

# **Improving Mental Health through Multimodal Emotion Detection from Speech and Text Data**

A project report submitted for the partial fulfilment of the  
Bachelor of Technology Degree in

**Information Technology under Maulana Abul Kalam Azad University of  
Technology**

by

**Adarsh Sarda**

Enrolment No:12019002004109

Registration Number: 014473 of 2019-20

Exam Roll No.: 10400219100

Academic Session: 2019-2023

Under the Supervision of  
Prof. Pulak Baral



**Department of Information Technology**

**Institute of Engineering & Management**

Y-12, Salt Lake, Sector 5, Kolkata, Pin 700091, West Bengal, India Affiliated To

**Maulana Abul Kalam Azad University of Technology**

BF 142, BF Block, Sector 1, Kolkata, West Bengal 700064

**May 2023**



**INSTITUTE  
OF ENGINEERING & MANAGEMENT**  
Salt Lake Electronics Complex, Kolkata - 700091, WB, INDIA

Phone : (033) 2357 2969/2059/2995  
(033) 2357 8189/8908/5389  
Fax : 91 33 2357 8302  
E-mail : [director@iemcal.com](mailto:director@iemcal.com)  
Website : [www.iemcal.com](http://www.iemcal.com)

## **CERTIFICATE**

### **TO WHOM IT MAY CONCERN**

This is to certify that the project report titled **“Improving Mental Health through Multimodal Emotion Detection from Speech and Text Data”**, submitted by **Adarsh Sarda**, Enrolment No: **12019002004109**, Exam Roll: **10400219100**, Registration No: **014473** of 2019-20, student of **Institute of Engineering & Management** in partial fulfillment of requirements for the award of the degree of **Bachelor of Technology in Information and Technology**, is a bonafide work carried out under the supervision of **Prof. Pulak Baral** during the final year of the academic session of 2019-2023. The content of this report has not been submitted to any other university or institute for the award of any other degree. It is further certified that the work is entirely original and the performance has been found to be satisfactory.

---

**Prof. Pulak Baral**  
Assistant Professor  
Department of Information Technology  
Institute of Engineering & Management

---

**Prof.(Dr.) Moutushi Singh**  
H.O.D.  
Department of Information Technology  
Institute of Engineering & Management

---

**Prof.(Dr.) Arun Kumar Bar**  
Principal  
Institute of Engineering & Management

# INSTITUTE OF ENGINEERING & MANAGEMENT



## DECLARATION FOR NON-COMMITMENT OF PLAGIARISM

I, Adarsh Sarda, student of B.Tech in the Department of Information Technology, Institute of Engineering & Management has submitted the project report in partial fulfillment of the requirements to obtain the above-noted degree. I declare that I have not committed plagiarism in any form or violated copyright while writing the report and have acknowledged the sources and/or the credit of other authors wherever applicable. If subsequently it is found that I have committed plagiarism or violated copyright, then the authority has full right to cancel/reject/revoke our degree.

Name of the Student: ADARSH SARDA

Full Signature: \_\_\_\_\_

Date: \_\_\_\_\_

# Contents

<b>1. Abstract.....</b>	<b>4</b>
<b>2. Acknowledgements.....</b>	<b>5</b>
<b>3. Keywords.....</b>	<b>6</b>
<b>4. List of Figures.....</b>	<b>7</b>
<b>5. List of Tables.....</b>	<b>8</b>
<b>6. Introduction.....</b>	<b>9</b>
<b>7. Related work.....</b>	<b>11</b>
<b>8. Basic underlying concepts.....</b>	<b>13</b>
8.1 Natural Language Processing.....	13
8.2 Long-Short-Term Memory.....	13
8.3 Word2Vec (W2V).....	14
8.4 F1 score.....	15
8.5 GPT-3.....	15
<b>9. Proposed Approach.....</b>	<b>16</b>
<b>10. Project Architecture.....</b>	<b>17</b>
10.1 Input and Preprocessing.....	17
10.2 Processing.....	18
10.2.1 Using Natural Language Processing (NLP).....	18
10.2.2 Using Long-Short-Term Memory (LSTM).....	18
10.2.3 Using GPT-3.....	19
10.3 Using Output.....	19
<b>11. Algorithm.....</b>	<b>20</b>
<b>12. Outcomes.....</b>	<b>30</b>
<b>13. Conclusion.....</b>	<b>35</b>
<b>14. References.....</b>	<b>36</b>

