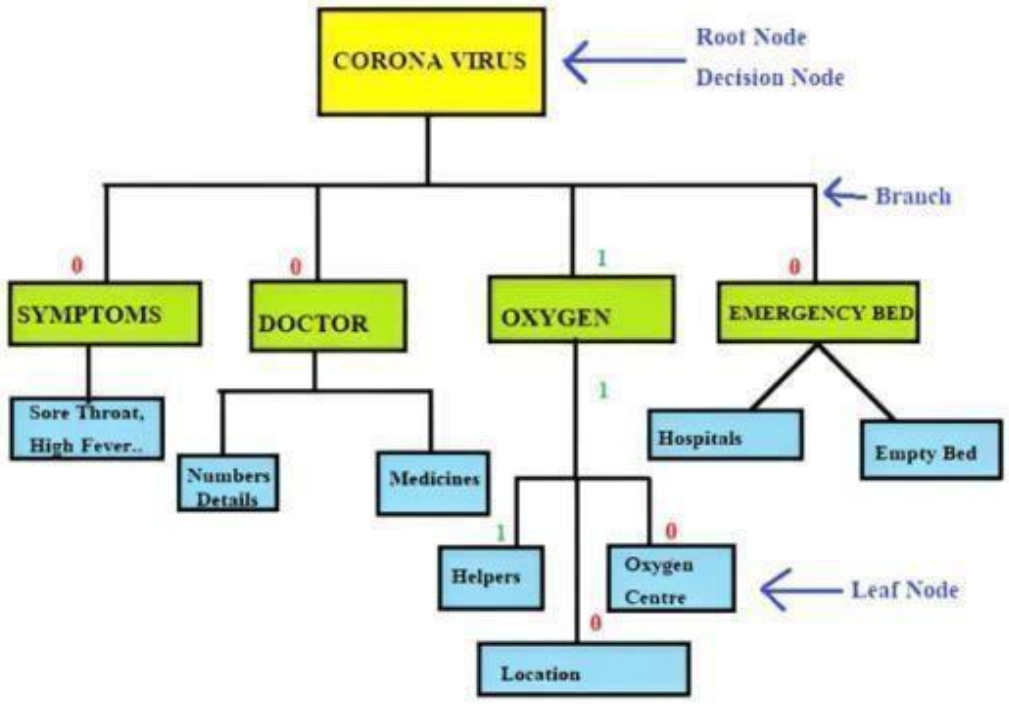


FIGURE 8. Demonstration of Decision Tree Model as it is applied to our chatbot model.

4.2 PROPOSED MODEL

Whenever the user writes anything to the interface of the bot, the bot will reply and answer the corresponding questions or someone greets people using ‘‘good morning’’ or ‘‘good evening’’, the bot will also greet them by saying ‘‘good morning, it’s a nice day’’. Even if someone asks for the time, the bot eventually tells the user what time it is. Moreover, if the user asks some different queries to the bot, the bot also predicts the most accurate queries for it. For example, if the user expresses his/her emotions that are not in the training model, by saying, ‘‘it’s a bad day to me’’ the bot eventually replays by predicting the near, answers, ‘‘I am here to help you, tell me your problem ‘‘. In this chatbot, ‘‘voice recognition’’ is also implemented as another feature.

If the user wants to say something in voice instead of texting, the bot also replies to it in voice and in text, both modes. Here, before making the bot ready, we have to train our bot using ‘‘tags’’ and ‘‘responses’’. As an example, in the tag section, if we write ‘‘greetings’’ and in the response section, we write ‘‘good

morning'', or ''it is a nice day'', the bot eventually replies whenever it finds something closer to its response. With the machine-learning concept and using NLP and TensorFlow, we make predictions for the bot what will be accurate answers after training it.

Here, the training model will be the main part, because in the training model all the training like what type of questions or queries the bot should be answered is implemented. The TensorFlow helps to build the NLP for chatbots and utilizes deep neural network architecture after building the network for our chatbot. The bot will predict the correct answers to users' queries. Even if it is not in the training model, the bot also tries to predict it closer by checking the sentences and its word, which will be closer to the training model's response.

In Fig.4, we can see that a user can freely decide what its mode will be. If the user chooses text mode, just to write its query to the chatbot interface, the bot will make a prediction using the training model, then reply and give a perfect reply to it. If the user chooses voice mode, the voice recognition module will recognize the voice and convert it to the text. Then the bot will do the same thing just for text mode and, in the bot, will reply to the query and give a speech about it. In the end, if the user wants to continue, they can continue and if the user doesn't want to continue, can click on the exit button, then the chatbot will turn off.

In the future, whenever a user wants to use it, they can use it. In the case of predicting the user's text and giving an accurate reply by the bot, the following algorithm will represent how it is working, queries. The bot will detect which type of query it is and predict the most accurate answers for it. Fig. 9 shows the block diagram or process flow diagram of our proposed chatbot system. In Alogirthm-1, we have to make a json file at first which will contain tags, question patterns, and its response. Then we have to load the json file into the training file. After that, we have to make a list and also make a list that will ignore unnecessary words.

Then whatever words are stored in the list, we have to ''lemmatize'' all of the stored words and then again store them in a new variable, ''words'' in the form of sorted. Then we have to open a file in binary mode, ''words.pk1'' and make a

dump file of it. Again, we have to ‘‘lemmatize’’ the patterns and store the corresponding output in the training list. Then shuffle all elements present in the training list. Then store one part of the training list in train_x variables and another part store in train_y variables. After that, we have to make a neural network by adding Dense and Dropout. Then, compile it and save it as ‘‘chatbot_model.h5’’.

After that whenever a user writes something on chatbot interfaces, the bot will predict the closer reply by using the training model. Even if the user wants to go on voice mode, they will hear the reply from the bot even though they can command the bot through their voice, and the bot will convert the voice to text and do the same work as it is done in text mode. In the case of making the chatbot interface. Here the Tkinter library is used. Tkinter gives a good GUI. Where the chitchat between the user and the bot is done in the bot’s interface, which is made in python using the Tkinter library.

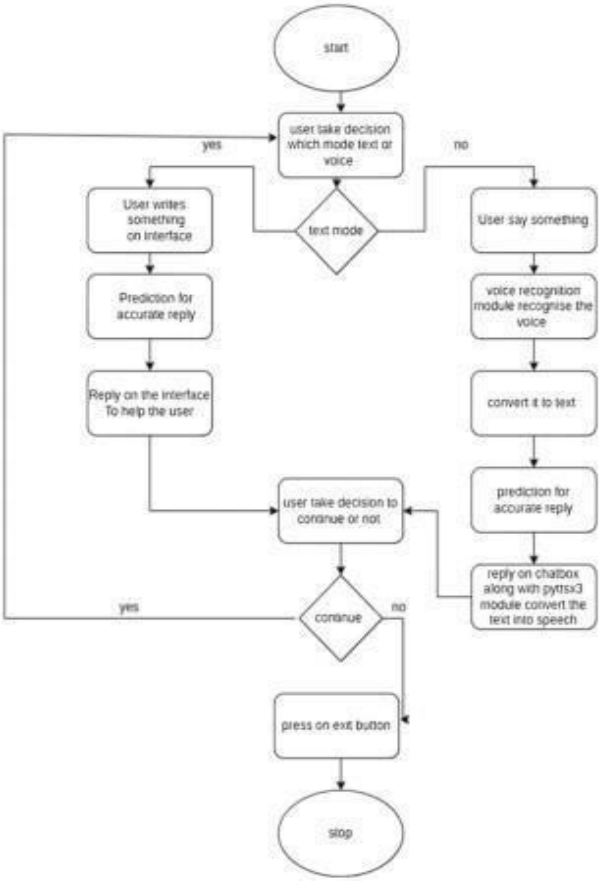


FIGURE 9. Flowchart of the overall chatbot system.

5.RESULT

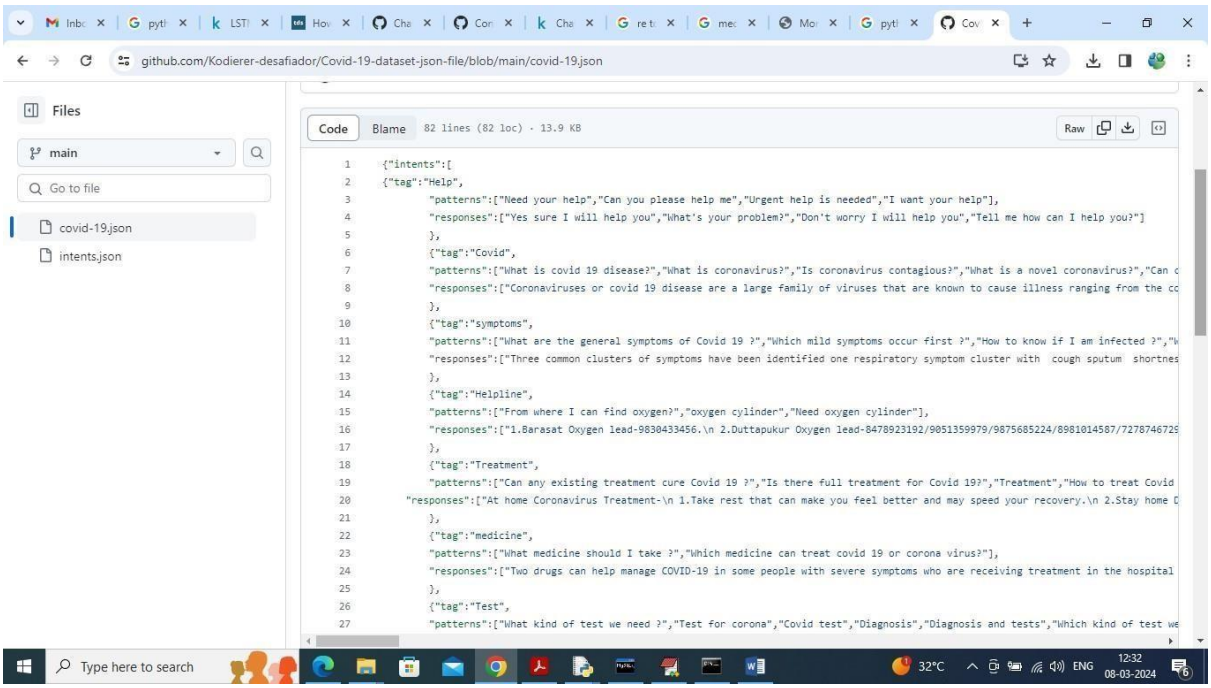
An AI-Based Medical Chatbot Model for Infectious Disease Prediction

In this paper author utilized LSTM algorithm to train a model which will accept question as input from the user and then predict closed matched answer and then display to user as Chatbot reply.

