Intelligent Virtual Assistant Devlopment Using Python

Kushagra Sharma
SCSE, IILM University
Greater Noida, India
as1908296@gmail.com

Dubey Shivam Jagdamba Prasad
SCSE, IILM University
Greater Noida, India
dubeyshivam2002rd@gmail.com

Pulkit Dwivedi
SCSE, IILM University
Greater Noida, India
pdwivedi1990@gmail.com

***Abstract*—This paper will brief the education sector about how to use chatbot in helping students make proper decisions related to their career choices, thus saving them from suicidal activities. It can be done through creating awareness and providing the right consultancy to students about choosing the right kind of career, which will eventually enable them to reach the correct decision instead of being misled to follow the wrong type of career, which might lead them to depression and stress levels. We have developed a preliminary model of natural language processing, which might make human interaction and career counseling more involving with students and educational resources. The present paper aims to present the AI-based chatbot system deployed with deep learning techniques and a feed-forward multilayer perceptron in guiding students toward fulfilling career decisions. Therefore, our analysis was presented by the gap that existed with in the existing methods that lacked practical guidelines and personalized recommendations for career counseling. Additionally, the paper includes the comparative study which proclaims the time efficiency and accuracy of our proposed model. The minimum loss which could be recorded with in this paper was 0.0984% and the maximum accuracy that could be achieved from our model is 0.9576%. This research propounds concepts explaining functionalities and applications relating to career guide chatbot. This article addresses further issues of these new technologies regarding their possible connections with student suicidality. According to the authors, findings in this research will facilitate the researchers in gaining a better understanding of how such technology is actually designed and used because the time being, as of today, student wellness and academic results are undergoing constant improvements.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

A student is given a lot of choices in today's modern educational scenario. These choices will play critical roles in deciding his future course of action both in academics and professionally. Competing for these choices and fear of an irreversible wrong decision generate a lot of stress and apprehension very often. The increasing number of student suicides world wide points towards a serious and systematic lack of pipelines for providing enough guidance and support through these critical decision junctures. Traditionally, students have sought advice from guidance counselors, classroom teachers, and family members. But these are rarely available, as most schools, particularly those in less fortunate circumstances and those located in the country side, do not retain enough professional help. High student-to-counselor ratios even in well-resourced environments leave many students without the personalized support likely to be provided for them to adequately negotiate the complex worlds of academic and career decision-making. For the same reason,This proposal talks about the development of a system that would be termed the Chatbot Intelligent Virtual Assistant Development using Python. The AI tool is going to be named Friendly, and it will help the students be very well informed about the making of proper career choices and steer them in the right course toward higher education. In other words, Friendly will be there as an all-time coach to guide and help them in knowing options and making their choice that aligns them with aspirations. The AI chatbot can scale; hence, the system is available from any point at any time. Here, an AI-based chatbot following the steps of a virtual counselor will be developed using sophisticated NLP techniques for understanding and responding to the student's queries conversationally. This chatbot system is based on Python because of its wide libraries and frameworks that make the inclusion of machine learning algorithms possible and are a critical part of making this chatbot learn and improve constantly. The system uses Python-based tools for its functionality, including NLTK for natural language understanding, TensorFlow for machine learning, and Flask for web deployment. This technical framework allows for processing and analyzing the inputs of the student in order to be linked to the respective data and originate a response that allows for making a decision by students. The Chatbot Assistant System has been developed to provide the right and individualized counseling and to find crisis-in-the-making emotional distress signals provoked by a student. In this way, the chatbot working helpfully—without being critical—with the students will encourage them to do systematic work toward career possibilities, considering options, weighing the pros and cons of the possible choices, and laying out realistic goals. But the approach by means of AI counseling overcomes the weaknesses of the traditional guidance system and proactively finds solutions to reduce suicide risks among students. The Chatbot Assistant System will let the students make appropriate decisions that are in line with their interests, strengths, and goals through a reliable, approachable, and sympathetic helping hand given to the students. The paper discusses the design, development, and probable impacts of the Chatbot Assistant System. Also, the second part of the paper discusses some of the existing tools and approaches used for student advice and identifies the vacuum space that this system is going to fill. We will look into the technical aspects of the chatbot, including artificial intelligence and natural language processing technologies powering its operations. The architecture and workflow of the system to be developed are elaborated in the following sections. Following this, preliminary experimental results and user feedback will indicate the effectiveness of systems. In this paper, the most dramatic implications of such systems in terms of ethics and future research directions that could allow such growth are taken into consideration. The paper is organized as follows: Section II provides a review of pertinent literature related to AI-driven student counseling systems. Section III elaborates on the technical background of the tools and technologies used for building the chatbot. Section IV details how the architecture and workflow enable the chatbot to process and answer the questions asked by the students. Section V: Results explain the performance and potential effect the chatbot might have on student decisions. Section VI demonstrates the application of the system and its strengths for students and educational institutions. Finally, Section VII concludes the paper, summarizing the major findings and suggesting some directions for further research.

