

Received 17 November 2022, accepted 2 December 2022, date of publication 6 December 2022,
date of current version 13 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3227208

RESEARCH ARTICLE

An AI-Based Medical Chatbot Model for Infectious Disease Prediction

SANJAY CHAKRABORTY¹, HRITHIK PAUL², SAYANI GHATAK², SAROJ KUMAR PANDEY³,
ANKIT KUMAR³, KAMRED UDHAM SINGH⁴, AND MOHD ASIF SHAH^{5,6}

¹Computer Science and Engineering Department, Techno International Newtown, Kolkata 700156, India

²Computer Science and Engineering Department, JIS University, Kolkata 700109, India

³Department of Computer Engineering and Applications, GLA University, Mathura 281406, India

⁴Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701, Taiwan

⁵Department of Economics, Kebri Dehar University, Kebri Dehar 250, Ethiopia

⁶School of Business, Woxsen University, Hyderabad 502345, India

Corresponding author: Mohd Asif Shah (drmohtasifshah@kdu.edu.et)

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.

I. 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 twenty-first 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

The associate editor coordinating the review of this manuscript and approving it for publication was Shadi Alawneh¹.

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 health-related 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 AI-assisted 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.

II. RELATED WORK

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 corpus-based 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 SARS-CoV-2 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 patient-dependent 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 scene-text 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 position-specific score matrices [35].

III. 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 h_{t-1} and the information from the current cell can be stated as x_t . 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. h_{t-1} and x_t 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. h_{t-1} and x_t 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.

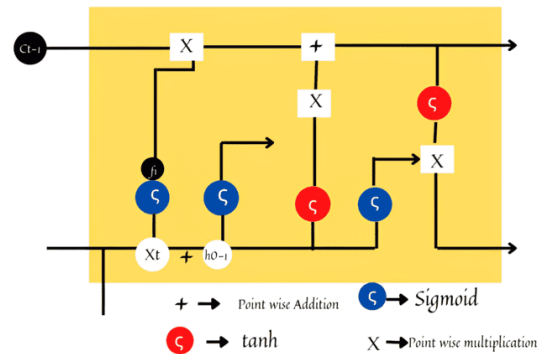


FIGURE 1. Functioning of Forget gate in LSTM algorithm.

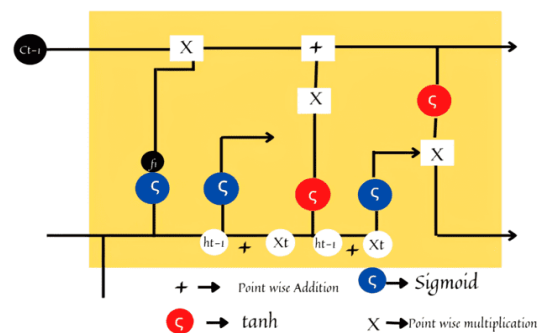


FIGURE 2. Functioning of the input gate in LSTM algorithm.

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

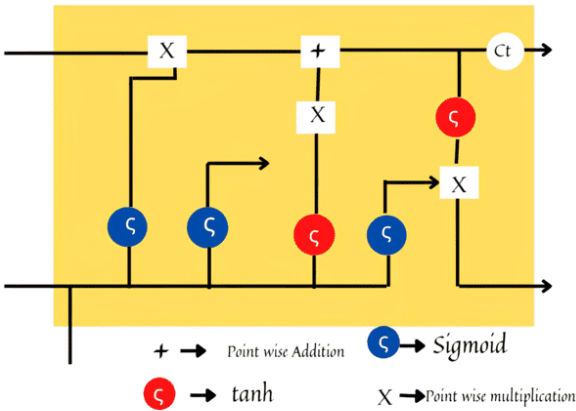


FIGURE 3. Functioning of output gate in LSTM algorithm.

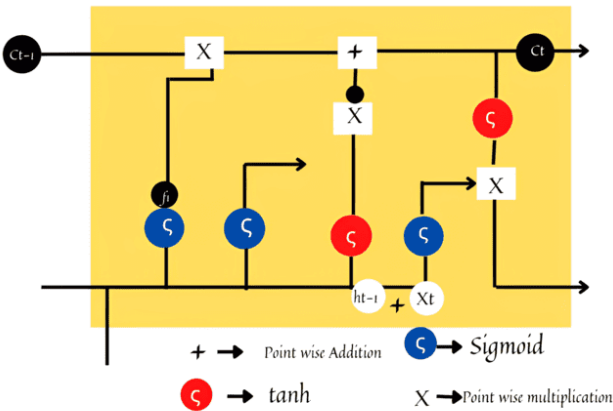


FIGURE 4. Functioning of the cell gate in the LSTM algorithm.

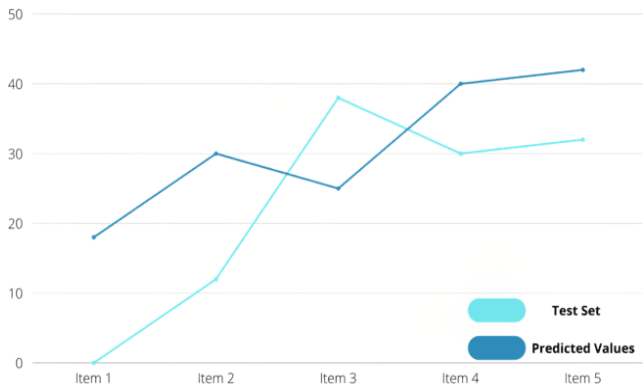


FIGURE 5. Graph showing test set versus predicted values using LSTM in our model.

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.

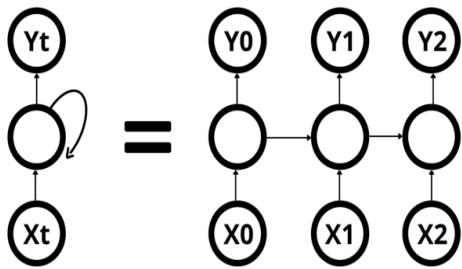


FIGURE 6. The working procedure of the general RNN algorithm.

In Fig.6, the working of the general RNN algorithm is shown, where it can be seen that the algorithm works with an efficient process flow in which the input is sent to a hidden layer that stores the information of the recent past and then, in turn, it produces it in the output layer after analyzing it with the present input. This saves a lot of time during the training phase and helps us to decrease the time complexity of our model.