# 1. Abstract

---

In today's world of cut-throat competition, where everyone is running an invisible race, we often find ourselves alone amongst the crowd. The advancements in technology are making our lives easier, yet man being a social animal is losing touch with society. As a result, today a huge part of the population is suffering from psychological disorders. Inferiority complex, inability to fulfill dreams, loneliness, etc., are considered to be the common reasons to disturb mental stability which may further lead to disorders like depression. In extreme cases, depression causes the loss of precious lives when an individual decides to commit suicide. Assessing an individual's mental health in an interactive way with the core help of machine learning, is the primary focus of this work. To realize this objective, we have used the most suitable Long short-term memory (LSTM) architecture. It is an artificial recurrent neural network (RNN) in the field of deep learning on Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) and FastText datasets when fed with model-patient conversational data. Further, we discussed the scope of enhancing cognitive control capabilities over psychiatric disorders which may even lead to severe levels of depression and suicidal attacks. Here, the proposed system will help to track patients and aims to help health experts with better diagnoses. The AI-based chatbot will talk to the patients and help them reveal their thoughts, which they are otherwise not able to communicate to their peers. The novelty of this work is in the emotional analysis of voice chat, which therefore creates a comfortable environment for the user along with features for communication.

## 2. Acknowledgements

---

I must not forget to acknowledge everyone who has provided constant support to me during my B.Tech course. First and foremost, I would like to express sincere gratitude to my supervisor **Prof. Pulak Baral** for his continuous support and motivation in fueling the pursuit of carrying out this project endeavor. Without his guidance and persistent encouragement, this project work would not have been possible. He has been a tremendous mentor for me throughout this academic journey. Much of his academic advice about my career growth has been priceless.

I would like to convey sincere gratitude to **Prof.(Dr.) Moutushi Singh** for providing me with constant inspiration to stand firm against several setbacks throughout the course. Additionally, I would like to thank all the technical, non-technical, and office staff of our department for extending and facilitating cooperation wherever required. I also express gratitude to all of my friends in the batch for providing a friendly environment to work on the project work.

I would also like to thank my Director **Prof.(Dr.) Satyajit Chakraborti** for providing me with an outstanding platform in order to develop my academic career. In addition, I also preserve a very special thankful feeling about my Principal **Prof.(Dr.) Arun Kumar Bar** for being a constant source of inspiration.

A special thank is due to my family. Words cannot express how grateful I am to my parents for all the sacrifices that they have made while giving me the necessary strength to stand on my feet.

Finally, I would like to thank everybody who has provided assistance, in whatever little form, towards the successful realization of this project but an apology that I could not mention everybody's name individually.

### 3. Keywords

---

Keywords	Description
LSTM	Long Short-Term Memory.
RNN	Recurrent Neural Network
RAVDESS	Ryerson Audio-Visual Database of Emotional Speech and Song
GPT	Generative Pre-trained Transformer
W2V	Word to Vector
NLP	Natural Language Processing
Epoch	During machine learning training, an epoch is a single run over a complete dataset.
OHE	One Hot Encoding

## 4. List of Figures

---

Figures	Description
Fig 1.	Architecture of a Long-Short-Term Memory Network Model
Fig 2.	Schematic diagram of the multimodal motion detection system
Fig 3.	Project Architecture
Fig 4.	Tokenizing
Fig 5.	Padding
Fig 6.	Embedding in numeric representation
Fig 7.	Character embedded in Table
Fig 8.	Performing OHE on the dataset
Fig 9.	Importing the pre-trained words
Fig 10.	Training the model
Fig 11.	Confusion matrix with 30 epochs
Fig 12.	Taking user voice input along with the name
Fig 13.	Output Dataframe
Fig 14.	Setting up resources
Fig 15.	Interacting with the user to take voice input
Fig 16.	Recording Audio
Fig 17.	Using GPT3 to analyze the audio
Fig 18.	Sending the message using Twilio API



## 5. List of Tables

---

Tables	Description
Table 1.	Comparative analysis with some of the existing methods
Table 2.	Experimental results based on some random messages
Table 3.	Data stored in Excel
Table 4.	Message sent to specialist