II. Related Work

E. Tebenkov et al. [1] have carried out a detailed survey of the important issues within the scope of the chatbot, which are knowledge management, techniques for response generation, natural language processing, frameworks of the machine learning model and strategies for using and testing datasets. K. H. Koundinya et al. [2] discussed the use of AI and ML to enhance the efficiency of chatbot for services like accessing college websites using the approach of finding a best response by mapping input with a closely related statement followed by choosing one from the predefined set of responses for that match. M. M. Hossain et al. [3] discussed their research on the design, development, and evaluation of a career assistant chatbot, enabling users to ask personal queries related to the education sector and obtain their career guidance. The system also proves to be accessible from various departments. Any user can access the information without going in search of a physical assistant. F. Mehfooz et al. [4] discussed the implementation of a retrieval-based chatbot with text support. Existing chatbots were discussed and how such systems help students and others in searching for critical information related to career guidance and other pertinent concerns. M. Herriman et al. [8] discussed the functionality of the modern chatbot specifically designed to use institution-specific responses to answer questions on careers. The key requirements are unique response mapping, advanced contextualization, and dynamic validation, driven by content-focused industry leaders and developed in collaboration with machine learning and natural language processing companies. S. Altay et al. [6] did research with 701 French participants and proved that even for a short period of interacting with the chatbot makes users highly disposed to searching for solutions and helps to give positive opinion toward the service offered by chatbots. In a nutshell, it would help the service users to minimize their hesitations to find assistance related to suicide of students in their life, especially through an adequately prepared and updated regularly. P. Amiri et al. [7] have studied application cases of chatbot for the purpose of support in the health services of students during panic and depression periods. The authors used filters for abstracts and keywords from relevant articles, obtaining information about applications, usage context, and design techniques for the studies. M. Herriman et al. [8] and M. Almalki et al. [9] conducted a review with the aim of analyzing the current role of chatbot on student suicide in healthcare. They concentrated on classifying and identifying new career guidance technologies and applications and discussing the solutions to the related problems. M. Herriman et al. [8] and M. Almalki et al. [9] conducted a review with the aim of analyzing the current role of chatbot on student suicide in healthcare. They concentrated on classifying and identifying new career guidance technologies and applications and discussing the solutions to the related problems. M. Herriman et al. [8] interviewed 29 participants to analyze daily positive and negative experiences in relation to conversational agents (CAs). From the current user perception, the authors were able to identify key criteria that would inform future CA model design, thereby contributing to a better user experience by fine-tuning guidelines for efficient and seamless CA functionality. M. Herriman et al. [8] describe the development of a Student Career Guidance Chatbot, in collaboration with Verily, Google Cloud, and Quantiphi, which is a Google Cloud strategic partner. The authors explained that user interactions, for example, assessments regarding preventing student suicide, must be designed to adapt and update in line with the capacities, resources, and pathways of the existing education system so that it effectively conveys important information while managing limited resources. U. Bharti et al. [10] have proposed a conversational chatbot on Google Cloud Platform (GCP) for career guidance in India. The aim of the proposed chatbot is to enhance user access to education and career information by making use of AI to address the current demand-supply gap among student guidance providers. G. Battineni et al. [11] have created a highly advanced AI chatbot that can be put to the test by means of diagnostic technologies and allow the students to obtain speedy and timely interventions each time they encounter suicide cases. The virtual assistant can be helpful in monitoring and analyzing the suicide cases in terms of the students. M. Ciotti et al. [12] is gave really useful insights into various matters of career guidance, comprising statistical information about the number of students who choose this or that profession, popularity fields of science, and tactics for the support of making such decisions of students. Such an automated system of a chatbot should be helpful in supporting the student by showing him his personalized recommendations adjusted to the student's own interests and academic background within a clear and helpful structure of guidance. S. J. Daniel et al. [13] have discussed the problems of the students in career choices because they do not have much access to individualized guidance. The proposed chatbot assistant system will be designed to overcome the challenges and will provide students with individualized career advice so that they can explore and make the right choices for their unique strengths and interests. S. Majumder and A. Mondal et al. [14] have focused on the importance of chatbot in student career guidance systems. Their well-rounded research work covered issues like the complexities of career paths, inadequate access to personalized advice, and need for low-cost solutions among students, which can be addressed in real-time with the help of these chatbot. S. J. Fong et al. [15] discussed barriers students faced in making decisions about careers and proposed the possible way in which they could be overcome by means of self-directed career guidance, easily accessible information platforms; friendly chatbot technologies, and analytical decision support systems to provide individualized career choices. S. Chakraborty and L. Dey et al. [16] discussed the most critical attributes of AI methods, machine learning, and deep learning applied while making predictions regarding career progressions for the students. Such an AI-centric approach facilitates the application of the chatbot in making personalized suggestions to the students based on their academic records, personal interests, and prospective opportunities in the future. L. Dey et al. [17] developed various machine learning models that can predict students' career outcomes depending on their skills and interests and validated these by the users' feedback. Additionally, they developed an advanced version of the chatbot with functionalities such as visualized career paths and question-and-answer capabilities. S. Al-Imamy and Y. Hwang [21]discuss the implications of algorithmic information processing for the attitude and choice of the students in a career guidance system driven by AI. The findings show the significance of AI in decisions made by the students as a result of customized recommendations and guidance based on the students' profile of academics as well as their interests. D. Shin et al. [22] proposed a cognitive model where user interaction with the CCGB was analyzed, especially personalization and comprehensibility. The study illustrates under what conditions these factors affect student engagement and understanding in seeking career advice through the chatbot. T.N.K. Hung et al. [23] proposed an AI-driven machine learning model that predicts the possible effectiveness of various career alternatives and education paths for a student. This model speeds up the selection process of providing personal career suggestions such that time and efforts that take place in producing an output from the system minimize the response time in accurate career advice. N. Q. K. Le et al. [24] developed a deep learning model to analyze and predict students' preferences for careers. This model facilitates neural networks and highly specialized scoring matrices to enhance the capabilities of a chatbot in offering personalized career guidance. The model improves the quality of advice that comes from the education sector on informing the students' career choices by providing a scientifically accurate match of their skills and interests with a given occupational profile.

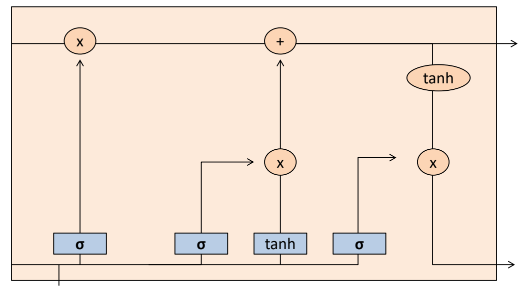

Fig. 1. The functioning of the LSTM Algorithm

III. BACKGROUND LSTM

Artificial intelligence and deep learning widely accept LSTM for its usefulness in solving complex problems, such as machine translation, speech recognition, and neural network-based input-output mapping. The four fundamental gates made LSTM learn any patterns very effectively compared to other architectures of the neural networks. All four gates - forget, input, output, and cell-play distinct roles collectively that form the basis for learning.

Forget Gate: The forget gate determines which information from the previous state needs to be retained for updating the current cell state and what could be discarded as irrelevant. It merges two inputs: the hidden state from the previous time step, ht-1 and the current input, xt, to determine which data should be carried forward.

It helps in determining which information should be held or forgotten. The corresponding equation contains the forget gate, which plays a big role in deciding to forget or remember.

$$Z = (u_1 X + \omega H_2) \tag{1}$$

Then the equation becomes :

$$F = \sigma(Z) \tag{2}$$

This equation represents the linear Combination

$$\mu_1 X + \omega H_2 \tag{3}$$

as Z, which might be useful for clarity in cetain context

where Z represents the linear combination of weighted input and μ_1 is the weighted associated with the input features. X, ω is the weight associated with feature H_2, H_2 is the other input feature or variable, F is the output after applying the activation function, and σ is the activation function applied to the linear combination Z.