C. WORKING WITH RNN

In our model, sequential data has been provided during the training of the chatbot model, and the RNN algorithm has helped segregate all the different kinds of queries given as inputs and provide the correct output, which is the information provided to the user. The two major issues regarding RNN have been dealt with efficiently in our model:

- **Exploding Gradients** – Sometimes, the algorithm assigns unusually high importance to the weights, which results in exploding gradients. We have reduced such a possibility significantly by truncating and squashing gradients by providing all types of relevant weights and training the model with the inputs.
- **Vanishing Gradients** - It can happen when the values given for a gradient become very small, and the model stops studying or takes time very much, increasing the time complexity as a result. We have strictly maintained

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.

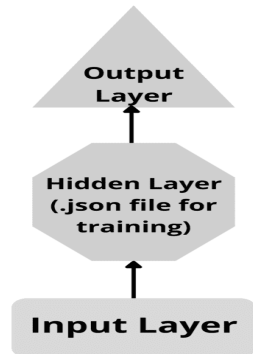


FIGURE 7. Demonstration of the RNN algorithm as applied to our chatbot model.

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.

D. 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.

$$\text{Information Gain} = \text{Entropy}(S) - [(\text{Weighted Average}) * \text{Entropy (each feature)}]$$

$$\text{Gini Index} = 1 - \sum_j P_j^2 \quad (1)$$

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).

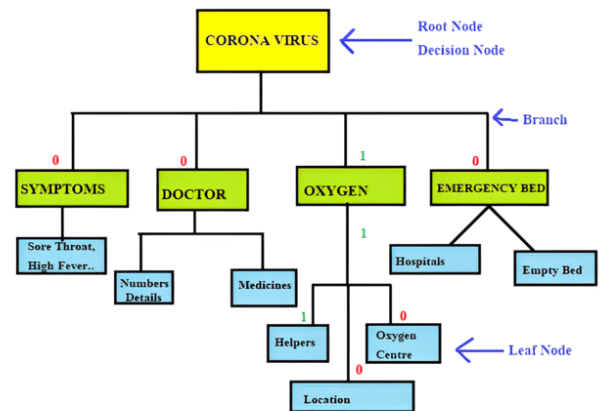


FIGURE 8. Demonstration of Decision Tree Model as it is applied to our chatbot model.

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.

IV. 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

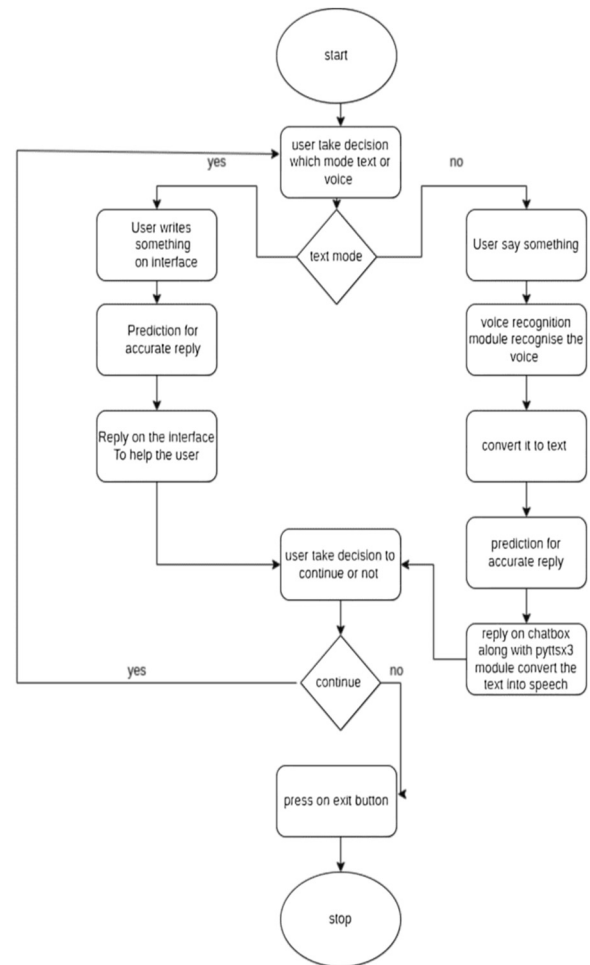


FIGURE 9. Flowchart of the overall chatbot system.

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 chat-chat between the user and the bot is done in the bot’s interface, which is made in python using the Tkinter library.

V. RESULT ANALYSIS AND DISCUSSION

In order to fully understand our model and its results, it is necessary to discuss the dataset that was used in the making of the model. The reason for this is that text-based predictions are needed to facilitate inter-communication with the user in order to solve their complex issue.

A. DATASET DESCRIPTION

This dataset is in JSON format and has been incorporated into our model. We have included some keys which have been used for pattern recognition and input-output mapping. Key factors include questions and responses. In order to separate each section’s key from the others, a parent key named tag has been used. In each key, there are text-based features.

Algorithm 1 Pseudocode of the Proposed Chatbot System

Step 1: Make a json file along with tags, patterns and response. Here, tags will represent the group, patterns represent the question or query format of the user and the response represents how the bot will respond.

Step2: load json file, intents
`=json.loads(open('intents.json').read())`

Step3: ignore unnecessary letters, ignore_letters= ['?', '!']

step 4: Store each word of patterns and tags, lemmatize them.
`words=[lemmatizer.lemmatize("word") for word in words if word is not in ignore_letters]`

Step 5: sorts the words for not repeating the same word in model, `words=sorted(set(words))`

Step 6: open the files in binary mode and make dump file for it,
`pickle.dump(words, open('words.pkl', 'wb'))`
`pickle.dump(words, open('class.pkl', 'wb'))`

Step7: add all words, patterns after lemmatizing, to training list `training= []`, empty list
`word_patterns=[lemmatizer.lemmatize("word.lower") for word in word_patterns]`, for lemmatizing Loop : word in words list -(will take each element from words list to word)
`bag.append(1)` if word in word_patterns or if it is not in word_patterns, bag append the value of "0"
`output_Row=list of output empty, list(output_empty)`
`output_Row of classes.index of documents=1`
`training.append(list of bag and output_Row)`

Step 8: shuffle all words in training list, `random.shuffle(training)`

step 9: store training list one part to train_x and another part to train_y
`train_x=stores list one part,0th index elements, list (:0)`
`train_y=stores 1st index elements, list (:1)`