5.1 DATASETS

To train LSTM algorithm author has given some medical question dataset which can be download from below URL https://github.com/Kodierer-desafiador/Covid-19-dataset-jsonfile/blob/main/covid-19.json

Above dataset contains some questions showing in below screen



So trained LSTM Chatbot can reply for any question available in above dataset screen and I saved this dataset inside 'Dataset' folder.

As per your request we have made this application work for both Text and Voice based Chatbot. Chatbot reply to use in both English and Telugu and for translation we have used Google translation which will work for few questions as this free based API. If application stuck then you can consider translator not working so you

can start after some time. You can run for five queries at a time and may work for more queries also but some time it may not work.

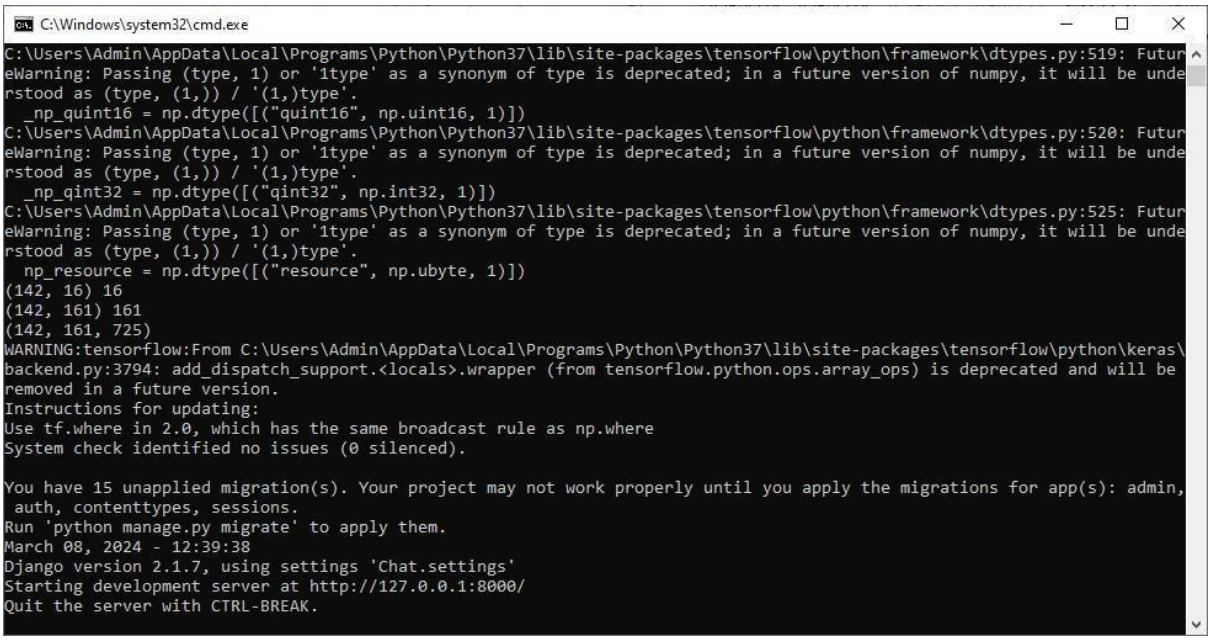
To implement this project we have designed following modules

1) Sign up: user can sign up with the application
2) User Login: after sign up can login to application
3) Train LSTM Algorithm: after login you can run this module to train and load LSTM algorithm and then calculate training accuracy and graph
4) Interact with Voice Based Chatbot: using this module u can interact with Chatbot in voice based mode
5) Text Based Chatbot: can interact with Chatbot in text mode
6) View History: can view all chats of history

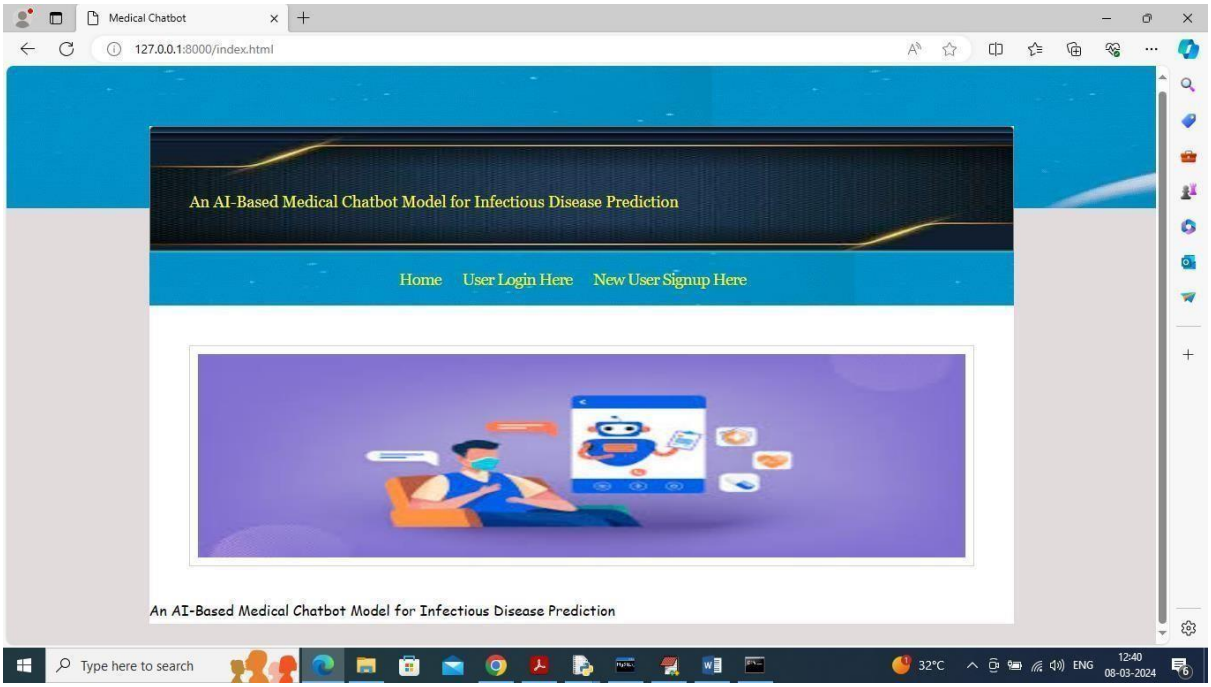
5.2 RESULTS & TEST ANALYSIS

SCREEN SHOTS

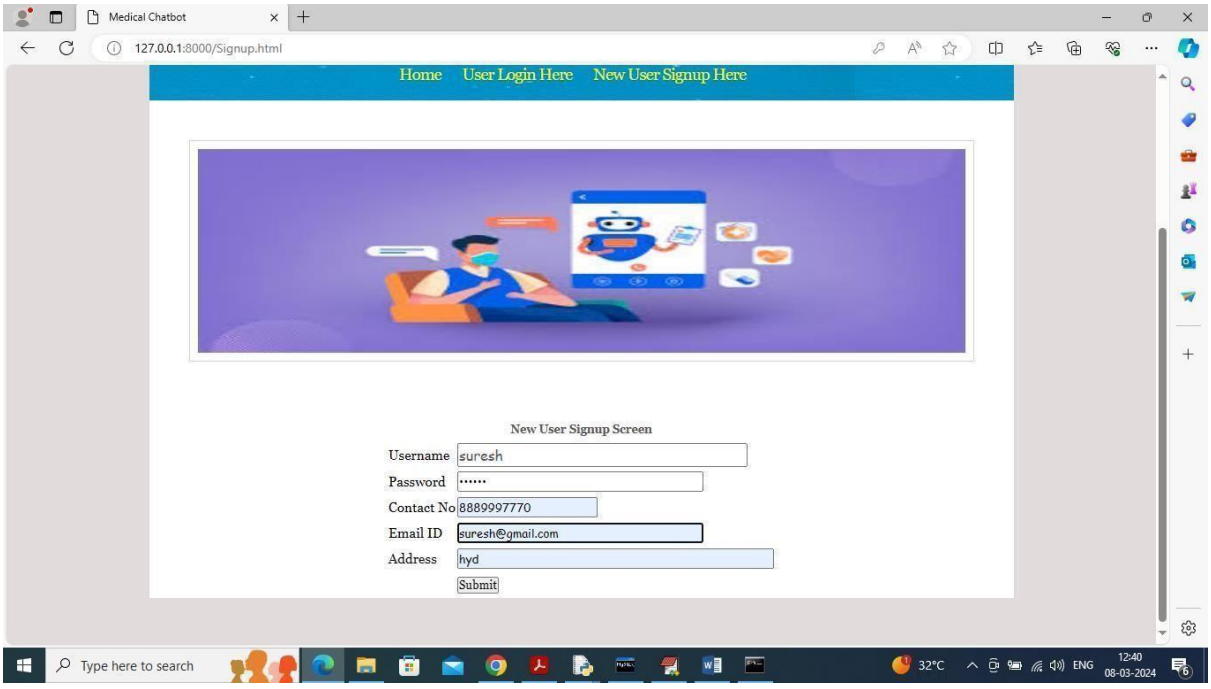
To run project install python 3.7 and then install all packages given in requirement.txt file and then install MYSQL and then copy content from DB.txt file and paste in MYSQL console to create database. Now double click on run.bat file to start python web server and get below page