## 6. Introduction

---

Mental health of an individual mostly depends on psychological, emotional, and social factors. It determines our ability to handle stress and make choices. Maintaining good mental health at every stage of life is important. However, instability affects our healthy lifestyle in terms of our way of thinking, mood fluctuation, and changes in behavior. Some biological factors, life experiences, and family history may also contribute to mental health issues. It is common but recoverable with proper and timely action. Generally, sufferers of depression tend to internalize their feelings and express hesitation with human interrogation. But with technologies advancing, people have now found a ground of comfort with general chatbots. Considering these accounts, our idea is to create an AI-implemented model, for better interaction and thus adding a benefit to both ends; detection and proper analysis. A user-friendly chatbot has been built on an AI model having detecting capabilities from Neural Networking, a close study of tracking a human brain. Further inclusion of voice support will get this model the extra mile with users. To measure the severity of depression, emotional analysis is essential. We have performed multimodal emotional analysis using AI-bot for necessary remedies. This would be an effective way to reach more people suffering from depression, as the patient won't have to open up about personal troubles to the unknown or have a fear of getting judged. A chatbot can simply lend a helping hand and heed the need. The work is focused on planning to execute the first and foremost thing needed would be the data collection and analysis for understanding human emotions. The most viable option was to get it by taking speech input from the user. AI-based models are then used to analyze the answers. Open-ended answers are preferable here rather than using the MCQ-based versions for more variations. The system has been developed for emotion analysis using human interaction. We implemented the LSTM architecture using the w2v method converting each word into a vector. Those vectors can be used to calculate the emotions further. Finally, to achieve the goal for sentiment and emotion analysis, we fed the RAVDESS dataset to our model which categorized the data into numerous classes along with five basic emotions namely; neutral, joy, sad, fear, and anger. In further advancements of the project, we have used GPT-3 which does not limit itself to the aforementioned five emotions only. It can detect emotion by recognizing voice, text, and social media-influenced language. In contrast, advancements in lifestyles and reasons have outnumbered depression reaching the heights of statistics. We took the challenge to contribute to the main-field research by

incorporating remedies in correspondence with situations. This work also considers specialists' recommendations and active communication with physical help centers.

The main contributions of the work are as follows:

- Multimodal emotion recognition via speech and text.
- Development of communication and updation with specialists
- Development of a support system to improve mental health.

Extraction of emotion from different social media is one of the trending works nowadays [16]. There are many state-of-the-art methods that support emotional detection [5,6]. The posts given on various social media platforms are being analyzed by the researchers to understand the current state of mind. The combination of speech, text, gesture, voice, EEG, and other biomarkers are also considered powerful parameters for emotion detection [12]. However, most of the works focus on a specific content in this case [14]. Voice frequency, modulation of speech, and tone were used in a study to detect emotion [17], [2]. Emotion detection from videos by watching gestures and facial expressions was done in another study [8]. Emotion models can be divided into two categories based on some existing theories. They are Categorical and Dimensional. Emotions that are discrete in nature, are enlisted into the Categorical models, and Dimensional category models depend upon some dimensions with parameters. The terms valence, which is the positivity and negativity of emotions, arousal, which is the excitement level of emotion, and dominance which is basically the level of control of emotions, are used in dimensional emotional models [13]. Lexicon-based emotion detection is another well-known method against a given input dataset [3], [11]. The model was rebuilt using GPT-3 technology, which allowed for the storage of emotions associated with the user's name and timestamp in an Excel file. If necessary, this information can be sent to experts to enable ongoing monitoring.

## 7. Related work

---

### A. Existing work

Table 1. Comparative analysis with some of the existing methods.

Name of the Apps from the Google Playstore	Objective questions	Descriptive	Voice Assistant	Interactive games	Awareness or community support
Moodpath	Yes	No	No	No	Yes
TalkLife	Yes	No	No	No	Yes
Daylio	Yes	No	No	No	No
Youper	Yes	No	No	No	No
What's up	Yes	Yes	No	No	Yes