Input Gate: The input gate enables the cell to update itself; it only selectively states which information needs to be remembered and which is not required. It filters out unnecessary data and selects relevant data to store in the memory of the cell. ht-1 and xt are processed, one through a sigmoid and the other through a tanh function, and the tanh regulates the amount of data passed into the network.

The input gate in a neural network, particularly in the LSTM (long-short Term Memory) network, helps control which value gets updated in the cell state. The input gate equation is typically expressed as:

$$I = 2H + \mu_2(X) + \omega \tag{4}$$

Similarly, I is the output of a function, which represents a combination of influences is the input features and bias. H is scaled by weight 2, meaning it directly and rigidly affects the outcome. where X contributes to the output and its influence controlled by the weight μ_2. The bias term ω adjusts the output independently of the input feature, providing flexibility. These elements from a linear combination that determines the final output I.

Output Gate: It therefore decides the new hidden state that should produce final output from the output gate. It takes inputs ht-1 and xt through a sigmoid function, with the updated cell state passed through the tanh function multiplied with the sigmoid output to calculate the output of the new hidden state.

The Output gate controls the final output given by:

$$0 = \sigma(1/2(\mu_4 X + omega_4 H_2') + 1/2(mu_4 X - \omega H_2')) \tag{5}$$

$$H' = Q.Sin(C)/Cos(C) \tag{6}$$

$$Output = \sum_i X_i.H_i' \tag{7}$$

In this equation, σ represents an activation function, while μ_4 and ω_4 are weights associated with the input features X, and H_2' is a transformed variable that calculates the trigonometric functions involving parameters Q and C. The final output is determined by summing the product of input feature X_i and their transformed counterparts H_i'.

Figure: 1 shows the functioning of different types of gates in LSTM algorithm.

Cell Gate: The cell gate is used to calculate the new cell state by integrating information that is available. The updated cell state is multiplied with the outputs of the previous steps, where it can reduce the values of the state if they are multiplied with some number close to zero.

The cell gate in LSTM Combine information from the forget gate and input gate to update the cell state. The standard equation for updating the cell state is:

The cell gate equation typically represented as:

$$C = C'.F + IG \tag{8}$$

Where C is the output variable, C' is the coefficient parameter, and F represents the factor that influences C. The term I is another coefficient, and G is an additional factor or variable to the contributing overall result. Together, symbols define how C is influenced by a combination of parameters and factor.

The time complexity of the chatbot interface is O(n), depending on how long it will take to load and format its screen. Similarly, the training algorithm as well as a response prediction mechanism of this system have a time complexity of O(n) while processing a query made by the user. This means that the building of the training model and the layered structure of the neural network all have a time complexity of O(4h(3d.h.d)), Here, d and h denote the layers of a neural network, which matches the expected time complexity of our proposed LSTM configuration.

A. *WORKING WITH LSTM*

LSTM works exactly as shown in Figure: 1 The inputs fed into the model get stored inside its neural system. Throughtout the training process, The model trverses -references The latest information alongsides already stored data before it finally generates the output. This algorithm works quite well as a classifier by filtering previously fed data from the past time steps and modifying the network's memory with the refined output. After going through the forget gate, that data goes to the output gate, where it prepares the prediction for the next time step. This mechanism improved the accuracy of the model along with optimizing space complexity. Fig.1 represent the used to explain how the components of the LSTM work.

B. *DECISION TREE MODEL*

A decision tree is predominantly applied to classification problems but can also be put into action in regression tasks. Internal nodes symbolize features of the dataset, a branch represents a decision rule, and a leaf node represents the outcome. Major nodes are: Decision that describes The way in which the algorithm functions and Leaf node that displays the outcome of all decision taking. Basically, a decision tree is an illustration of a set of possible solutions under certain conditions. The tree is arranged as having a root node with branches leading to leaf nodes. We applied the CART algorithm in drawing the decision tree in the following problem. The decision tree depicts or mimics the human decision-making process. It starts with the choice of a root node covering the entire dataset; by employing ASM, we choose the best subset of attributes of the dataset and partition it accordingly, recursively making nodes in the decision tree until the final solution is achieved. We used mathematical formulas like information gain, Gini index, and entropy for creating the decision model tree

$$\begin{split}\text{Signal gain} = \text{Entropy value (S)} - [\text{Weightes value})* \\ \text{Entropy (of each attribute)}]\end{split} \tag{9}$$

$$\text{Gini Index} = 1 - \sum_j P_j^2 \tag{10}$$

In information gain equation measures the amount of uncertainty that has been reduced after applying a feature. Thus, it is calculated as the difference between the overall entropy S of the dataset S and the weighted average of the entropies of individual features. Entropy S qualifies uncertainty in the dataset, whereas the weighted average considered the importance of feature contribution toward uncertainty. The Gini index measures the impurity given by 1-sum_j P_j^2, where P_j denotes the probability of class j. The lesser the Gini index is, the more purified is the dataset, which makes these metrics all-important while assessing the degree of effectiveness of features in the classification tasks. The decision tree model applies to our chatbot. For instance, when a user inquires about Student Suicide Cases and the information required is on Academic Interests, Skill and Confidence, Mental Health and Well-Being, Carrer Guidance, Mental Health Crisis, or Resources, the decision tree starts with the root node (Student Sucide Cases, determined by ASM). The subsequent progression then includes the decision nodes (Academic Interests, Skill and Confidence, Mental Health and Well-Being, Carrer Guidance, Mental Health Crisis, or Resources), which further splits into more nodes depending on the corresponding labels. Ultimately, the machine learning components reaches the last leaf node (Online Courses and Workshops). This method enables the chatbot to determine the relevant solution based on the user's query, such as identifying Career Option.