Step 10: make neural networking by adding Dense and dropout to model
`model=linear stack of layers`
`# To add Dense and Dropout`
`model.add(Dense of (128 and train_x))`
`model.add (Dropout of 0.6)`
`sequential () model. Add (Dense of (train_y[0])'s length)`

Step 11: compile it and save the model
`Compile (model, for compiling)`
`Fit the model, model.fit()`
`Save it as.h5 file, save(chatbot_model.h5)`

Step 12: after successfully making the training model whenever a user writes the text on the chatbox, the text will be Prediction (message or text)

step 13: give reply to the users
`response = chatbot_response for the message`
`chatbot.insert (inserting the "response" to chatbot interface)`

These datasets contain different symptoms and diseases such as Covid-19, Deep Fever, and chronic diseases. Furthermore, the dataset includes their treatments, methods, RT PCR tests,

Step 14: if the user wants to hear it on voice also

```
engine = pyttsx.init()
engine.say(command)
engine.runAndWait()
```

Where the command is the bot's reply, which is converted into the voice

step 15: this cycle will continue until the users press exit button

step 16: if the user presses the exit button, the application will close

TABLE 1. Dataset key and contain data type.

Key Named	Key-type	Contain type
symptoms	Parent	Recognition pattern
medicine	Parent	Recognition pattern
responses	Child	Shuffling answers
patterns	Child	Shuffling questions
Covid	Parent	Recognition Pattern

side effects, past infections, and more reliable solutions. The dataset contained doctor's numbers, nearby hospitals and all needy phone numbers. In addition to detailed treatment options, the user is provided with real-time phone numbers of doctors, ambulances, and many other services.

From the above TABLE 1, there are some keys categorized as parents and some keys used as child. There are more parent keys and child keys, more than 20 in number. All the child keys contained text-based data. Whenever a user writes a query, the bot will recognize at which pattern the responses should be given about after creating a successful neural network-based model using the above Pattern keys. As a result, it helps to understand the pattern and text for the model better.

After the dataset description, the most critical points that should be addressed as significant topics are the configurations of the hardware and the software on the system. All the code has been written in python language and text-editors are used here spyder vs code. In addition, the application is built in Tkinter GUI, which is comfortable for any type of pc. The test and demo work have been done in mentioned configuration hardware: -

- Intel core i3 10th gen
- 1 TB HDD
- 4 GB RAM
- Windows 10th operating system

Even in some other operating systems and hardware requirements, the application works perfectly.

- Ubuntu operating system, Fedora operating system
- 128 GB SSD
- 1 GB RAM TO 12GB RAM up to

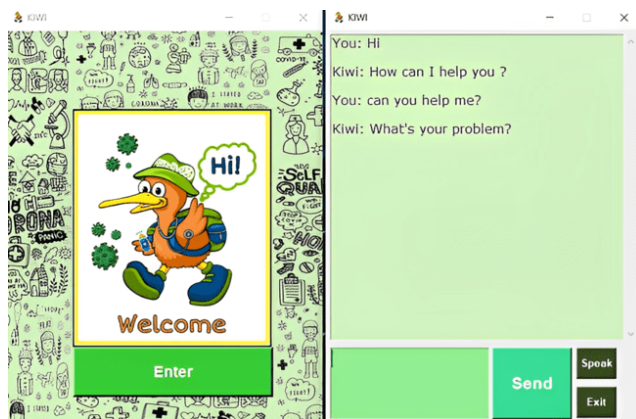
The key model parameters are shown in TABLE 2 above, just as we mentioned when discussing about the hardware

TABLE 2. Proposed deep learning model architecture.

Layers	Units-Neurons	Activation Function	Optimizer
Input Layer	128units	-	-
Hidden Layers	64units	ReLU	Categorical_crossentropy and Adam
Output Layer	23units	Softmax	-

configuration before. To create more accurate predictions and intricate free neural networks, the 50% dropout is also applied while using the hidden layer. Furthermore, in order to create a more biased less model, both parameters kernel_initializer and bias_initializer are also set to uniform and zeros.

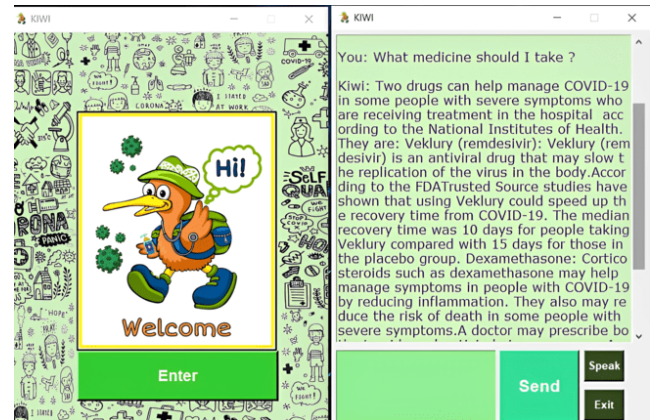
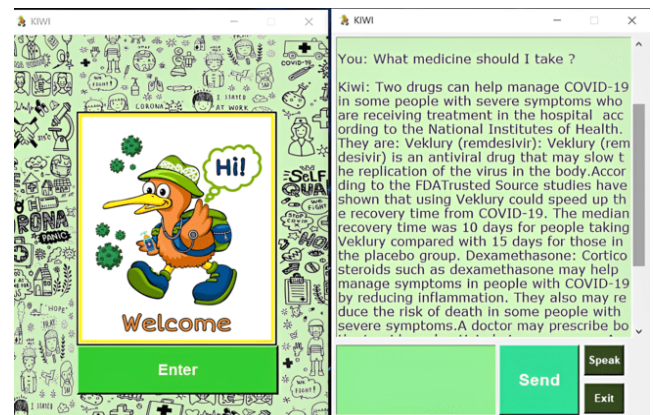
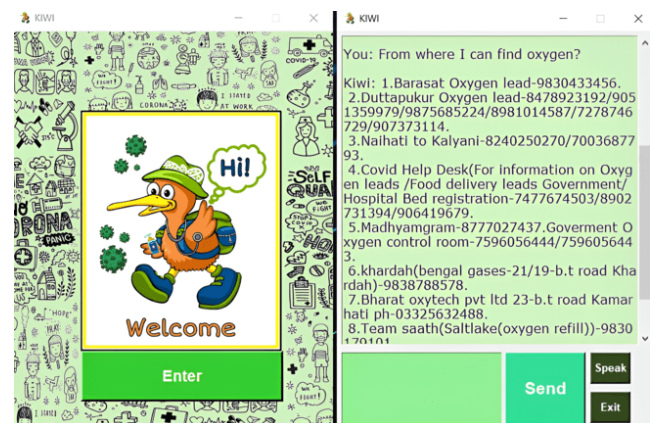
As we mentioned model architecture and software-hardware configuration, the application operates without a hitch in a variety of operating systems and hardware configurations due to its lightweight and superior architectural model. Therefore, we can say almost every device not only as exe file even in mobile-app mode the application works perfectly. In addition, some pictures of how the bot interacts with users are shown below. The below Fig. 10 to Fig.15 show how the bot starts its interaction with the user