- For those who want to keep track of and enhance their emotional well-being, there are many resources accessible that are related to mental health and well-being. One such tool is **Moodpath**, which provides a mental health examination over the course of 14 days. Through this resource, people can evaluate their emotional well-being and decide whether or not to seek professional help if necessary.
- Another tool that provides community support and interaction in a manner akin to group therapy is **TalkLife**. It offers a supportive network of thousands of individuals who are eager to communicate, listen, and make people feel less lonely while being anonymous. For those who might not have access to traditional therapy or who feel more comfortable receiving support from an anonymous group, this might be extremely helpful.
- **Daylio Journal** is a mobile application that enables users to graphically track their daily activities and moods, which might help them better understand their moods. On the basis of the data gathered, the app also produces statistics and trends, and reminders to assist users in maintaining their tracking and writing habits.
- Another smartphone application that offers tailored support depending on a user's responses is called **Youper**. The software guides users through any strategies they might require at any given time and distills their interactions

and chats into insights to help them learn more about their emotional health.

- Another mobile app, **‘What's Up?’**, teaches dozens of coping skills, such as breathing exercises and grounding exercises. It draws inspiration from cognitive behavioral therapy (CBT) and acceptance and commitment therapy. The program also aids users in identifying skewed mental processes like catastrophizing and binary thinking.

Automated voice response systems have been shown to be effective mobile health technology for addressing a number of chronic illnesses in addition to these sources. The ability to quantify symptom severity, provide self-management options, and offer electronic reminders and support through these call-handling systems enables people to enhance patient adherence to medication.

Overall, those who want to enhance their mental health and well-being can benefit greatly from using these tools. Individuals can choose the resource that best satisfies their requirements and tastes thanks to the diversity of possibilities accessible.

## **B. Literature Survey**

- **Adrian Rosebrock's "Emotion Detection Using Facial Landmarks, Python, DLib, and OpenCV"**: This study addresses how to identify and categorize emotions from facial expressions using facial landmarks, Python, DLib, and OpenCV.
- **S. Saha and S. Ghosh's "Emotion Recognition Using Speech Signals"**: In this study, techniques for machine learning are used to identify emotions in speech data. To categorize emotions, the authors employ Mel Frequency Cepstral Coefficients (MFCCs) and Support Vector Machines (SVMs).
- **T. A. Alghamdi and N. Kumar's "Emotion Detection from Text Using Machine Learning Techniques"**: This study analyzes text data to identify emotions using machine learning techniques. To categorize emotions in text, the authors combine feature extraction methods and classification algorithms.
- **"Emotion Detection from EEG Signals Using Machine Learning Techniques" by V. S. Bhatt and S. S. Chaudhary**: This study examines how to extract emotions from EEG waves using machine learning algorithms. To categorize emotions using EEG data, the authors combine signal processing methods and machine learning algorithms.

## **8. Basic underlying concepts**

---

In this section, the concepts required for this project have been outlined.

### **8.1 Natural Language Processing**

The study of natural language processing (NLP) focuses on how people interact with computers through language. One of the most popular NLP uses is sentiment analysis, which determines if data is positive, negative, or neutral. It is frequently used to track consumer sentiment towards brands and products and to comprehend what the public wants.

Sentiment analysis uses a combination of natural language processing (NLP) and learning approaches to give sentiment scores to themes, categories, or entities inside a phrase. To accomplish this, divide the text into manageable chunks, then go over each chunk and analyze it to determine its sentiment by looking at the words used and the context.

Sentiment analysis can help organizations understand more about their target markets through the examination of consumer comments, social media monitoring, and market research.

Here are a few pre-steps performed in NLP:

1. Data Acquisition
2. Text Cleaning
3. Tokenization
4. Stop Word Removal
5. Stemming/Lemmatization
6. Part-of-Speech Tagging

### **8.2 Long-Short-Term Memory**

It is a form of recurrent neural network (RNN) architecture created to address the problem of vanishing gradients in conventional RNNs. Vanishing gradients arise when the gradients used for backpropagation during training are repeatedly multiplied by the weights in each layer of the network, resulting in very small gradients that make it challenging to update the weights in the network's earlier layers. In order to learn long-term dependencies, LSTM addresses this problem by introducing a gating mechanism that selectively controls the flow of information through the network. Three primary gates are used to do this:

- The input gate selects the data from the current input that should be saved in the cell state.
- The forget gate decides which information from the previous cell state should be forgotten.
- The output gate: The information from the current input and the prior cell state that should be sent to the output is decided by this.