IV. Proposed Methodology

Whenever a user interacts with the graphical interface of the chatbot, the bot promptly respond to greeting or inquiries on it's end For instance, every time the user utters good morning or good evening, there will be a reply in the form of good morning, nice day. In case a user asks about the time, then the bot will inform the user about the time. The response of the chatbot is always appropriate whenever the user tends to ask some very complex questions or out-of-the-blue questions. For instance, if a user happens to say that today is not going well for me, a kind of feeling that has never been covered with in the training framework, the bot predicts a response similar to I'm here to assist you please share me your problem. The chatbot should also be enabled to identify voices so that users can communicate verbally instead of literally, along with the bot will respond both in voice and text. To support this functionality,We need to train the model regarding tags and responses. For instance, we can label a greeting with a response such as Morning greetings So, if the Chatbot identifies such an input again, then it will select a corresponding response. The training model is the heart and core of what makes this bot function. it defines how the bot will react to hundreds and thousands of questions or queries. This means that, on falling outside the training model, the bot will give a close approximation through structure analysis of the sentence, as well as a kind of recognition of word-meaning similarities. In the interaction with the user, the bot offers two interaction modes: text mode and voice mode.

Text mode: He types in his question; a prediction is made based on his model for training, and then with a response corresponding to that, the proper answer will follow. Voice mode Another important feature is voice mode, though in this mode, only the voice recognition module can be seen working, as it takes the words spoken out by the users and parses them into text, as the bot processes the query again but this time, it will be a spoken message to all of the answers written out. Upon finishing the interaction, the user has the option to keep chatting or Selecet the exit button to close the chatbot window. At any given time from then onwards, the bot can be restopped. The bot predicts and gives answers to queries of the user by using some algorithm that first helps recognize the kind of query and then responds accordingly with an accurate response.

Algorithm 1: In the algorithm we had created a JSON file which had tags, question patterns and responses loaded in the training file. So, a list is created to remove unnecessary words. Then the words are lemmatized and assigned to a new variable sorted in ascending order. Lastly, the data are written to a binary file 'words.pk1'. Scrambles the training list and splits it into two variables: 'train x' and 'train y'. Builds a neural network with layers of Fully connected layer and Random deactivation layer, Compiles the model and stores it as chatbot model h5. When invoked, given an input by a user, this model predicts the nearest response toward it. If the user prefers voice mode then it has converted the voice input to text and processes it as if it were a text based query. The sentences are still quite similar, but the entire text phrased differently now streams and flows better and more clearly.

A. *ALGORITHM 1: Pseudocode for the proposed Chatbot System*

Step 1: In the JSON data file, therefore, there would be three main elements: tags, patterns, and responses. Tags are categories or group together any type of user query this is highly beneficial for the chatbot in organizing and knowing what type of question one is asking. Patterns refer to the various ways the user could ask a question, and then the chatbot knows how a question may be written in several ways. Responses refer to the answers that the chatbot is going to output once these patterns are detected. These components of forms enable the chatbot to classify, understand, and then give the appropriate responses to the user inputs.

Step 2: At this step, we would import our JSON file holding the intents. We are going to load the JSON file via Python's json module reading in data to a format from which the chatbot can work. The syntax for this step would be as follows: Here, the "intents.json" is only a reference name for the JSON file containing tags, patterns, and responses. The one, which will be loaded to the variable intents that the chatbot later on will refer to, in processing queries in the code below.

Step 3 In this step, the input given by the user is freed from extra punctuation marks and other special characters. This will let the chatbot focus on the content of the query rather than its useless characters.

This is the list of punctuation marks that we want to delete. Then we update this list with any other characters we may wish to filter away. We use this list in processing the input

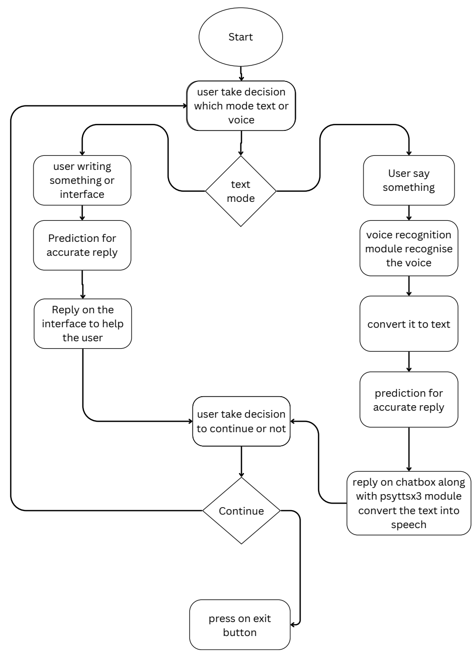

Fig. 2. flowchart of the overall chatbot system.

from the user, filtering away these characters from the input before we try to match the query with the patterns of the chatbot.

Step 4: Here, we will also fetch all words from patterns and tags generated with in our JSON file, then lemmatize words so they normalize. Lemmatization brings a word back down to its base, or root form. This helps the chatbot identify different forms of a word; for instance, run becomes running.

It iterates through each word, removes unwanted characters from the ignore letters, then it applies the lemmatizer so that all the words are finally in their root form. So, all the words work correctly and fit well into the brain of the chatbot.

Step 5: Sort the list of words that we obtained and remove all the duplicates so that the model will only work with unique words. And this is due to processing redundancy avoidance to retain perfect flow of understanding on an efficient pace of the chatbot.

In this code, set(words) removes any duplicate words in the list, and sorted() sorts words by letter, from A to Z. Thus, our final list is free of duplicated words and also in alphabetical order, ready to be passed onto your chatbot for more processing.

Step 6: Here, we are creating the dump files to save the processed data like words and classes based upon the usage of the Pickle module. This will enable us to save the data in such a way that later on we can use it, and it also will save some reprocessing time when our chatbot is running.

- The pickle.dump() is used to save the data to a file.
- 'words.pkl' and 'classes.pkl' will be the storing places for words and classes lists respectively.
- 'wb' stands for write binary, the mode in which the files are opened to write.
- This step ensures that you can load the processed data quickly in the future without having to run the code to calculate

Step 7: Here, We get our training data ready since you're going to create a bag of words model for each pattern along with the respective class label. This is achieved through the following sub-steps:

- Blank List: We have a blank list to store the training data as follows:
- Lemmatize and Lower case Each Word: Normalize each word in the word patterns by lemmatizing and lowering the case of all characters in the word.
- Build Bag of Words For each word in words-our sorted list from Step 5-check if it's in word patterns. If it is, append 1 to the bag; if not, append 0. This is the binary representation of the pattern based on the words it contains
- Output Row Create an output row as an empty list and set the position of the class (tag) of the current document (query) to 1. This creates a one hot encoded list where only the index corresponding to the correct class is set to 1.
- Attach the Bag and Output Row to the Training List Finally, attach both the bag (the binary word vector)

and output row (the class label) to the training list. This associates the input data: the bag of words, to the corresponding output data: the class label.