**FIGURE 10.** Shows the bot interaction.

B. TRAINING AND TESTING PHASES

After loading the json file and making a training model, the training model helps the bot to predict the perfect answers for the user. Therefore, before using it in the main file we have to check the training and testing phases and also check the accuracy and loss percentage. The below table is about to check the accuracy and a loss percentage of the training model.

As we can see in table 1, the accuracy and loss change per epoch. It depends on the training model. Here, the accuracy increases as the epoch increases and loss decreases.

**FIGURE 11.** User asks a different question about medicine (not mentioning particular and still bot gives closer reply by giving the user all type of medicines details).**FIGURE 12.** User asks a direct question about covid-19 and the bot gives a direct answer to it (direction question and direct reply).**FIGURE 13.** Users ask for oxygen, and the bot gives the phone numbers of oxygen supplier.

It means our bot's prediction becomes higher and more accurate. If the accuracy becomes low or not stable, it causes problems in the bot's prediction. Meanwhile, some unexpected replies will come from the bot when the user asks something or makes a query. In the above table, the highest



FIGURE 14. Users ask the bot in voice mode (the asking question is also written in chatbot, whatever the user says) and the bot reply in both mode voice and text.

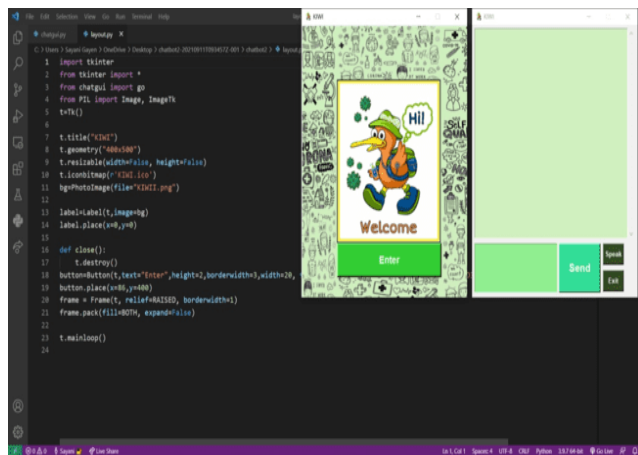


FIGURE 15. Shows the opening interface to a user along with the chatbot interface.

TABLE 3. Highest accuracy of training model.

Training model	epoch	Batch_size	loss	Accuracy
chatbot_mod el.h5	200/200	15	0.1232	0.9432

accuracy is shown on the epoch 200/200. Here the loss is a minimum, which is 0.1232 and the highest accuracy is 94.32%. For getting the results, the algorithm, which is used, LSTM has four different components. Forget gate, input gate, output gate, cell gate. All gates have different roles to perform. The forget gate helps to decide what information will be kept or thrown away, and the equation consists of the forget gate,

$$F = \sigma(u_1X + \omega_1H_2) \quad (2)$$

The input gate helps to tell what value will be updated, the equation which consists of the input gate equation: -

$$I = \sigma(u_2X + \omega_2H') \quad (3)$$

and the cell gate will combine all forget and input gate information which is decided to keep and the equation for the cell gate

$$C = C'F + IG \quad (4)$$

In the last output gate will give the output and its equation for it $O = \sigma(u_4X + \omega_4H'_2)$

$$H' = Q \tan^2(C) \quad (5)$$

$$\text{Output } Xh' \quad (6)$$

In addition, all these gates are combined and made a perfect training model containing different loss and accuracy. Fig. 16, the accuracy increases per epoch. However, when the epoch is 1/200 to 3/200 the accuracy slowly increases then again in epoch 5/200 to 7/200, the accuracy becomes steady, and in the 9/200 to up to last, the accuracy becomes higher than the previous epoch. In the case of test cases, we can see figure 2 how the bot starts its interaction with the user. Here, as the accuracy of the training model goes well and increases per epoch, the user gets a closer reply. If the accuracy becomes low or not proficient, the replies will not be closer or unexpected, or unnecessary for the user. Therefore, if we want a closer prediction, we should maintain the accuracy higher and higher. Because higher accuracy makes better proficient, neural layers and these layers consist of the data that is helping to build a perfect bot.

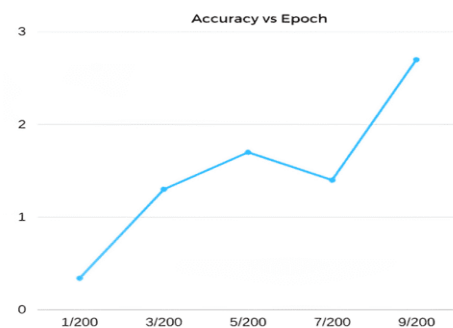


FIGURE 16. The accuracy vs epoch graph.

Here, the encoder-decoder configuration, which is hidden, plays an important role in the higher accuracy. The information is passed from another layer to another through a repeating vector layer. To get the loss and accuracy, a bidirectional wrapper is added to the LSTM layers. It allows the model for both results and makes a great performance.

According to TABLE 4, we can see that the highest accuracy of our model testing is 93.45 and the phase number is 4. That means after 4 phases and many more phases, we got the highest accuracy and perfect reply from our bot. In other

TABLE 4. Accuracy testing phase of model.