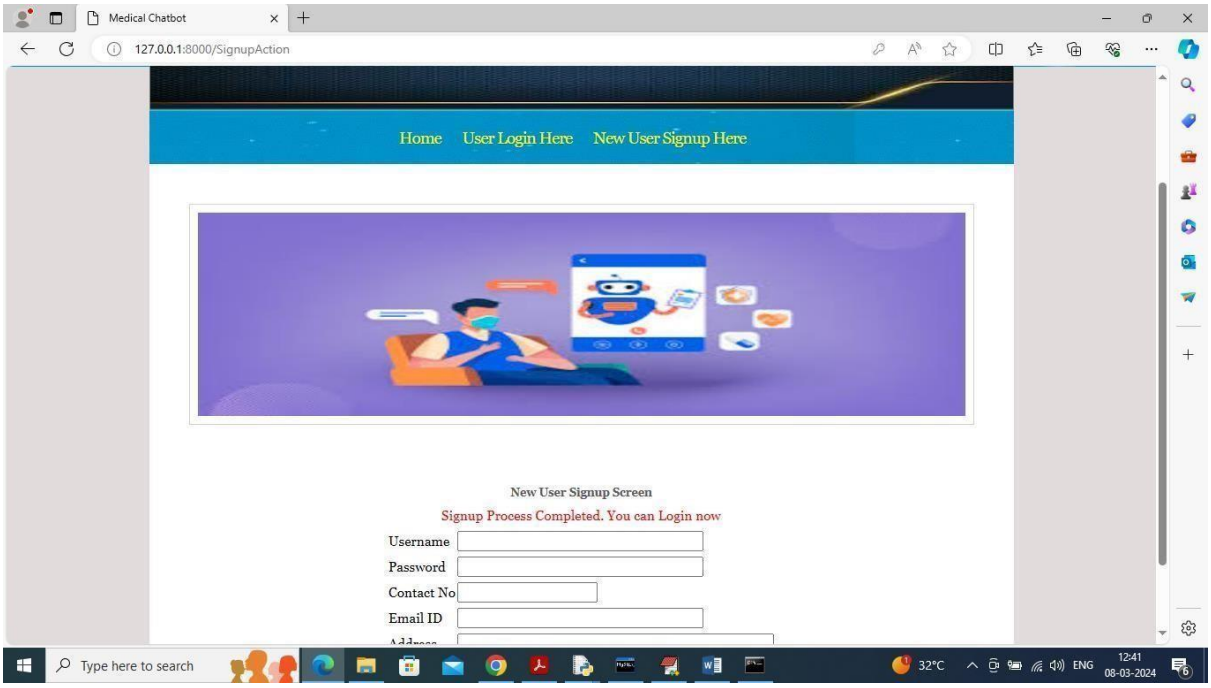
In above screen python server started and now open browser and enter URL as http://127.0.0.1:8000/index.html and press enter key to get below page



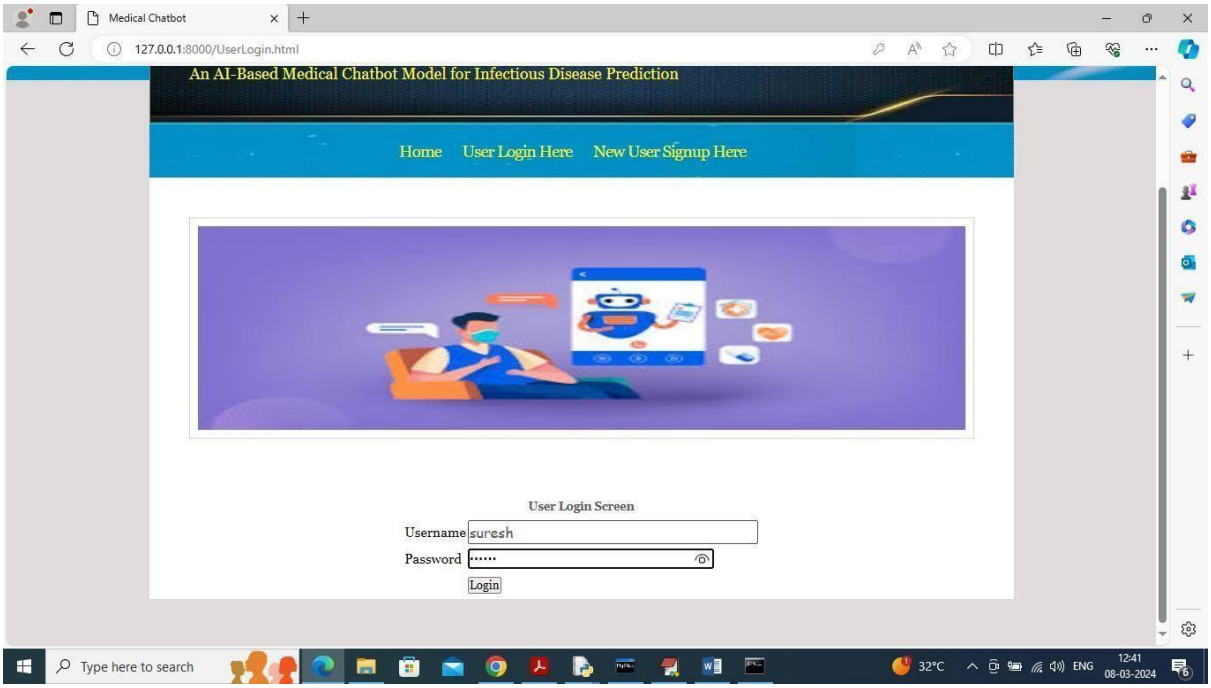
In above screen click on 'User Sign up' link to get below page



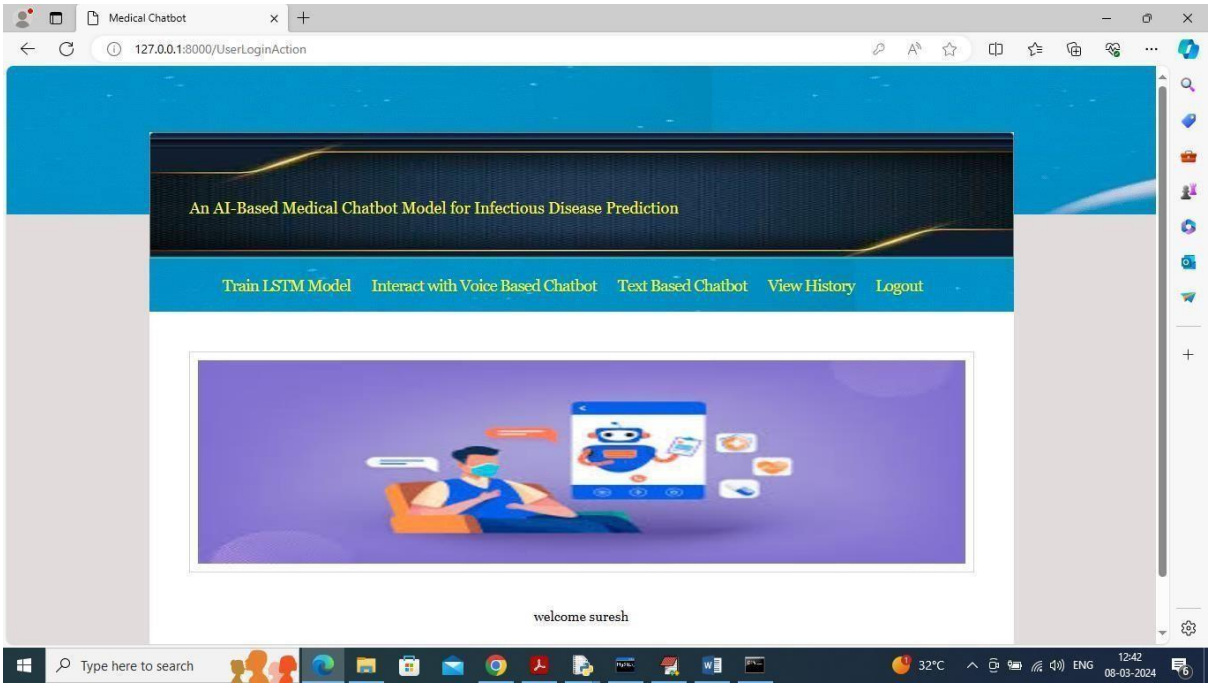
In above screen user is entering sign up details and then press button to get below page



In above screen user sign up completed and now click on 'User Login' link to get below page