At the end of this step, our training list will include input-output pairs that can be used for the training model of a chatbot. The said list will form a set of pairs consisting of a binary vector representing the presence of specific words in the pattern and a one-hot encoded vector representing the class, or the tag, associated with that pattern.

Step 8: Shuffle the data in the training list, so the order of the input-output pairs is randomized. This would avoid the model from being trained on any patterns related to the order of the data rather than its actual content.

This will shuffle the training list in place, so that input output pairs are in random order. This helps model generalization better and prevents over fitting based on sequence of data.

Step 9: Here, we are going to divide our training list into two lists: trainx for the input feature and trainy for the output labels.

The list of all input feature vectors (bag of words representation) goes into train x and the corresponding list for the output labels-one-hot encoded class labels-go into trainy. This will enable the model to understand the relationship that should be established between the inputs and outputs during the training stage.

- train x holds all the input features, which are stored at index 0 of each training row.
- train y holds all the relevant output labels, which just so happen to be stored at index 1 of each row in the training data.

This separation is important, because during training, input and output data do have to be separated treated.

Step 10: We are going to implement here a neural network model in which Dense and Drop out layers added to it. This kind of structure will help make a model that can learn well from the training data and avoid over fitting

Here is how We can arrange the neural network with sequential model.

- Sequential Model: It creates linear stack of layers, which makes easier to build the model layer by layer.
- Add a Dense Layer: A dense layer is a fully connected layer. So, the neurons here are going to receive input from all the neurons of the preceding layer. We will add number of neurons and input shape here.
- Add Dropout Layer: The Dropout layer randomly sets a fraction of the input units to 0 at each update during training time, which helps prevent over fitting.

Step 11: We will now compile our neural network model and fit it with our training data. After that, We will save our trained model as an .h5 file for later use.

Here's how We can do it:

- Compile Model: At this point, We are required to mention the optimizer, the loss function, as well as all the evaluation metrics you desire.
- Fit the Model: Here, fit the model to our training data. We have to pick up the training data and the labels; epochs and batch size should also be specified.
- Save the Model: Finally save our fitted model to a file and so you may load the model later without having to retrain.

Step 12: Once the model has been satisfactorily trained We can add the functionality that determines the intent of a user's messages when the user is inputting into the chatbox. This is how the chatbot can respond based on what a user has input to the chatbox.

Here is how We add the predict step:

- Pre process the User Input: We preprocess the user input just as We pre process the training data. That means, We tokenize, lemmatize, and conversion to bag of words.
- Making Predictions: We would use the trained model and predict on the processed input based upon intent.
- Getting the Predicted Class: We identify the class which has the maximum probability using the output of the model.

Step 13: Use insert to add the response to the chatbot display.

Step 14: If We want to add voice response to our chatbot, We can use pyttsx3. The pyttsx3 library can convert text responses to speech. This is really useful when the purpose is a more interactive and user-friendly usage experience. More so, it'll be particularly useful for users who need auditory feedback.

Here's how We can do this for voice response functionality.

- Initialize Text-to-Speech Engine : Import pyttsx3 and initialize the text-to-speech engine.
- Play response generated by chatbot using the engine.
- Run Engine: This will execute the voice command and play the speech.

Step 15: In order to make the chatbot execute and respond back to user input until and unless the user decides to quit, We would include a loop, making the interactive session keep flowing. It will keep on asking for the inputs, process them, and come up with appropriate responses until the user clicks the exit button or intends to quit the conversation.

This is how We do it:

- Create a Loop: Using a while loop which will always prompt the user to input something.
- Check for Exit Condition: If the user wants to exit the program and types in a command word (such as "exit" or "quit"), it breaks the loop and the program terminates.
- Process User Input: This involves to accept the user input within the while loop, come up with responses to this, and may be give out voice output.

Step 16: When the user clicks the exit button, the application must close gracefully,terminating all process and releasing any resource it has used. This ensure that the system is prepared for future sessions without any lingering active process.

V. Experimental Results

A. Description of Dataset

The data set is provided in JSON formate was included in our model. It has been applied with a large number of keys important for recognizing patterns and correlating inputs to outputs, mainly in questions and responses. To differentiate sections with in the dataset, a parent key 'tag' was employed. In addition to the issues and problems a student would be facing, each key has text features associated with them, such as career options, mental health issues, and stress to conform to the academic ideal. The dataset also offers support resources in the form of hot line numbers, counseling services, and near by educational institutions. In addition to that, it directly guides the students to such resources through the chatbot and connects them with mentors and counselors and with the peer support groups. It therefore helps in setting the right career path for the students, meanwhile allowing the benefit of emotional and mental health concerns, which eventually reduces suicidal risks by students facing immense educational stress.

Some keys are categorized into parent child groups as shown in the Table 1, with the number of keys exceeding 20. The child keys hold text values that can be used to answer many queries of students data. For instance if a user types in a question, the chatbot identifies the appropriate pattern associated with the provided keys. This process is supported by a developed model of the neural network that makes use of the pattern keys to assure effective response generation. The method at this point in time, enhanced the capabilities of the chatbot in comprehending and interpreting questions by students along with delivering tailor-made career guidance and support for the students. In the process the chatbot reduces stress being exerted on the students and, positively adds up to preventing cases of suicide which later becomes an aftermath of stressors education-related.

In Table 2 main model parameters As explained, dropout has been used at training for hidden layers with a rate of 50 to improve the accuracy of the prediction for the network and get a robust neural network. For the initializer of the kernel and initializer of bias, uniform and zero values have been defined to limit the bias in the model. Thus, such a design is thoughtful since it helps the chatbot actually go on and provide career guidance to the students while being supportive in minimizing the effects of the education system that contribute to mental health problems.

As discussed earlier about the model architecture and software configuration of the hardware, the application goes well with an extensive variety of operating systems and hardware. Its lightweight and efficient nature lets it work with almost every type of device, right up to mobile applications where it works perfectly.