Model	Testing phase	Accuracy (%)
chatbot_model.h5	1	45.2
chatbot_model.h5	2	66.6
chatbot_model.h5	3	87.8
chatbot_model.h5	4	93.45
chatbot_model.h5	5	92.5

phrases, the chatbot doesn't give a close reply to the user, and the accuracy after collecting all the information that is obtained from the conversion between the chatbot and the user is lower than the others. The testing phase is so important because without this testing phase, we can't say how much the bot is going to give a close reply or make good conversions with the user. Even if we implement it in its low accuracy state, the bot will not be helpful for the future user and causes problems when it's about to give the perfect reply which is wanted by the user. Therefore, the testing phase is important and should be accurate. Moreover, the accuracy will be higher if we test the phase continuously. If we do more testing phases, after more phases we can get better accuracy than in phase 4. So, if we represent it in the sense of equation we got,

$A \propto T$ where, T = testing phases and A = accuracy

$A = nT$, where n = constant which represents the number of phases.

So, here the accuracy and testing phases are proportionally the same. If we do more testing phases, we get more good accuracy. And it will increase if the number of testing phases increases.

In figure 9, it is shown that the accuracy increases during the testing phase. In that case, we can say if we want more good accuracy, we should do more test phases. After all this, we get more good models along with good results where users also get the best replies from the bot. In figure 10, how the testing case will look like in each overall accuracy, is shown, In Fig 17, testing cases depend on how the overall accuracy looks like. If the overall accuracy is more proficient than the previous accuracy, the testing cases will also go well and be well predicted. If the accuracy is not better than the previous, the testing cases will not be good. In all test cases, the different result outcomes will be determined, and after that, the training model is chosen in which testing cases become better, as shown in the graph. The layers also consist of good data after collecting the perfect training model. In addition, as the training model and layers become more proficient, the results outcomes also become friendly and good for the users and also make a great interaction by giving them the right predicted replies.

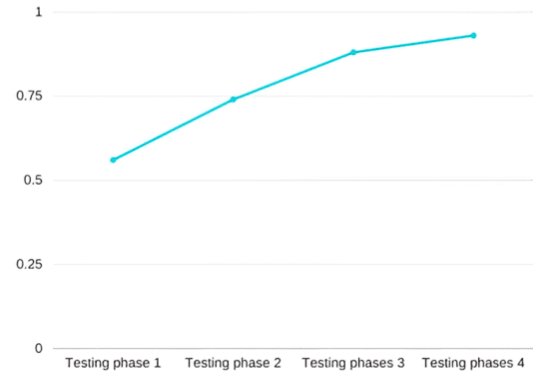


FIGURE 17. Testing phase vs accuracy.

C. TIME COMPLEXITY ANALYSIS

The chatbot interface is made in the Tkinter library. For loading the library and making it a perfect layout for the bot, it takes $O(n)$ time complexity. And for the training model and to predict the correct reply to the user's question, the algorithm's running time cost $O(n)$, and as we know for making the training model and putting different layers on it to make a perfect neural network, time complexity will be - $O(4h(3d+h+d))$ where the neural network layers d , h and it is the proposed time complexity for the LSTM algorithm. Therefore, the total time complexity (along with training model): - $O(n+(4h(3d+h+d)))$ where h , d denotes layers and only for layout and put predict replies,

- Total time complexity: - $O(n)$
- Space complexity: - $O(n)$

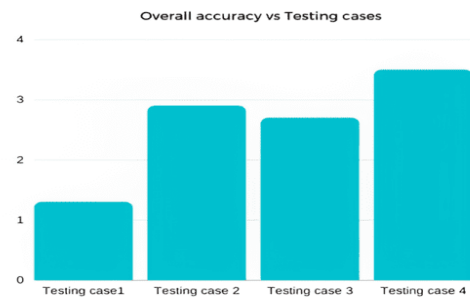


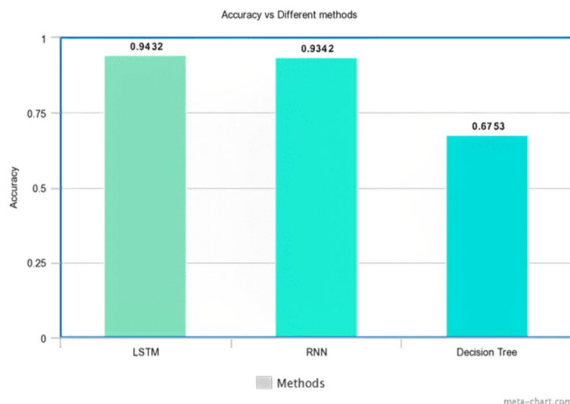
FIGURE 18. Accuracy vs testing cases.

According to TABLE 5, after performing, different methods on our chatbot or making the training model using a different approach, we can see the different accuracy on each different method. Here, LSTM in the case of single-layer and for testing and training approaches the time complexity where, in the case of RNN, the time complexity is the same, but in the case of the decision tree, the time complexity depends on the depth of the tree. Moreover, in the case of accuracy, the highest accuracy is from LSTM. The LSTM

TABLE 5. Accuracy and complexity comparison.

Method	Accuracy	Time complexity
LSTM	0.9432	$O(n)$
RNN	0.9342	$O(n)$
Decision Tree	0.6753	$O(nkd)$

used different gates and also used a gradient to work as a neural network, or we can say a better RNN with a sequential network. In the case of RNN, it stores sequential input and performs suitable operations when it is needed as sequential data output. In addition, the last decision tree makes different nodes and sub-node, and each node performs to store data for users' use. Here, according to test cases and training cases, LSTM, then RNN, and lastly, the Decision Tree got the highest accuracy.

**FIGURE 19. Graphical representation of accuracy in different methods.**

However, according to Fig. 19, we can see that with different methods the accuracy increases differently. LSTM has the highest accuracy, whereas RNN is close to LSTM's accuracy. However, the decision tree has the lowest accuracy among these three methods. Therefore, we can say that for our proposed model and training model, the best approach to use is LSTM. The rest of the two methods are also the best approaches to different proposed models. However, as for our accuracy in the case of LSTM. Therefore, we choose the LSTM approach to make neural network layers and also make our training model to get the best results in cases of interaction with users and getting the perfect or expected reply from a bot.

In the following table, we have outlined some existing approaches to our chatbot. Based on this comparison, it can be clearly seen that our proposed model approaches are more robust with respect to inter-user communication as well as

TABLE 6. Comparison in terms of interface and other properties.