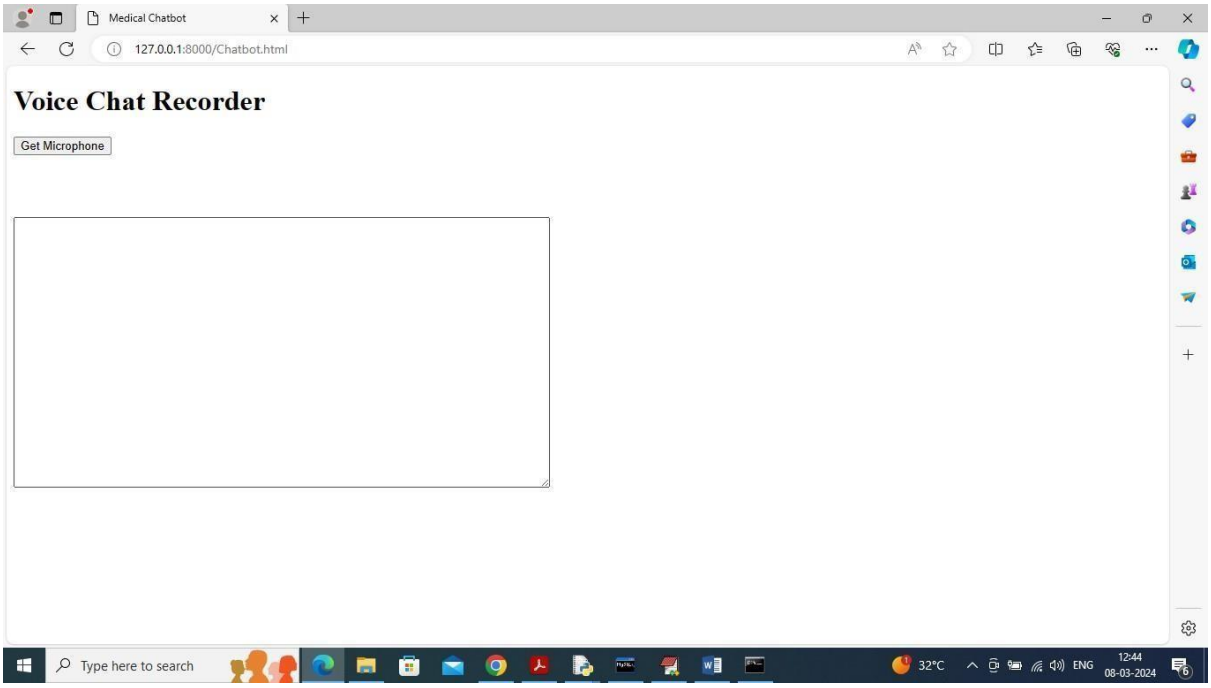
In above screen user is login and after login will get below page



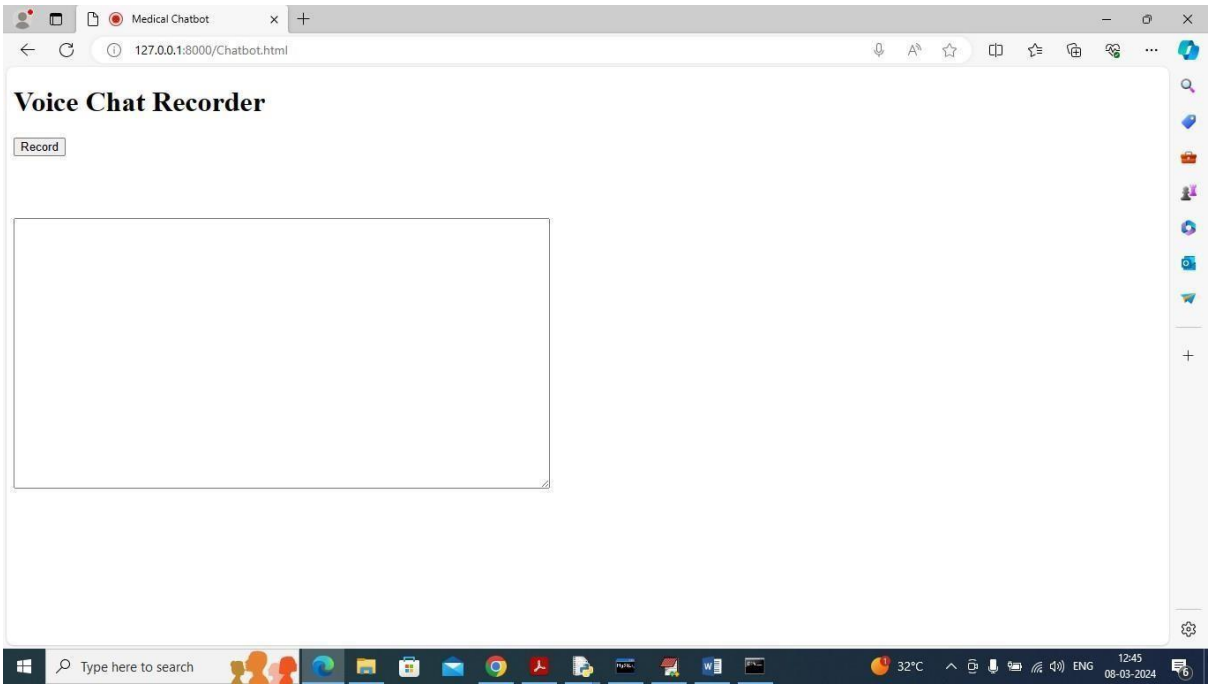
In above screen user can click on 'Train LSTM Model' link to get below page

In above screen LSTM training completed and in blue colour text can see LSTM accuracy is 99% and in graph x-axis represents training EPOCHS and y-axis represents Accuracy/LOSS values and then green line represents Accuracy and red line represents LOSS and can see with each increasing epoch accuracy got increase and reached closer to 1 and loss got decrease. Now click on 'Interact with

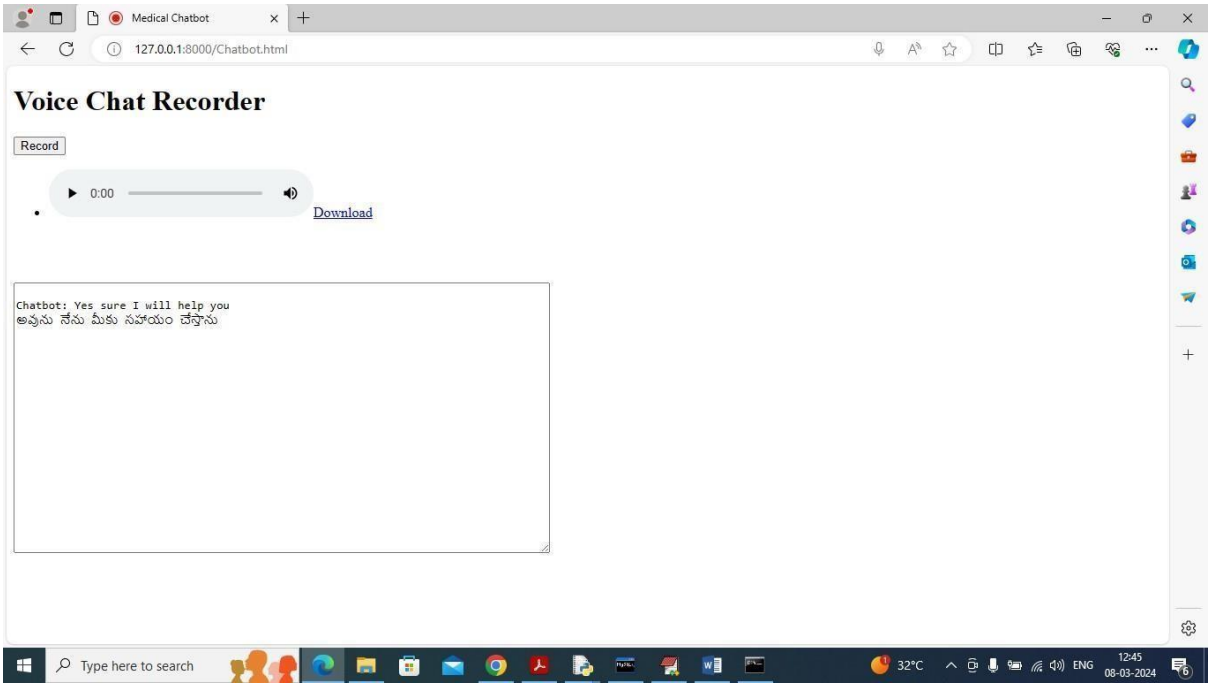
Voice Chatbot' link to get below voice recorder



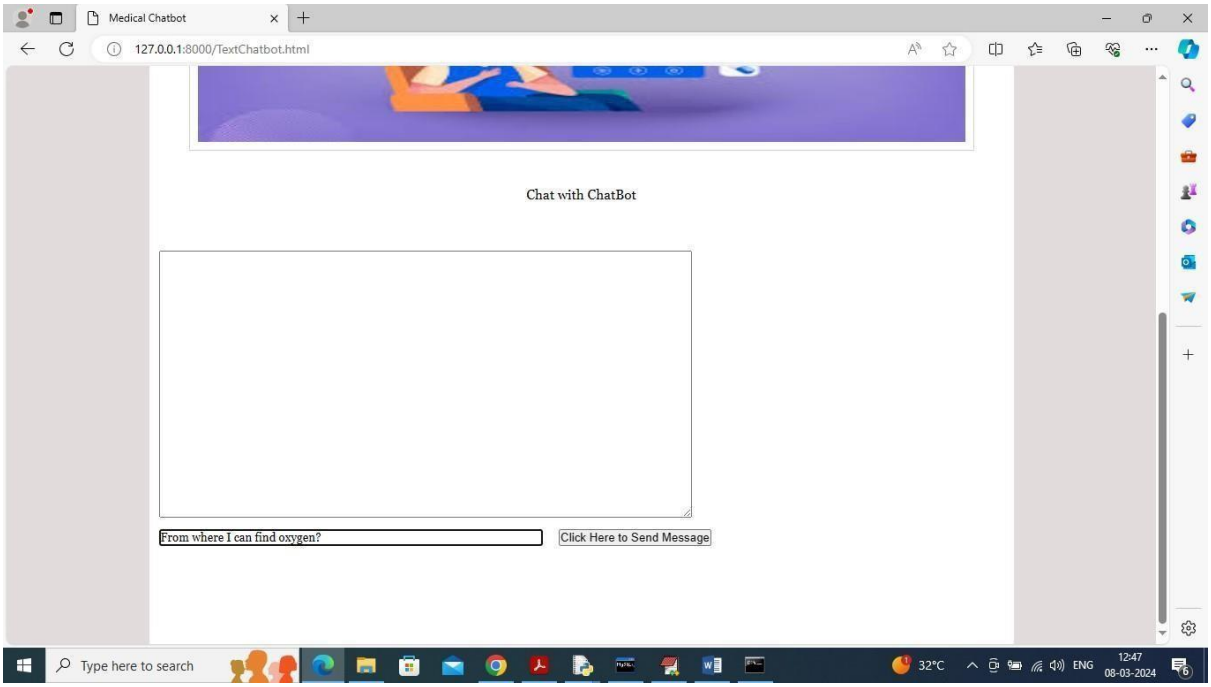
In above screen click on 'Get Microphone' link to connect to micro phone and get below page