To demonstrate the user interaction of the bot Some of the screenshots illustrating how users interact with the bot are placed below. Figs 3 to 4 demonstrate how the bot initiates the process of engaging the student on matters concerning careers, leading the students in solutions toward what might be of concern to them at their academic levels. This interaction is essential as it ensures the way forward for the students as they try navigating through their educational journey without mental health breakdowns due to stress with academics.

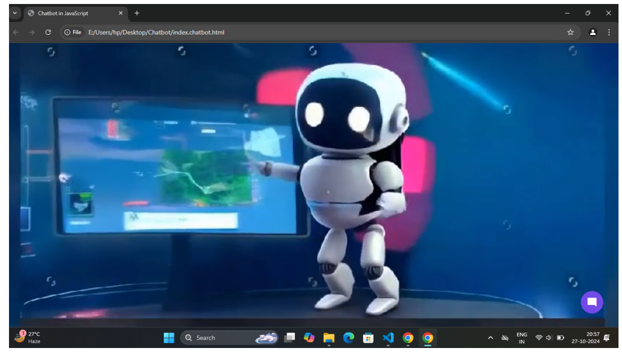

Fig. 3. Show the bot Interaction

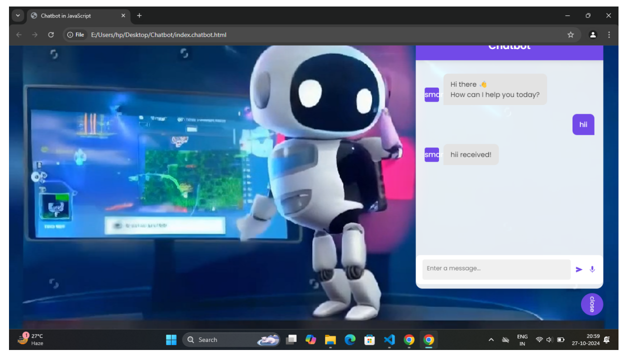

Fig. 4. Introductionary Interface to the user along with the chatbot interface system

B. Training and Evaluation Phases

After Loading the JSON file and constructing the training model the chatbot is capable of making accurate prediction for the user. But before integrating it into the core file, testing of the phases of training and testing is a must in order to validate the correctness of the model and loss percentage. Below is the table of accuracy and loss percentages from the learning model.

As shown in Table 1. the data illustrates the accuracy as well as the loss changing with each epoch based on the training model used. In this regard accuracy becomes better with the increased number of epochs whereas the loss will be lesser.

It means the predictions of the bot will increasingly be right over time. On the other hand, if the accuracy is low or unstable, it may precipitate problems in the response by the bot. The outcome may be unexpected or wrong replies by the bot every now and then when users come up to request queries or questions. The best accuracy achieved appears at epoch 150/150 in the table. Loss has bottomed at 0.0984%, with an accuracy of 0.9576%.

The achieved results are based on the algorithm used being the LSTM model which consists of four core parts.

TABLE I
DATASET KEY AND CONTAIN DATA TYPE.

Key Name	Type	Contain Type
Career_Choices	Parent	Recognition Pattern
Counseling_tips	Parent	Recognition Pattern
Student_responses	Child	Shuffling Answers
Guidance_patterns	Child	Shuffling Questions
Mental_health_tips	Parent	Recognition Pattern
Success_stories	Child	Shuffling Answers

TABLE II
PROPOSED DEEP LEARNING MODEL ARCHITECTURE FOR THE ASSISTANT CHATBOT SYSTEM.

Layers	**Units-Neurons**	**Activation Function**	**Optimizer**
Input Layer	256 units	-	-
Hidden Layer 1	128 units	ReLU	-
Hidden Layer 2	64 units	ReLU	Categorical Crossentropy and Adam
Output Layer	3 units	Softmax	-

TABLE III
ACCURACY OF THE ASSISTANT CHATBOT TRAINING MODEL.

Training Model	Epoch	Batch Size	Loss	Accuracy
chatbot_career_guidance_model.h5	150/150	20	0.0984%	0.9576%

In general the LSTM model includes four gates as part of the model: forget gate, input gate, output gate, and cell gate. Each performs a unique function in managing information flow through the network.

All the gates are combined into an optimized model of training with varying loss and accuracy. The accuracy is improving constantly with epochs. For epochs 1/200 to epoch 3/200 accuracy is constantly increasing. However for epochs 5/200 to 7/200 accuracy remains stable. But after epoch 9/200 it increases with every epoch drastically. As shown in figure 1, output gate and cell gate can refer to the dynamics of how the bot begins to communicate with the user. For example in this case as the training model increases with each epoch so do the responses given by the bot which become even more accurate and relevant. On the other hand if accuracy lowers so do the responses which may become loose and irrelevant. Therefore the entire training process should display as much precision as possible to better and more closely predict it. Higher accuracy would therefore be able to possess better functioning neural layers that should serve as a key component in the development of an excellent chatbot.

Here the model, though hidden, encoder-decoder configuration plays a significant role to achieve high accuracy. Information flows from one layer to another by repeating the vector layer. A wrapper of bidirectional is applied on the LSTM layers so that both loss and accuracy are calculated. This improves the overall performance of the model while analyzing its direction.

Here are the lines provided with an alternative version: As can be seen in Table 4, the highest accuracy reached in testing this model is 94.1%, which occurs at phase 2. This indicates that once the model completed phases 2, while continuing further into those phases, the model reached peak accuracy, meaning the bot could indeed produce the most pertinent and relevant responses.

In the initial tests, the responses returned to the chatbot will neither be easy to the system nor highly relevant and precise for the users. Of course, there is little accuracy to know what could later materialize. Thus, this is a particularly crucial testing stage. It is until we test them appropriately that we cannot assess how good it is to produce relevant responses or up to how long it continues to have a conversation going on. This may make the bot poorly experienced for the user, some what dampening its use and possible benefits later in interactions and as a way to meet better expectations from users. Thus, the testing phase is quite crucial and must be done with precision. In addition, its accuracy improves if the model is tested continuously. The more phases of testing we have, the closer we get to the real accuracy than what was obtained in phase 4. In other words, the more testing we do, the more we will be able to fine-tune our model to achieve higher accuracy. We got, Let T represent the testing phases and denote the accuracy. We can express their relationship as:

$$A = nT \tag{11}$$

where n is a constant, which denotes the proportionality factor. This expression means that the accuracy. A is proportional to the number of testing phases T, with n being the scale of this proportionality.