A memory cell in the LSTM is also in charge of storing and transmitting information over time. The three gates described above control this memory cell by controlling which data is added, erased, or sent to the output.

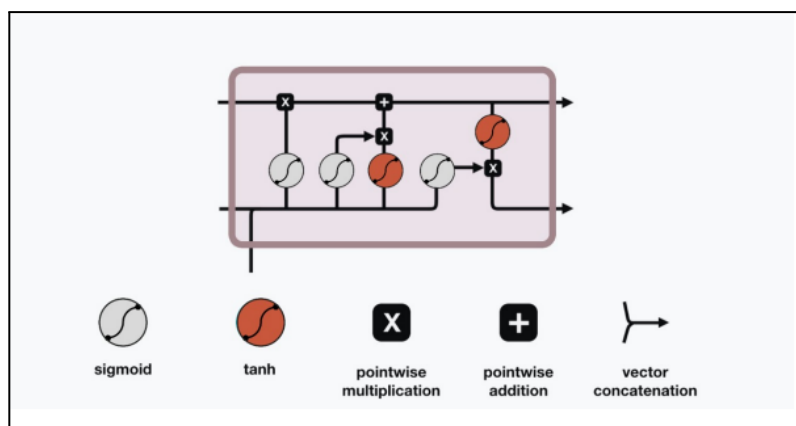


Fig. 1: Architecture of a Long-Short-Term Memory Network Model

### 8.3 Word2Vec (W2V)

Word2Vec (W2V) is a method of natural language processing that maps words to high-dimensional numerical value vectors. Based on patterns of word co-occurrence in a huge corpus of text, this particular sort of neural network learns to represent words as dense vectors of continuous numbers, encapsulating the semantic and syntactic links between words. Word2Vec's capacity to capture word similarity and relatedness, enabling a more accurate representation of language data, is one of its main advantages. In a Word2Vec model, for instance, words with similar meanings like "happy" and "joyful" are likely to have comparable vector representations.

When Word2Vec and LSTM are combined, a potent language model that can faithfully represent the emotional context of a text can be produced. The words in the incoming text are first converted into their matching Word2Vec vectors as part of the preprocessing phase. The LSTM network then receives these vectors and learns to categorize the input text according to its emotional content. The LSTM network modifies its weights throughout the training phase to reduce the discrepancy between predicted and real emotional labeling.

After being trained, the model may be used to categorize the emotional content of the fresh text by putting it through the network to get the expected emotional label.

## 8.4 F1 score

Evaluation metrics are used in machine learning and natural language processing to gauge a model's effectiveness on a particular job. One such metric that is frequently used to assess the effectiveness of a classification model is the F1 score.

The harmonic mean of recall and precision is the F1 score. Recall measures how many of the actual positive samples were predicted as positive by the model, whereas precision evaluates how many of the projected positive samples are actually positive. These two measurements are used to provide the F1 score, which offers a fair assessment of the model's performance.

- Defining an F1 score is as follows:

$$\text{F1 score} = 2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$$

where recall is the ratio of the number of true positives to the total number of true positives, and precision is the number of true positives divided by the total number of true positives and false positives.

The F1 score has a range of 0 to 1, with 0 representing subpar performance and 1 representing excellent precision and recall. A high F1 score means that the model is accurate in predicting both positive and negative samples, and it also means that the model has high precision and good recall.

## 8.5 GPT-3

The modern language model GPT-3 (Generative Pre-trained Transformer 3) created by OpenAI is based on the Transformer architecture. It is a deep learning model that has been pre-trained on a vast quantity of text data, enabling it to provide responses in natural language that are similar to those of humans to a variety of tasks. The fundamental idea behind GPT-3 is to pre-train the model on massive volumes of text data using unsupervised learning in order to enable it to recognize the underlying linguistic patterns and correlations. During this pre-training, the model is trained to predict the next word in a sequence using a self-supervised learning strategy. Language modeling is the term for this activity. The model can be fine-tuned for particular natural language processing tasks after pre-training, including text classification, question-answering, and language translation, among others.



## 9. Proposed Approach

Our suggested solution employs complex models to detect user emotions, allowing us to gather useful insights. To do so, we employ Natural Language Processing (NLP) to clean and preprocess the data, as well as datasets like FastText and RAVDESS to train and test the model on patient data.