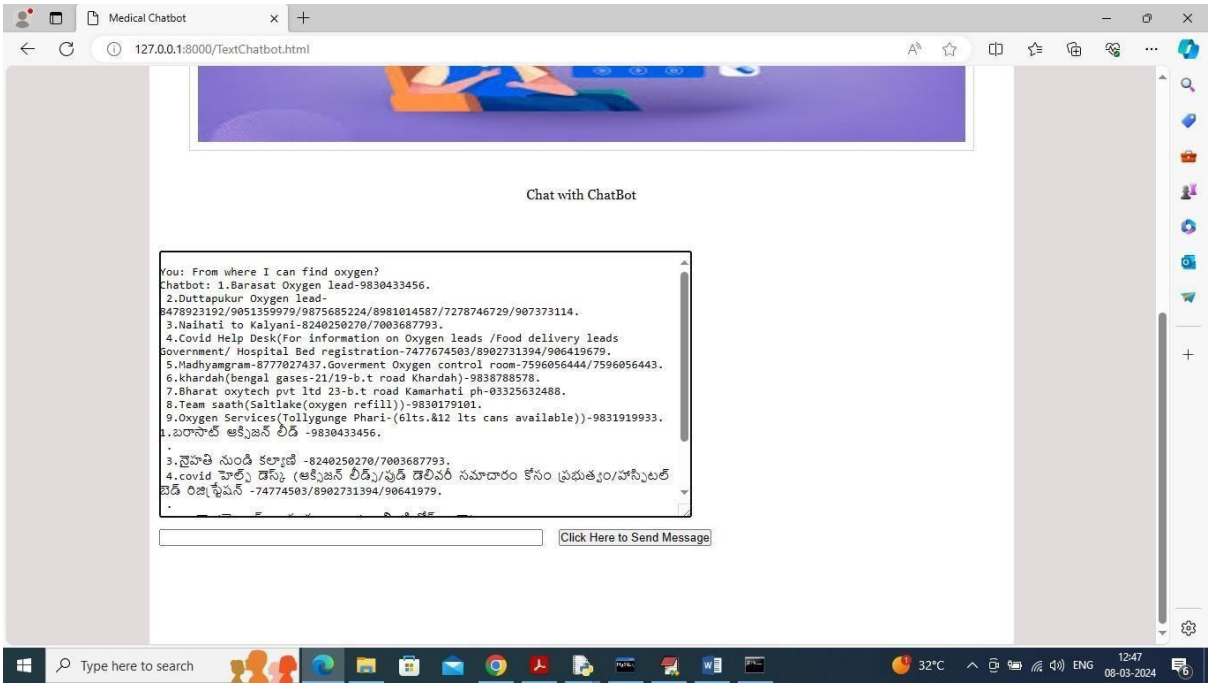
In above screen click on 'Record' button and start speaking and once done click 'Stop' button to get reply from Chatbot



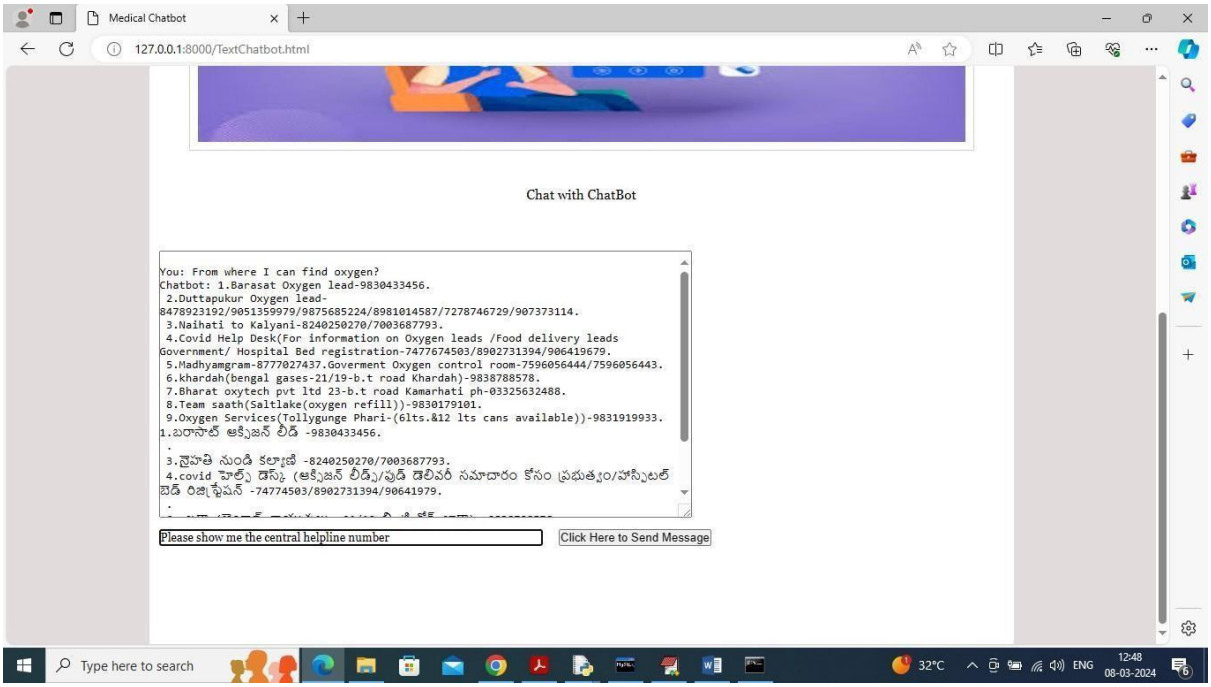
In above screen I spoke word as ‘Need Your Help’ and then got reply from Chatbot in both English and Telugu and similarly you can record and get output from Chatbot and now click on ‘Text Based Chatbot’ to get below page



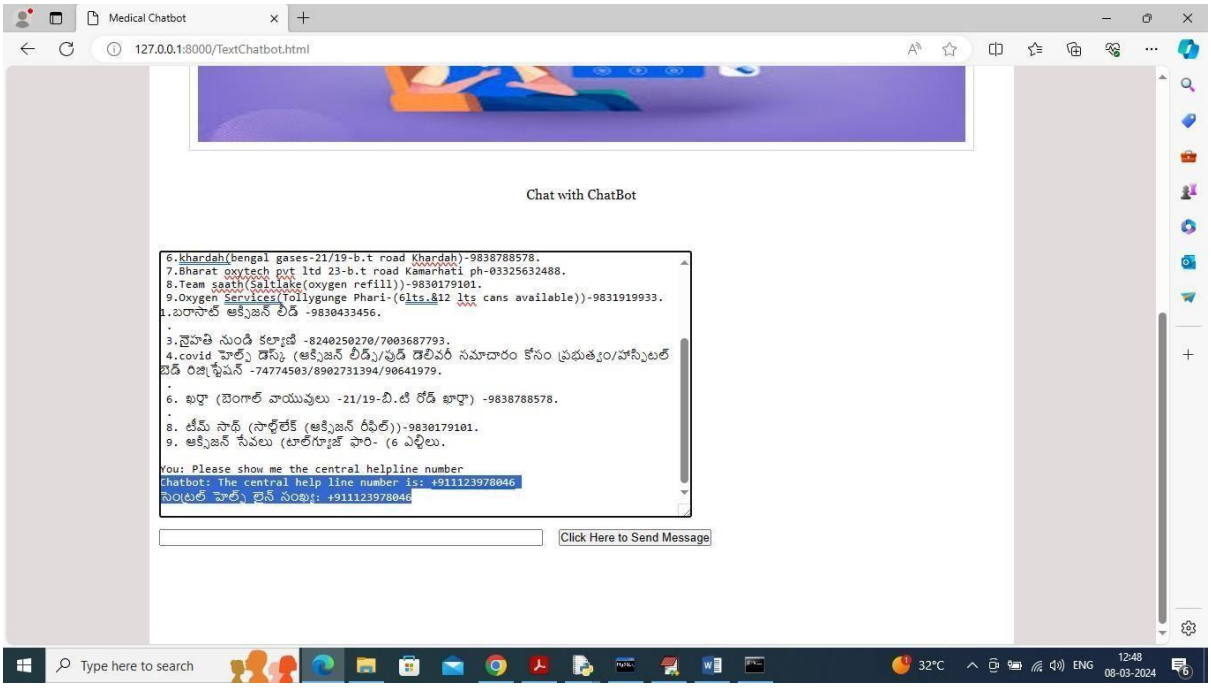
In above screen I asked question about ‘Oxygen Cylinder’ and press button to get below page