Approaches	Interface	Method
Benjamin-Hochberg method	Blank interface	Using regression methods and null hypothesis to make an interaction using survey less inter-connection friendly
NLP Based RAD model	Single interface	Using NLP to interact directly with users and give respond. Take more time to give accurate results
(Medbot)Hybrid model, using NLP and Dialogflow	Strong multiple-interfaces	Making hybrid models, using NLP and Dialogflow, to make predictions through machine-learning models and predict more accurate results. Predicts more accurate predictions due to hybrid model.
Proposed Method	Double interfaces	Using NLP and LSTM. LSTM used for making deep-learning model and NLP for text-based patterns recognition. Give more accurate and user-friendly interface to user due to double interface and deep learning model.

providing a quicker, more accurate, and more understandable response time.

VI. APPLICATIONS

A. PERSONALIZED CONVERSATIONS

Our chatbot, Kiwi will provide personalized conversations where the users will be able to gain information and at the same time, it will feel like a natural experience to them. The chatbot will be able to show empathy and be able to perform personalized word choices and adjust its language and tone based on the context of the conversation.

B. OMNI-CHANNEL

Our chatbot Kiwi can broadcast information across different major channels including government websites and important organizations, such As WHO which contain relevant information about Covid 19. If the user provides identifying information such as the composition of medicines, hospital or doctor requirements, the chatbot will provide all the necessary details and even provide links to websites containing the required information by the user.

C. ACCESSIBILITY

The chatbot interface is accessible to all users regardless of their platform. The users only need a stable internet

connection to be able to use it. It can read the information aloud if the user prefers to hear the information which he or needs. It is helpful for the visually impaired as they can speak to the chatbot about their queries and the chatbot, in turn, speaks to them and provides important information.

D. CONTEXTUAL ASSISTANCE

The chatbot can also be a source of contextual assistance to the health officials and organizations for updating information about Covid 19. We can get a better understanding of the problems, by troubleshooting customer issues and the work for the next updates can be streamlined with the necessary improvements that need to be undertaken.

E. BOOKINGS

The chatbot will provide all the necessary information about the availability of hospital beds in an area where the user wants the patient to be taken. It would also provide the contact number for the hospitals and the ambulances, so the booking of hospital beds would be easier for the user as they would not have to travel long distances to get a hospital bed. Users can also book an appointment with the doctor if they feel they are infected and need advice right from the safety of their homes. The contact numbers for the doctors will be provided by the chatbot and the users don't have to get out of their homes to gather the information.

F. CONVERSATIONAL SERVICE

The chatbot will be able to provide conversational service for any hospital or health organization by conversing with the patients, and it would be helpful in optimizing the treatment procedure based on the reactions of the patients. The medications can be selected and the people can also be made aware of the dangers posed to them if they fail to take the necessary precautions.

G. STUDENT ASSISTANCE

Students conducting research on COVID-19 or the use of technology in medical science to treat COVID-19 can find useful information from the chatbot. They would find out the accuracy level required to maintain a stable conversation and provide the correct information with respect to the queries given by the user.

VII. BENEFITS

- ✓ **24/7 Support:** Chatbots are always at an advantage when it comes to all-around support even without having to scale your team. The chatbot will be available to offer instant and accurate responses around the clock without the need to take any breaks, as is the case with human deployment.
- ✓ **Save Resources:** Chatbots are always useful in saving the time of operators as well as the users. The users do not have to spend their time searching for health centers or hospitals to get information about Covid 19.

It will also be cost-effective as deploying a chatbot saves money for any organization since they do not need to pay for the manpower required to maintain the equivalent facilities for the users. It also saves the user money as he does not need to use any transportation and travel anywhere to inquire about his needs.

- ✓ **Reduce Stress for users:** It is generally observed that patients avoid getting in contact with the customer service of hospitals. They are more comfortable if they are talking to a chatbot and getting their information. The survey conducted by Help shift discovered that 79% of people prefer live chat to other channels and 55% would choose to use a chatbot if it was available. Chatbots are widely considered to be a quicker and more efficient solution to the users' problems.
- ✓ **Handling Information:** Chatbots are helpful at handling huge amounts of information and providing the correct information for the given query without wasting any time. Our chatbot, Kiwi is trained in such a way that it handles all the information given to it seamlessly and provides it as an output to the user's queries with a high rate of accuracy.

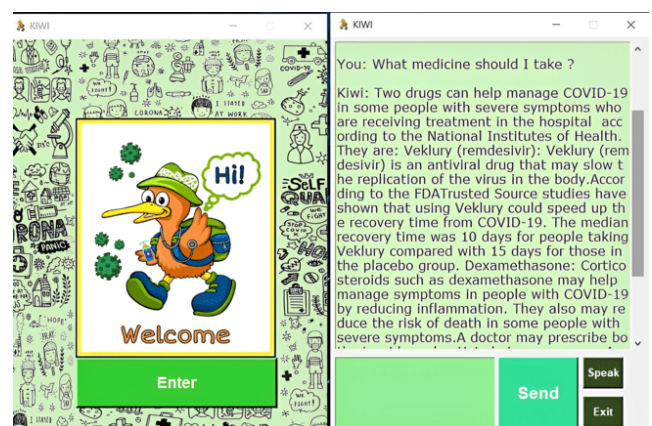


FIGURE 20. The chatbot replies regarding medicines that helps the user to determine his next course of action.

In the given Fig. 20, the user asks the chatbot about what medicines he can take if he is infected. The chatbot provides all the information about the available medications and therapies for Covid 19 along with their advantages and side effects. Therefore, the user gets a full description of his query and will be able to decide his next course of action after going through the information given to him.

- ✓ **Instant response:** Chatbots are generally known for their ability to respond to any query immediately, and they can handle queries from thousands of users at the same time and provide information to the users immediately. The simultaneous answers provided by our chatbot, Kiwi would also help us to analyze the average response time and improve its activities so that the users can get an even more advanced system at their hands.

VIII. CONCLUSION

From this paper, we can conclude that the chatbot is very easy to use for all people; they can use this chatbot in their language. This bot offers medical-related information like doctor's contact details, address of nearby hospitals, contact details for getting an oxygen cylinder, about the disease its symptoms, its prevalence, diagnosis, and its treatment procedures. We consider that our findings will help researchers to take advantage of the advanced information in the layout and other stuff of these innovative technologies, which may be required for continuous development in the functionality of medical chatbots and may help prevent COVID 19. This medical chatbot has wide future opportunities. People in remote areas can also receive benefits from this. Here we use 'TensorFlow', which helps to build the NLP for chatbots and utilizes deep neural network architecture. After building the network for our chatbot, it will predict the correct answers to the user's queries. Even if it is not in the training model, the bot tries to predict it closer by checking the sentences and their words which will be closer to the training model's response.