We use speech-to-text conversion to improve user comfort by allowing users to communicate with the chatbot. Furthermore, our approach uses recent GPT-3 advances to identify emotions, making it more intelligent and capable of generating human-like reactions.

Our concept intends to help professionals by allowing them to track patient emotions on a regular basis. To accomplish this, we save emotion data in a database that includes text, timeframe, and count for each user, allowing for improved diagnosis. This information can be communicated with relevant parties to enable real-time monitoring of the patient's mental state and, if necessary, early action.

Overall, our suggested method employs advanced models and natural language processing (NLP) to detect emotions in users, providing useful insights to assist specialists in their diagnosis. We hope to construct a more sophisticated program that can increase emotion recognition accuracy by integrating speech-to-text and GPT-3.

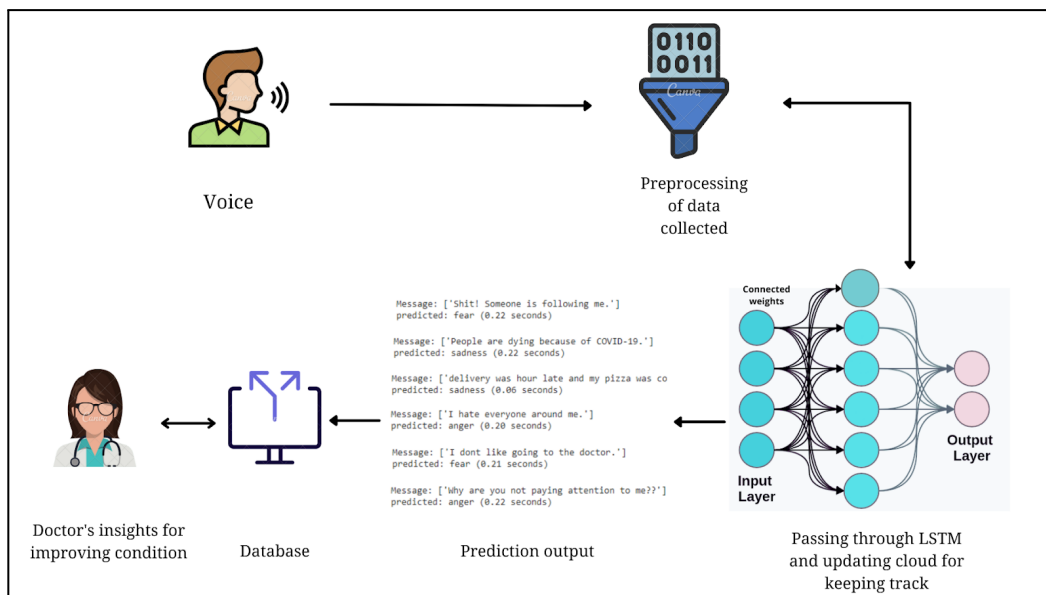


Fig 2: Schematic diagram of the multimodal emotion detection system.