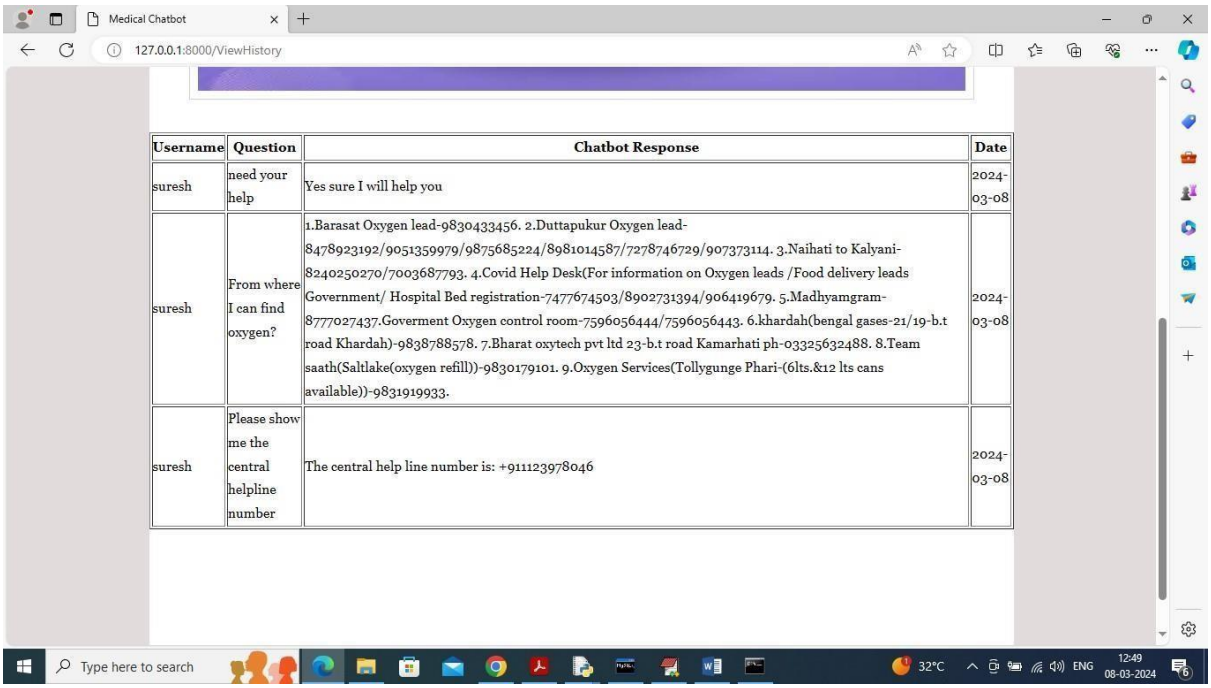
In above screen got reply from Chatbot in both English and Telugu and below is another question



In above screen asking for ‘covid help line number’ and below is the response



In above screen can see response for help line in both Telugu and English and now click on 'View History' link to get below page



Username	Question	Chatbot Response	Date
suresh	need your help	Yes sure I will help you	2024-03-08
suresh	From where I can find oxygen?	1.Barasat Oxygen lead-9830433456. 2.Duttapukur Oxygen lead-8478923192/9051359979/9875685224/8981014587/7278746729/907373114. 3.Naihati to Kalyani-8240250270/7003687793. 4.Covid Help Desk(For information on Oxygen leads /Food delivery leads Government/ Hospital Bed registration-7477674503/8902731394/906419679. 5.Madhyamgram-8777027437.Goverment Oxygen control room-7596056444/7596056443. 6.khardah(bengal gases-21/19-b.t road Khardah)-9838788578. 7.Bharat oxytech pvt ltd 23-b.t road Kamarhati ph-03325632488. 8.Team saath(Saltlake(oxygen refill))-9830179101. 9.Oxygen Services(Tollygunge Phari-(6lts.&12 lts cans available))-9831919933.	2024-03-08
suresh	Please show me the central helpline number	The central help line number is: +911123978046	2024-03-08

In above screen user can view all question he asked and the response from the Chatbot.

Similarly by following above screens you can run Medical Chatbot in both voice and text format .

6.CONCLUSION

The AI-based medical chatbot project described in the provided sources focuses on developing a sophisticated chatbot model for infectious disease prediction, particularly during the Covid-19 pandemic. The project incorporates advanced technologies like chatbots, language translation modules, speech recognition, and the Django web framework to enhance user interaction and accessibility. By utilizing neural network architectures, LSTM models, and decision tree algorithms, the chatbot demonstrates high accuracy in predicting diseases, providing treatment information, and offering personalized healthcare guidance. The system's ability to track user history, integrate multilingual support, and enable voice-based interaction showcases its potential to revolutionize healthcare accessibility and disease management. Through rigorous testing and continuous improvement, the project aims to provide a user-friendly, efficient, and personalized healthcare solution for a diverse user base.

FUTURE SCOPE

Based on the information provided in the search results, here are some potential future scope and improvements for the AI-based medical chatbot project:

1. Expanded Disease Coverage:The current chatbot model is focused on COVID-19, but the future scope could involve expanding the disease coverage to include a wider range of infectious diseases. This would allow the chatbot to provide comprehensive medical guidance and support for users dealing with various health concerns.

2. Personalized Recommendations: ● The user history tracking feature using MySQL can be further leveraged to provide more personalized recommendations and guidance to users over time. By analyzing the user's past interactions and medical history, the chatbot could offer tailored advice, treatment suggestions, and follow-up reminders.

REFERENCES

[1] M. Jiang, Y. Chen, M. Liu, S. T. Rosenbloom, S. Mani, J. C. Denny, and H. Xu, "A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries," J. Am Med Inform Assoc, vol. 18, no. 5, pp. 601–606, 2011.

[2] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang,"Disease prediction by machine learning over big data from healthcare communities" IEEE Access, vol. 5, no.1, pp.8869–8879, 2017.

[3] Sayali Ambekar, Rashmi Phalnikar, "Disease RiskPrediction by Using Convolutional Neural Network" IEEE, 978-1-5386-5257-2/18, 2018.

[4] Naganna Chetty, Kunwar Singh Vaisla and Nagamma Patil, "An Improved Method for Disease Predictionusing Fuzzy Approach" IEEE, DOI 10.1109/ICACCE.2015.67, pp. 569-572, 2015.

[5] Dhiraj Dahiwade, Gajanan Patle and Ektaa Meshram, "Designing Disease Prediction Model Using Machine Learning Approach" IEEE Xplore Part Number: CFP19K25-ART; ISBN: 978-1-5386-7808-4, pp. 1211-1215, 2019.

[6] Lambodar Jena and Ramakrushna Swain, "ChronicDisease Risk Prediction using Distributed Machine Learning Classifiers" IEEE, 978- 1-5386-2924-6/17, pp. 170-173, 2017.

[7] Dhomse Kanchan B. and Mahale Kishor M., "Study of Machine Learning Algorithms for Special Disease Prediction using Principal of Component Analysis" IEEE, 978-1-5090-0467-6/16, pp. 5-10, 2016.

[8] Ankita Dewan and Meghna Sharma, "Prediction of Heart Disease Using a Hybrid Technique in Data Mining Classification" IEEE, 978-9- 3805-44168/15, pp. 704-706, 2015.

APPENDIX

PYTHON

1.1 Introduction

* One of the most popular languages is Python. Guido van Rossum released this language in 1991. Python is available on the Mac, Windows, and Raspberry Pi operating systems. The syntax of Python is simple and identical to that of English. When compared to Python, it was seen that the other language requires a few extra lines.

*It is an interpreter-based language because code may be run line by line after it has been written. This implies that rapid prototyping is possible across all platforms.

Python is a big language with a free, binary-distributed interpreter standard library.

* It is inferior to maintenance that is conducted and is straightforward to learn. It is an object-oriented, interpreted programming language. It supports several different programming paradigms in addition to object-oriented programming, including functional and procedural programming.

* It supports several different programming paradigms in addition to objectoriented programming, including practical and procedural programming. Python is mighty while maintaining a relatively straightforward syntax. Classes, highly dynamic data types, modules, and exceptions are covered. Python can also be utilised by programmes that require programmable interfaces as an external language.

Here are some key features and characteristics of Python:

- Readability: Python emphasizes code readability with its clean and intuitive syntax. It uses indentation and whitespace to structure code blocks, making it easy to understand and maintain.

- Easy to Learn: Python's simplicity and readability make it an excellent choice for beginners. Its straightforward syntax and extensive documentation make it accessible for newcomers to programming.
- Interpreted Language: Python is an interpreted language, meaning that it doesn't need to be compiled before running. The Python interpreter reads and executes the code directly, making the development process faster and more interactive.
- Cross-platform Compatibility: Python is available for major operating systems like Windows, macOS, and Linux. This cross-platform compatibility allows developers to write code once and run it on different platforms without modifications.
- Large Standard Library: Python comes with a vast standard library that provides ready-to-use modules and functions for various tasks. It covers areas such as file I/O, networking, regular expressions, databases, and more, saving developers time and effort.
- Extensible and Modular: Python supports modular programming, enabling developers to organize code into reusable modules and packages. Additionally, Python allows integrating modules written in other languages, such as C or C++, providing flexibility and performance optimizations.
- Wide Range of Libraries and Frameworks: Python has a vibrant ecosystem with numerous third-party libraries and frameworks. These libraries, such as NumPy, pandas, TensorFlow, and Django, extend Python's capabilities for specific domains, making it a powerful tool for diverse applications.
- Object-Oriented: Python supports object-oriented programming (OOP) principles, allowing developers to create and work with classes and objects. OOP provides a structured approach to code organization, promoting code reuse and modularity.