COMPLIANCE WITH ETHICAL STANDARDS

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

FUNDING

There is no external funding involved in this research.

SUPPLEMENTARY

<https://github.com/Kodierer-desafiador/Covid-19-dataset-json-file>

REFERENCES

- [1] J. Cahn, "CHATBOT: Architecture, design, & development," Dept. Comput. Inf. Sci., Univ. Pennsylvania School Eng. Appl. Sci., Philadelphia, PA, USA, Tech. Rep. EAS499, 2017.
- [2] E. Tebenkov and I. Prokhorov, "Machine learning algorithms for teaching AI chat bots," *Proc. Comput. Sci.*, vol. 190, pp. 735–744, Jan. 2021.
- [3] A. S. Lokman and M. A. Amedeen, "Modern chatbot systems: A technical review," in *Proc. Future Technol. Conf.* Cham, Switzerland: Springer, Nov. 2018, pp. 1012–1023.
- [4] A. Kumar, P. K. Meena, D. Panda, and M. Sangeetha, "Chatbot in Python," *Int. Res. J. Eng. Technol.*, vol. 6, no. 11, 2019.
- [5] S. Raj and K. Raj, *Building Chatbots With Python*. New York, NY, USA: Apress, 2019.
- [6] K. H. Koundinya, A. K. Palakurthi, V. Putnala, and K. A. Kumar, "Smart college chatbot using ML and Python," in *Proc. Int. Conf. Syst., Comput., Automat. Netw. (ICSCAN)*, Jul. 2020, pp. 1–5.
- [7] S. A. Sheikh, "Artificial intelligence based chatbot for human resource using deep learning," Ph.D. dissertation, Dept. Comput. Sci. Eng., Manipal Univ., Manipal, India, 2019.
- [8] M. M. Hossain, S. Krishna Pillai, S. E. Dansy, and A. A. Bilong, "Mr. Dr. Health-assistant chatbot," *Int. J. Artif. Intell.*, vol. 8, no. 2, pp. 58–73, Dec. 2021.
- [9] R. Dharwadkar and N. A. Deshpande, "A medical chatbot," *Int. J. Comput. Trends Technol.*, vol. 60, no. 1, pp. 41–45, 2018.
- [10] D. Madhu, C. J. N. Jain, E. Sebastain, S. Shaji, and A. Ajayakumar, "A novel approach for medical assistance using trained chatbot," in *Proc. Int. Conf. Inventive Commun. Comput. Technol. (ICICCT)*, Mar. 2017, pp. 243–246.
- [11] F. Mehfooz, S. Jha, S. Singh, S. Saini, and N. Sharma, "Medical chatbot for novel COVID-19," in *ICT Analysis and Applications*. Singapore: Springer, 2021, pp. 423–430.
- [12] M. Herriman, E. Meer, R. Rosin, V. Lee, V. Washington, and K. G. Volpp, "Asked and answered: Building a chatbot to address COVID-19-related concerns," *NEJM Catalyst Innov. Care Del.*, vol. 1, pp. 1–13, Jun. 2020.
- [13] S. Altay, A. S. Hacquin, C. Chevallier, and H. Mercier, "Information delivered by a chatbot has a positive impact on COVID-19 vaccines attitudes and intentions," *J. Exp. Psychol., Appl.*, vol. 27, pp. 1–11, Oct. 2021.
- [14] P. Amiri and E. Karahanna, "Chatbot use cases in the COVID-19 public health response," *J. Amer. Med. Inform. Assoc.*, vol. 29, no. 5, pp. 1000–1010, Apr. 2022.
- [15] M. Almalki and F. Azeez, "Health chatbots for fighting COVID-19: A scoping review," *Acta Inf. Medica*, vol. 28, no. 4, p. 241, 2020.
- [16] P. Weber and T. Ludwig, "(Non-) interacting with conversational agents: Perceptions and motivations of using chatbots and voice assistants," in *Proc. Conf. Mensch Comput.*, vol. 1, Sep. 2020, pp. 321–331.
- [17] U. Bharti, D. Bajaj, H. Batra, S. Lalit, S. Lalit, and A. Gangwani, "Medbot: Conversational artificial intelligence powered chatbot for delivering telehealth after COVID-19," in *Proc. 5th Int. Conf. Commun. Electron. Syst. (ICCES)*, Jun. 2020, pp. 870–875.
- [18] G. Battineni, N. Chintalapudi, and F. Amenta, "AI chatbot design during an epidemic like the novel coronavirus," *Healthcare*, vol. 8, no. 2, pp. 1–8, Jun. 2020.
- [19] B. A. Shawar and E. Atwell, "Using dialogue corpora to train a chatbot," in *Proc. Corpus Linguistics Conf.*, Mar. 2003, pp. 681–690.
- [20] M. Ciotti, M. Ciccozzi, A. Terrinoni, W. C. Jiang, C. B. Wang, and S. Bernardini, "The COVID-19 pandemic," *Crit. Rev. Clin. Lab. Sci.*, vol. 57, no. 6, pp. 365–388, 2020.
- [21] S. J. Daniel, "Education and the COVID-19 pandemic," *Prospects*, vol. 49, no. 1, pp. 91–96, 2020.
- [22] N. Rosruen and T. Samanchuen, "Chatbot utilization for medical consultant system," in *Proc. 3rd Technol. Innov. Manag. Eng. Sci. Int. Conf. (TIMES-iCON)*, Dec. 2018, pp. 1–5.
- [23] K. Rarhi, A. Bhattacharya, A. Mishra, and K. Mandal, "Automated medical chatbot," Tech. Rep., 2017.
- [24] S. Majumder and A. Mondal, "Are chatbots really useful for human resource management?" *Int. J. Speech Technol.*, vol. 24, no. 4, pp. 969–977, Dec. 2021.
- [25] A. S. Ashour, A. El-Attar, N. Dey, H. A. El-Kader, and M. M. A. El-Naby, "Long short term memory based patient-dependent model for FOG detection in Parkinson's disease," *Pattern Recognit. Lett.*, vol. 131, pp. 23–29, Mar. 2020, doi: 10.1016/j.patrec.2019.11.036.
- [26] S. J. Fong, N. Dey, and J. Chaki, "AI-enabled technologies that fight the coronavirus outbreak," in *Artificial Intelligence for Coronavirus Outbreak*. Singapore: Springer, 2021, pp. 23–45.
- [27] S. Chakraborty and L. Dey, "The implementation of AI and AI-empowered imaging system to fight against COVID-19—A review," in *Smart Healthcare System Design: Security and Privacy Aspects*. 2022, pp. 301–311.
- [28] L. Dey, S. Chakraborty, and A. Mukhopadhyay, "Machine learning techniques for sequence-based prediction of viral-host interactions between SARS-CoV-2 and human proteins," *Biomed. J.*, vol. 43, no. 5, pp. 438–450, 2020.
- [29] R. K. Tripathi and A. S. Jalal, "A robust approach based on local feature extraction for age invariant face recognition," *Multimedia Tools Appl.*, vol. 81, pp. 21223–21240, Mar. 2022.
- [30] H. Sharma and A. S. Jalal, "A framework for visual question answering with the integration of scene-text using PHOCs and Fisher vectors," *Expert Syst. Appl.*, vol. 190, Mar. 2022, Art. no. 116159.
- [31] U. Singh and M. K. Choubey, "A review: Image enhancement On MRI images," in *Proc. 5th Int. Conf. Inf. Syst. Comput. Netw. (ISCON)*, 2021, pp. 1–6.
- [32] S. Al-Imamy and Y. Hwang, "Cross-cultural differences in information processing of chatbot journalism: Chatbot news service as a cultural artifact," *Cross Cultural Strategic Manag.*, vol. 29, no. 3, pp. 618–638, 2022, doi: 10.1108/CCSM-06-2020-0125.
- [33] D. Shin, "The perception of humanness in conversational journalism: An algorithmic information-processing perspective," *New Media Soc.*, vol. 24, no. 12, pp. 2680–2704, Dec. 2022, doi: 10.1177/1461444821993801.
- [34] T. N. K. Hung, N. Q. K. Le, N. H. Le, L. Van Tuan, T. P. Nguyen, C. Thi, and J.-H. Kang, "An AI-based prediction model for drug-drug interactions in osteoporosis and Paget's diseases from SMILES," *Mol. Informat.*, vol. 41, no. 6, Jun. 2022, Art. no. 2100264.
- [35] N. Q. K. Le and B. P. Nguyen, "Prediction of FMN binding sites in electron transport chains based on 2-D CNN and PSSM profiles," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 18, no. 6, pp. 2189–2197, Nov./Dec. 2021.