Since the number of testing phases and precision are inversely proportional, more testing phases imply better precision. Greater the number of testing phases, greater will be accuracy.

This means that if we are to get more accurate results, we should perform many testing phases. Ultimately, this leads

TABLE IV
ACCURACY TESTING PHASES OF CAREER GUIDANCE CHATBOT MODEL.

Model Name	Testing Phase 1	Testing Phase 2	Testing Phase 3	Accuracy
chatbot_career_guidance_model	50.2%	83%	96%	85.6%
chatbot_career_guidance_model	52.2%	85.6%	94.1%	94.1%
chatbot_career_guidance_model	72.6%	95.2%	99.7%	90.1%

to better models, and users will be getting the best possible response from the chatbot.

In the appearance of general accuracy for testing cases. It can be noted that performance in the case of testing depends on general accuracy. Thus, if general accuracy improves above the previous levels, then probably testing cases are likely to have good performance. Good predictions will be generated in that case. In contrast, if it doesn't improve above the previous levels of accuracy, then the performance of testing cases will fail to be what was expected.

All test cases will yield different results, and selecting a training model is based on test cases that result in improvement, as seen in the graph. Each layer of the model consists of high-quality data resulting from the optimization of the training process. Consequently, as the training model and its layers continue to advance in their competency, the outcome tends to be user-friendly and workable for meaningful interaction because it will offer relevant information to the user.

C. Analysis of Time Complexity

The user interface of the chatbot as created using the library. Time complexity for the loading up of the library and designing the optimal layout of the chatbot is O(n). In a similar way, the time complexity to train the model and provide the best possible predictions regarding queries of the user also falls with in the range of O(n). If we build a training model and integrate several layers into a proper neural architecture the time efficiency would be represented by O(4h(3d+h)), where d and h denote the neuron counts in the layer of the neural network. Applying this proposed time complexity specifically is to the LSTM algorithm, so it follows that the overall time complexity, that includes the training model, can be represented as: O(n(4h(3d+h))) This account is for both the layout design and the prediction of replies.

The total time complexity is:- O(n)

The Space Complexity is:- O(n)

Table 5 represented after using different methods about our chatbot are developing the learning framewrok by utilizing various approaches, We notice that each has different accuracy. In this case, considering the time complexity both in testing and training approaches, when we make use of the LSTM with just one layer. Additionally, the highest accuracy is found with the LSTM model.

considered an advanced model of sequence using various gates and gradients as that of a neural network. These types of models process sequential inputs and perform all the operations necessary to generate accurate output. And then, the decision tree creates several nodes and sub-nodes where in each node holds data for user applications. For the test cases and train cases, LSTM had the highest accuracy, with the following being the decision Tree.

TABLE V
ACCURACY AND COMPLEXITY COMPARISON FOR ASSISTANT SYSTEM CHATBOT

Method	**Accuracy**	**Time Complexity**
LSTM	0.9450	O(n)
Decision Tree	0.6753	O(nkd)

We observe that accuracy vary across different methods. For the approach discussed LSTM is as accurate as it can be compared to other methods. So, the decision-making tree method exhibits the minimal accuracy with in those considered approaches. So for our designed model and training methodology, LSTM is found to be the best-suited approach. The decision tree method might still be good for other models; however, in this approach, LSTM offers the best.

However, since the amount of accuracy obtained from LSTM is much higher, we opt to construct our neural network layers with the LSTM approach and design our training model accordingly. This will ensure the best performances in terms of user interactions, as appropriate and expected responses will be delivered by the bot.

The reflects an overview of the existing chatbot approaches. From this comparison, it is quite clear that our proposed model holds the position of being more effective inter-user communication facilitators. Also, the responses from our model are relatively faster, more accurate, and more easily understandable than the other approaches.

VI. APPLICATIONS

A. PERSONALIZED CONVERSATIONS

Friendly was built to interact with users in highly personalized, naturalistic conversations that are empathetic in language and tone, as every interaction builds on the previous topic to create a conversation that runs smoothly and is enjoyable, all the while providing relevant information. This aspect of personalizing word choice to the best for emotional resonance and desire fits into a world of excitingly accommodating supportiveness in the pursuit of meaning relationships with users.

B. OMNI-CHANEL

Our chatbot, Friendly, is a robust resource support tool for our students as the students avail career paths information and support while being helped overcome academic challenges. By gathering knowledge from credible sources with in education and occupational groups, Friendly actually serves value information on specific areas of concentration as well as opportunities in advancing careers. The chatbot will, with information from the users on their different areas of interest as well as academic backgrounds, give them personalized solutions such as links for them to access resources and organizations, which can give them much more relevant support for informed decisions concerning their careers. Lastly, Friendly tries to guide students on the way of education and avoid forces that may even push someone into mental health crises and suicidal thoughts.

C. ACCESSIBILITY

Friendly has an interface that can be accessed by all, and it is not even bound by the platform that a student is on. A stable internet connection will be enough for the student to enjoy its features. The students who will require auditory assistance will find it very resourceful since it reads out information for such visually impaired students. They will be able to question the chatbot verbally in regard to questions about their career, and Friendly will return with all information and leads that may prove critical in supporting them through their education and careers. This feature of Friendly ensures that all students have such support to enable them in making decisions and gaining advice to minimize mental health issues.

D. CONTEXTUAL ASSISTANCE

The Friendly Bot is a good source for education institutions and organizations to seek contextual help because it can provide the latest information regarding career pathways and student support. Generally speaking, the chatbot can only serve up a solution by learning the common concerns from student inquiries. This feedback cycle makes it possible for teachers and counselors to understand the challenges students face, there by streamlining efforts at implementing the necessary improvements and resources intended to help students make well-informed choices for their careers while at the same time preventing mental health crises triggered by educational pressures.

E. BOOKINGS

The Friendly chatbot is designed to provide information on different types of careers and education sources for students seeking to find their way. Users will be able to get details on educational institutions available in the area, scholarships, and career counseling services, thus allowing them to broaden their options with minimal research needed. The chatbot will provide access to booking appointments with career advisors and counselors, enabling advice and consultation to reach student's homes comfortably. By providing direct contact details for the above resources, Friendly removes unnecessary barriers in a student's pursuit of the necessary assistance in order to help guide them through educational paths and inform decisions about their future careers.