- Dynamic Typing: Python is dynamically typed, meaning variable types are determined at runtime. Developers do not need to declare variable types explicitly, which enhances flexibility and simplifies code writing.

1.2 Installation

To install Python on your computer, follow these basic steps:

- Step 1: Visit the Python website Go to the official Python website at https://www.python.org/.
- Step 2: Select the operating system Choose the appropriate installer for your operating system. Python supports Windows, macOS, and various Linux distributions. Make sure to select the correct version that matches your operating system.
- Step 3: Check which version of Python is installed; if the 3.7.0 version is not there, uninstall it through the control panel and
- Step 4: Install Python 3.7.0 using Cmd.
- Step 5: Install the all libraries that required to run the project
- Step 6: Run

1.3 Python Features:

1) **Easy:** Because Python is a more accessible and straightforward language, Python programming is easier to learn.
2) **Interpreted language:** Python is an interpreted language, therefore it can be used to examine the code line by line and provide results.
3) **Open Source:** Python is a free online programming language since it is open-source.
4) **Portable:** Python is portable because the same code may be used on several computer standard
5) **libraries:** Python offers a sizable library that we may utilize to create applications quickly.

6) **GUI:** It stands for GUI (Graphical User Interface)

7) **Dynamical typed:** Python is a dynamically typed language, therefore the type of the value will be determined at runtime.

1.4 Python GUI (Tkinter)

- Python provides a wide range of options for GUI development (Graphical User Interfaces).
- Tkinter, the most widely used GUI technique, is used for all of them.
- The Tk GUI toolkit offered by Python is used with the conventional Python interface.
- Tkinter is the easiest and quickest way to write Python GUI programs.
- Using Tkinter, creating a GUI is simple.
- A part of Python's built-in library is Tkinter. The GUI programs were created.
- Python and Tkinter together give a straightforward and quick way. The Tk GUI toolkit's object-oriented user interface is called Tkinter.

Making a GUI application is easy using Tkinter. Following are the steps:

1) Install the Tkinter module in place.

2) The GUI applicatioMakeske the primary window

3) Include one or more of the widgets mentioned above in the GUI application.

4) Set up the main event loop such that it reacts to each user-initiated event.

Although Tkinter is the only GUI framework included in the Python standard library, Python includes a GUI framework. The default library for Python is called Tkinter. Tk is a scripting language often used in designing, testing, and developing

GUIs. Tk is a free, open-source widget toolkit that may be used to build GUI applications in a wide range of computer languages.

1.5 Python IDLE

- Python IDLE offers a full-fledged file editor, which gives you the ability to write and execute Python programs from within this program. The builtin file editor also includes several features, like code completion and automatic indentation, that will speed up your coding workflow.
- Guido Van Rossum named Python after the British comedy group Monty Python while the name IDLE was chosen to pay tribute to Eric Idle, who was one of the Monty Python's founding members. IDLE comes bundled with the default implementation of the Python language since the 01.5. 2b1 release
- IDLE is used to execute statements similar to Python Shell. IDLE is used to create, modify, and execute Python code. IDLE provides a fullyfeatured text editor to write Python scripts and provides features like syntax highlighting, auto-completion, and smart indent.
- IDLE has two modes: interactive and script. We wrote our first program, “Hello, World!” in interactive mode. Interactive mode immediately returns the results of commands you enter into the shell. In script mode, you will write a script and then run it.
- The IDE Python IDLE is a good place to start as it helps you become familiar with the way Python works and understand its syntax. This IDE is good to start programming in Python due to its great debugger, but once you are fluent and start developing projects it is necessary to jump to another, more complete IDE.
- Python IDLE (Integrated Development and Learning Environment) is an interactive development environment included with the Python programming language. It provides a convenient way to write, execute, and debug Python code.

When you install Python, IDLE is typically installed along with it. To open IDLE, you can follow these steps: ○ Open the command prompt (Windows) or terminal (macOS/Linux). ○ Type "idle" and press Enter. Alternatively, you can specify the version with "idle3" or "idle2" for Python 3 or Python 2, respectively. ○ Once IDLE is launched, you will see the Python shell, which is an interactive environment where you can type and execute Python code directly.

Here are some features and functionalities provided by Python IDLE:

- Editor: IDLE includes a text editor where you can write your Python code. It offers syntax highlighting, automatic indentation, and code completion to enhance your coding experience.
- Interactive Shell: The Python shell in IDLE allows you to execute Python code interactively. You can type commands, statements, or function calls directly in the shell, and Python will execute them immediately.
- Debugging: IDLE provides basic debugging capabilities to help you find and fix errors in your code. You can set breakpoints, step through code, inspect variables, and track the program's execution.
- Python Help: IDLE provides access to the Python documentation and builtin help. You can access the help menu to find information about Python modules, functions, classes, and more.
- Script Execution: In addition to the interactive shell, IDLE allows you to run Python scripts stored in files. You can write your code in the editor and execute it as a script to see the output or interact with the program.
- Customization: IDLE can be customized to suit your preferences. You can modify settings related to syntax highlighting, indentation, fonts, and more.
- Python IDLE serves as a beginner-friendly development environment and learning tool. It is suitable for writing small scripts, testing code snippets, experimenting with Python features, and learning the language's basics. However, for more advanced development projects, you may consider using

other code editors or integrated development environments (IDEs) that provide additional features and better project management capabilities.

1.6 Libraries

In Python, libraries (also referred to as modules or packages) are collections of prewritten code that provide additional functionality and tools to extend the capabilities of the Python language. Libraries contain reusable code that developers can leverage to perform specific tasks without having to write everything from scratch.

Python libraries are designed to solve common problems, such as handling data, performing mathematical operations, interacting with databases, working with files, implementing networking protocols, creating graphical user interfaces (GUIs), and much more. They provide ready-to-use functions, classes, and methods that simplify complex operations and save development time.

Libraries in Python offer various advantages:

- Code Reusability:
- Efficiency:
- Collaboration
- Domain-Specific Functionality
- To use a Python library, you need to install it first.

There are some libraries following:

- **Pandas:**

Pandas are a Python computer language library for data analysis and manipulation. It offers a specific operation and data format for handling time series and numerical tables. It differs significantly from the release3-clause of the BSD license. It is a well-liked open-source of opinion that is utilized in machine learning and data analysis.

Pandas are a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy

and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python. Pandas are a Python library used for working with data sets.

- It has functions for analysing, cleaning, exploring, and manipulating data.
- The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.
- Pandas allow us to analyse big data and make conclusions based on statistical theories.
- Pandas can clean messy data sets, and make them readable and relevant.

Relevant data is very important in data science. Pandas are a Python library for data analysis. Started by Wes McKinney in 2008 out of a need for a powerful and flexible quantitative analysis tool, pandas have grown into one of the most popular Python libraries. It has an extremely active community of contributors. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

Its name is a play on the phrase "Python data analysis" itself.

- **NumPy:**

The NumPy Python library for multi-dimensional, big-scale matrices adds a huge number of high-level mathematical functions. It is possible to modify NumPy by utilizing a Python library. Along with line, algebra, and the Fourier transform operations, it also contains several matrices-related functions.

NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of highlevel mathematical functions that operate on these arrays and matrices.

- NumPy is a Python library used for working with arrays.
- It also has functions for working in domain of linear algebra, Fourier transform, and matrices.

- NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.
- NumPy stands for Numerical Python.
- In Python we have lists that serve the purpose of arrays, but they are slow to process.
- NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
- The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.
- Arrays are very frequently used in data science, where speed and resources are very important.

Matplotlib:

It is a multi-platform, array-based data visualization framework built to interact with the whole SciPy stack. MATLAB is proposed as an open-source alternative. Matplotlib is a Python extension and a cross-platform toolkit for graphical plotting and visualization.

Matplotlib is a popular Python library for creating static, animated, and interactive visualizations. It provides a flexible and comprehensive set of tools for generating plots, charts, histograms, scatter plots, and more. Matplotlib is widely used in various fields, including data analysis, scientific research, and data visualization.

Here are some key features and functionalities of the Matplotlib library:

- Plotting Functions
- Customization Options
- Multiple Interfaces
- Integration with NumPy and pandas Subplots and Figures:
- Saving and Exporting

- **Scikit-learn:**

The most stable and practical machine learning library for Python is scikit-learn. Regression, dimensionality reduction, classification, and clustering are just a few of the helpful tools it provides through the Python interface for statistical modeling and machine learning. It is an essential part of the Python machine learning toolbox used by JP Morgan. It is frequently used in various machine learning applications, including classification and predictive analysis.

Scikit-learn (also referred to as sklearn) is a widely used open-source machine learning library for Python. It provides a comprehensive set of tools and algorithms for various machine learning tasks, including classification, regression, clustering, dimensionality reduction, model selection, and pre-processing.

Here are some key features and functionalities of the Scikit-learn library:

- Easy-to-Use Interface:
- Broad Range of Algorithms:
- Data Pre-processing and Feature Engineering:
- Model Evaluation and Validation:
- Integration with NumPy and pandas:
- Robust Documentation and Community Support:

- **Keras:**

* Google's Keras is a cutting-edge deep learning API for creating neural networks. It is created in Python and is designed to simplify the development of neural networks. Additionally, it enables the use of various neural networks for computation. Deep learning models are developed and tested using the free and open-source Python software known as Keras.