SANJAY CHAKRABORTY received the B.Tech. degree from the West Bengal University of Technology, the M.Tech. degree from the National Institute of Technology, Raipur, and the Ph.D. degree from the University of Calcutta. He is currently working as an Associate Professor at the Techno International Newtown. He has published 55 research papers in various international journals, conferences, and book chapters. He has published two internationally authored books. His areas of interests include data mining and machine learning, feature subset selection, and quantum computing. He is a Professional Member of the International Association of Engineers, the Internet Society Kolkata Chapter, the Institute of Research Engineers and Doctors, and the International Computer Science and Engineering Society. He has a total of 11.5 years of teaching and research experience. He worked as a reviewer in several international conferences and journals. He was a recipient of the Silver Medal during his M.Tech. degree, the IEEE Young Professional Best Paper Award in CICBA 2017, the most-cited author in biomedical journal 2022, and the “Innovation Award” for his outstanding achievement in the field of innovation by the Techno India Institution’s Innovation Council, in 2019.



KAMRED UDHAM SINGH received the Ph.D. degree from Banaras Hindu University, India, in 2019. From 2015 to 2016, he was a Junior Research Fellow, and from 2017 to 2019, he was a Senior Research Fellow with University Grant Commission (UGC), India. In 2019, he became an Assistant Professor at the School of Computing, Graphic Era Hill University, India. He is currently a Postdoctoral Fellow at the Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan. He has published several research papers in international peer-reviewed journals. His research interests include image security and authentication, deep learning, medical image watermarking, and steganography.



HRITHIK PAUL is currently pursuing the bachelor’s degree with JIS University, Kolkata. His research interests include machine learning and deep learning.



ANKIT KUMAR received the M.Tech. degree from IIIT Allahabad. He is currently pursuing the Ph.D. degree with the Birla Institute of Technology. He is also an Assistant Professor at the Department of Computer Science, GLA University, Mathura, India. His research is in wireless sensor networks. He has published multiple papers in Taylor and Francis’s networking-related journals. His articles are published in over 20 international journals and six national journals. He has received five patents. He has received a Research Grant from TEQIP. His work has been profiled broadly in information security, cloud computing image processing, neural networks, and networks. His research interests include computer network information security, computational models, compiler design, and data structures. He is a reviewer and the editor at numerous reputed journals.



SAYANI GHATAK is currently pursuing the bachelor’s degree with JIS University, Kolkata. Her research interests include machine learning and deep learning.



MOHD ASIF SHAH received the B.A., M.A., and Ph.D. degrees in sound teaching and research skills. He is currently working as Associate Professor at Kebri Dehar University, Ethiopia. He has been working as an Assistant Professor at FBS Business School, Bengaluru, Karnataka, India; and Lovely Professional University, Punjab, India (AACSB Accredited). He has also served as a Lecturer at the Jamia College of Education and also helped his department with teaching assistance during his Ph.D. He has published more than 40 research papers (SCI/WOS/UGC Indexed) with 32 citations and the H-index is four (Google Scholar). He has attended more than 30 workshops and faculty development programs sponsored by the Government of India and other agencies. He has an excellent grasp of the subject material, with more than five years of using platforms, such as CANVAS, LMS, and UMS for online teaching.



SAROJ KUMAR PANDEY received the M.Tech. and Ph.D. degrees from the National Institute of Technology, Raipur. He is currently working as an Assistant Professor at GLA University, Mathura. He has a several years of teaching and research experience in various technical institutions and university. He has published many research papers in various international journals, conferences, and book chapters. His research interests include deep learning, soft computing, and biomedical signal processing.

...