F. CONVERSATIONAL SERVICES

The Friendly chatbot will give students an interactive interface with a personalized guide and support on the education journey. It can identify the challenges of the users and propose suitable ways based on their choices and goals by engaging in a conversation with the users. Friendly will also create awareness about the mental health imperative and the dangers of academic stress, urging students to help themselves when they need it. This would make the chatbot proactive, thus optimizing its response so that at the end of it all, the student gets the right kind of information and help to take the right decision and reduces the chances of feeling overwhelmed by having options that scatter their career choices.

G. STUDENT ASSISTANCE

Friendly is a chatbot that one can use to research career pathways or the impact of educational technology; such information will provide the required skill and qualification frameworks for different fields. This will help students know how to make proper decisions in their academic or career choices. The chatbot, by maintaining a stable and engaging conversation, will enable students to get the right information relevant to their queries, thus enabling them to make informed decisions and find the right direction in their educational journey. This method will not only help in working out long-term career development but will also help create a supporting environment and encourage these students to seek guidance and prevent feelings of uncertainty or despair.

VII. BENEFITS

The Intelligent Virtual Assistant Development Chatbot is the interface, offering personalized, 24/7 advice on career choices to students who could otherwise create uncertainty and anxiety in making decisions. It links what a student can best do and enjoys doing with career possibilities thus preventing wrong career choices that cause undue pressure associated with most academic decisions, there by affecting suicide prevention, an issue possibly caused by stress resulting from wrong career advice. This chatbot is scalable, cost-effective, and provides educators with valuable data insights that have equipped it with being a comprehensive tool for career guidance and mental health.

VIII. CONCLUSION

Therefore, it would be fair to mention that creating an IVA using Python is the gigantic leap forward to the overcoming of hindrances students face to make proper career choices. That is, the chatbot guides and supports students with advice through out the process of making smart decisions on which path to take at school in order to avoid the wrong choice made concerning their career. Moreover, this proactive approach not only helps in career planning but also plays a very critical role in suicide prevention. The immense pressure that the students are kept under and which often takes extreme steps makes it rather a dire need. It is something that can really be scaled up, accessed, and brought to any number of students at the most cost-effective prices. With this in mind, this chatbot might help students positively in their lives, as it will orient or reduce the stress in students' minds while giving a lifeline to those on the verge of committing life-changing mistakes in academic journeys.

REFERENCES

[1] E. Tebenkov and I. Prokhorov, Machine learning algorithms for teaching AI chat bots, Proc. Comput. Sci., vol. 190, pp. 735744, Jan. 2021.

[2] 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. 15.

[3] 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. 5873, Dec. 2021.

[4] 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. 423430.

[5] 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. 113, Jun. 2020.

[6] S.Altay, A.S.Hacquin,C.Chevallier, and H. Mercier, Information deliv eredbyachatbothasapositiveimpactonCOVID-19vaccinesattitudesand intentions, J. Exp. Psychol., Appl., vol. 27, pp. 111, Oct. 2021.

[7] P. Amiri and E. Karahanna, Chatbot use cases in the COVID-19 pub lic health response, J. Amer. Med. Inform. Assoc., vol. 29, no. 5, pp. 10001010, Apr. 2022.

[8] 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. 113, Jun. 2020.

[9] M. Almalki and F. Azeez, Health chatbots for ghting COVID-19: A scoping review, Acta Inf. Medica, vol. 28, no. 4, p. 241, 2020.

[10] U.Bharti,D.Bajaj,H.Batra,S.Lalit, S. Lalit, and A. Gangwani, Medbot: Conversational arti cial intelligence powered chatbot for delivering tele health after COVID-19, in Proc. 5th Int. Conf. Commun. Electron. Syst.(ICCES), Jun. 2020, pp. 870875.

[11] G. Battineni, N. Chintalapudi, and F. Amenta, AI chatbot design during an epidemic like the novel coronavirus, Healthcare, vol. 8, no. 2, pp. 18, Jun. 2020.

[12] 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. 365388, 2020.

[13] S.J.Daniel, Education and the COVID-19pandemic, Prospects,vol.49, no. 1, pp. 9196, 2020.

[14] S. Majumder and A. Mondal, Are chatbots really useful for human resource management? Int. J. Speech Technol., vol. 24, no. 4, pp. 969977, Dec. 2021.

[15] S. J. Fong, N. Dey, and J. Chaki, AI-enabled technologies that ght the coronavirus outbreak, inArti cialIntelligence for CoronavirusOutbreak. Singapore: Springer, 2021, pp. 2345.

[16] S.ChakrabortyandL.Dey, TheimplementationofAIandAI-empowered imaging system to ght against COVID-19 A review, in Smart Health care System Design: Security and Privacy Aspects. 2022, pp. 301 311.

[17] L. Dey, S. Chakraborty, and A. Mukhopadhyay, Machine learn ing techniques for sequence-based prediction of viral host interactions between SARS-CoV-2 and human proteins, Biomed. J., vol. 43, no. 5, pp. 438450, 2020.

[18] 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. 2122321240, Mar. 2022.

[19] H. Sharma and A. S. Jalal, A framework for visual question answering with the integration of scene-text using PHOCsandFishervectors, Expert Syst. Appl., vol. 190, Mar. 2022, Art. no. 116159.

[20] 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. 16.

[21] S. Al-Imamy and Y. Hwang, Cross-cultural differences in information processing of chatbot journalism: Chatbot news service as a cultural arti fact, Cross Cultural Strategic Manag., vol. 29, no. 3, pp. 618638, 2022, doi: 10.1108/CCSM-06-2020-0125.

[22] D. Shin, The perception of humanness in conversational journalism: An algorithmic information-processing perspective, NewMediaSoc.,vol.24, no. 12, pp. 26802704, Dec. 2022, doi: 10.1177/1461444821993801.

[23] T.N.K.Hung,N.Q.K.Le,N.H.Le,L.VanTuan,T.P.Nguyen,C.Thi,and J.-H. Kang, An AI-based prediction model for drug-drug interactions in osteoporosis and Pagets diseases from SMILES, Mol. Informat., vol. 41, no. 6, Jun. 2022, Art. no. 2100264.

[24] N. Q. K. Le and B. P. Nguyen, Prediction of FMN binding sites in electron transport chains based on 2-D CNN and PSSM pro les, IEEE/ACM Trans. Comput. Biol. Bioinf., vol. 18, no. 6, pp. 21892197, Nov./Dec. 2021.