Keras is a high-level deep learning library for Python. It is designed to provide a user-friendly and intuitive interface for building and training deep learning models. Keras acts as a front-end API, allowing developers to define and configure neural

networks while leveraging the computational backend engines, such as Tensor Flow or Theano.

Here are some key features and functionalities of the Keras library:

- User-Friendly API
- Multi-backend Support
- Wide Range of Neural Network Architectures Pre-trained Models and Transfer Learning:
- Easy Model Training and Evaluation:
- GPU Support:

- **h5py:**

* The h5py Python module offers an interface for the binary HDF5 data format. Thanks to p5py, the top can quickly halt the vast amount of numerical data and alter it using the NumPy library. It employs common syntax for Python, NumPy, and dictionary arrays.

h5py is a Python library that provides a simple and efficient interface for working with datasets and files in the Hierarchical Data Format 5 (HDF5) format. HDF5 is a versatile data format commonly used for storing and managing large volumes of numerical data.

Here are some key features and functionalities of the h5py library:

- HDF5 File Access Dataset Handling:
- Group Organization:
- Attributes:
- Compatibility with NumPy
- Performance

- **Tensor flow**

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow. TensorFlow is an end-to-end open source platform for machine learning. TensorFlow is a rich system for managing all aspects of a machine learning system; however, this class focuses on using a particular TensorFlow API to develop and train machine learning models.

TensorFlow is a popular open-source library for machine learning and deep learning. It provides a comprehensive set of tools, APIs, and computational resources for building and training various types of machine learning models, especially neural networks.

Here are some key features and functionalities of TensorFlow:

- Neural Network Framework:
- Computational Graphs
- Automatic Differentiation
- GPU and TPU Support Distributed Computing
- Deployment Capabilities

- **Tkinter**

Tkinter is an acronym for "Tk interface". Tk was developed as a GUI extension for the Tcl scripting language by John Ousterhout. The first release was in 1991. Tkinter is the de facto way in Python to create Graphical User interfaces (GUIs) and is included in all standard Python Distributions. In fact, it's the only framework built into the Python standard library.

Tkinter is a standard Python library used for creating graphical user interfaces (GUIs). It provides a set of modules and classes that allow you to develop interactive and visually appealing desktop applications.

Here are some key features and functionalities of Tkinter:

- Cross-Platform Compatibility
- Simple and Easy-to-Use
- Widgets and Layout Management
- Event-Driven Programming
- Customization and Styling
- Integration with Other Libraries

- **NLTK**

NLTK is a toolkit build for working with NLP in Python. It provides us various text processing libraries with a lot of test datasets. A variety of tasks can be performed using NLTK such as tokenizing, parse tree visualization, etc NLTK (Natural Language Toolkit) is the go-to API for NLP (Natural Language Processing) with Python. It is a really powerful tool to pre-process text data for further analysis like with ML models for instance. It helps convert text into numbers, which the model can then easily work with.

NLTK (Natural Language Toolkit) is a Python library widely used for working with human language data and implementing natural language processing (NLP) tasks. It provides a set of tools, corpora, and resources for tasks such as tokenization, stemming, tagging, parsing, sentiment analysis, and more.

Here are some key features and functionalities of NLTK:

- Text Processing
- Part-of-Speech Tagging
- Named Entity Recognition
- Chunking and Parsing Sentiment Analysis:
- WordNet Integration:

- **Scipy**

SciPy is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data.

SciPy is a powerful scientific computing library for Python that provides a wide range of mathematical algorithms and functions. It builds upon NumPy, another fundamental library for numerical computing, and extends its capabilities by adding additional tools for scientific and technical computing tasks.

Here are some key features and functionalities of SciPy:

- Numerical Integration:
- Optimization and Root Finding
- Linear Algebra
- Signal and Image Processing
- Statistics

Code :

Test.py

```
import pandas as pd
import json
import pickle
from keras import layers, activations, models, preprocessing, optimizers
from keras.utils import to_categorical
#from gensim.models import Word2Vec
```

```
import re
import numpy as np
import os

X = []
Y = []

with open("Dataset/covid-19.json") as f:
    json_data = json.load(f)

json_data = json_data['intents']
for i in range(len(json_data)):
    question = json_data[i]['patterns']
    answer = json_data[i]['responses']
    for i in range(len(question)):
        for j in range(len(answer)):
            X.append(question[i])
            Y.append(answer[j])

tokenizer = preprocessing.text.Tokenizer()
tokenizer.fit_on_texts( X + Y )
VOCAB_SIZE = len( tokenizer.word_index )+1

vocab = []
for word in tokenizer.word_index:
```

```
    vocab.append(word)


def tokenize(sentences):
    tokens_list = []
    vocabulary = []
    for sentence in sentences:
        sentence = sentence.lower()
        sentence = re.sub('[^a-zA-Z0-9]', ' ', sentence)
        tokens = sentence.split()
        vocabulary += tokens
        tokens_list.append(tokens)
    return tokens_list, vocabulary


#encoder_input_data
tokenized_questions = tokenizer.texts_to_sequences(X)
maxlen_questions = max( [len(x) for x in tokenized_questions ] )
padded_questions = preprocessing.sequence.pad_sequences(
tokenized_questions, maxlen = maxlen_questions, padding = 'post')
encoder_input_data = np.array(padded_questions)
print(encoder_input_data.shape, maxlen_questions)


# decoder_input_data
tokenized_answers = tokenizer.texts_to_sequences(Y)
maxlen_answers = max( [ len(x) for x in tokenized_answers ] )
padded_answers = preprocessing.sequence.pad_sequences(
tokenized_answers , maxlen=maxlen_answers , padding='post' )
```

```
decoder_input_data = np.array( padded_answers )

print( decoder_input_data.shape , maxlen_answers )


# decoder_output_data

tokenized_answers = tokenizer.texts_to_sequences(Y)

for i in range(len(tokenized_answers)) :

   tokenized_answers[i] = tokenized_answers[i][1:]

padded_answers = preprocessing.sequence.pad_sequences( tokenized_answers , maxlen=maxlen_answers , padding='post' )

onehot_answers = to_categorical( padded_answers , VOCAB_SIZE )

decoder_output_data = np.array( onehot_answers )

print( decoder_output_data.shape )


encoder_inputs = layers.Input(shape=( maxlen_questions , ))

encoder_embedding = layers.Embedding( VOCAB_SIZE, 200 , mask_zero=True ) (encoder_inputs)

encoder_outputs , state_h , state_c = layers.LSTM( 200 , return_state=True )( encoder_embedding )

encoder_states = [ state_h , state_c ]


decoder_inputs = layers.Input(shape=( maxlen_answers ,  ))

decoder_embedding = layers.Embedding( VOCAB_SIZE, 200 , mask_zero=True) (decoder_inputs)

decoder_lstm = layers.LSTM( 200 , return_state=True , return_sequences=True )

decoder_outputs , _ , _ = decoder_lstm ( decoder_embedding , initial_state=encoder_states )
```

```
decoder_dense = layers.Dense( VOCAB_SIZE , activation='softmax')

output = decoder_dense ( decoder_outputs )


model = models.Model([encoder_inputs, decoder_inputs], output )

model.compile(optimizer=optimizers.RMSprop(),
loss='categorical_crossentropy', metrics = ['accuracy'])


if os.path.exists("model/model.h5") == False:

    hist = model.fit([encoder_input_data , decoder_input_data],
decoder_output_data, batch_size=50, epochs=150)

  model.save('model/model.h5')

  f = open('model/lstm_history.pckl', 'wb')

  pickle.dump(hist.history, f)

  f.close()

else:

  model.load_weights('model/model.h5')


def make_inference_models():

  encoder_model = models.Model(encoder_inputs, encoder_states)

  decoder_state_input_h = layers.Input(shape=( 200 ,))

  decoder_state_input_c = layers.Input(shape=( 200 ,))

        decoder_states_inputs = [decoder_state_input_h,
decoder_state_input_c]

          decoder_outputs, state_h, state_c =
decoder_lstm(decoder_embedding ,
initial_state=decoder_states_inputs)

  decoder_states = [state_h, state_c]
```

```
    decoder_outputs = decoder_dense(decoder_outputs)

        decoder_model = models.Model([decoder_inputs] + decoder_states_inputs, [decoder_outputs] + decoder_states)

    return encoder_model , decoder_model


def str_to_tokens( sentence : str ):

    words = sentence.lower().split()

    tokens_list = list()

    for word in words:

        tokens_list.append( tokenizer.word_index[ word ] )

     return preprocessing.sequence.pad_sequences( [tokens_list] , maxlen=maxlen_questions , padding='post')


enc_model , dec_model = make_inference_models()

for i in range(len(X)):

    states_values = enc_model.predict( str_to_tokens( X[i] ) )

    empty_target_seq = np.zeros( ( 1 , 1 ) )

    empty_target_seq[0, 0] = tokenizer.word_index['wearing']

    stop_condition = False

    decoded_translation = ''

    while not stop_condition :

        dec_outputs , h , c = dec_model.predict(empty_target_seq + states_values )

        sampled_word_index = np.argmax( dec_outputs[0, -1, :] )

        sampled_word = None

        for word , index in tokenizer.word_index.items() :
```

```
            if sampled_word_index == index :
                decoded_translation += ' {}'.format( word )
                sampled_word = word

            if sampled_word == 'end' or len(decoded_translation.split()) > maxlen_answers:
                stop_condition = True

        empty_target_seq = np.zeros( ( 1 , 1 ) )
        empty_target_seq[ 0 , 0 ] = sampled_word_index
        states_values = [ h , c ]
    print(X[i])
    print( decoded_translation )
    print()
```