## 10. Project Architecture

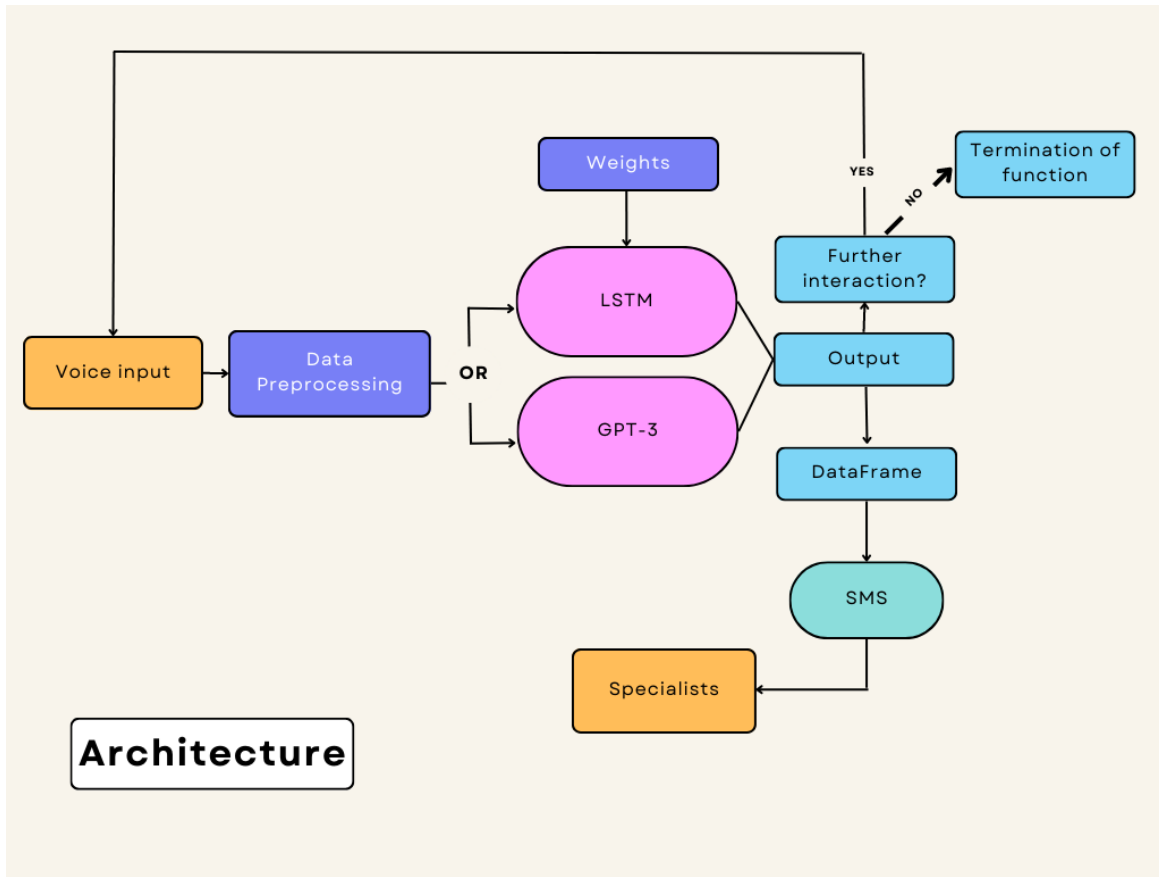


Fig 3: Project Architecture

### 10.1 Input and Preprocessing

In our project, voice input serves as the main input source. The first step entails textifying the voice input before performing sentiment analysis during the preprocessing stage. Word2Vec (W2V), a technique that assigns high-dimensional numerical value vectors to words, is used to do this. With the aid of this technique, we can see that words with related meanings, such as happy or joy, have related numerical value vectors. This enables us to correctly connect words that sound similar to emotions. The input and preprocessing phases of our project are when these tasks are completed.

## **10.2 Processing**

After the input and preprocessing stages, the data is susceptible to analysis using GPT-3, NLP, and long-short-term memory technologies. How we use these technologies to assess and analyze user emotions and sentiments is described below:

### **10.2.1 Using Natural Language Processing (NLP)**

The study of how language interacts with computer systems is known as natural language processing (NLP). Sentiment analysis is a well-known use of NLP that involves determining whether an input is good, negative, or neutral. This procedure includes a number of steps, including stop-word elimination, part-of-speech tagging, text cleaning, and data gathering.

To determine the underlying sentiment of a statement, NLP breaks it down into smaller parts. However, focusing entirely on NLP might not produce the best results for recognizing emotions. As a result, other complementary strategies must be used in addition to NLP.

### **10.2.2 Using Long-Short-Term Memory (LSTM)**

In our project, we combine Word2Vec (W2V) and Long-Short Term Memory (LSTM) to reliably identify emotions.

An artificial neural network with the ability to learn sequential patterns in data is called LSTM. We are able to identify the emotion that goes with a sentence by using LSTM to assess its emotional context. This is accomplished by teaching the LSTM on a big corpus of text data, enabling it to identify particular linguistic patterns that correspond to various emotions. In our implementation, the preprocessed text input is fed into the LSTM model, which outputs an emotion label.

The W2V method, on the other hand, converts words into multidimensional numerical vectors. In order to recognize words with comparable meanings and give them equivalent numerical values, our emotion detection system uses W2V. We can precisely map phrases that sound similar to the respective emotions using W2V. For instance, since the words "happy" and "joyful" evoke similar emotions, they might be given comparable numerical vectors.

### **10.2.3 Using GPT-3**

In our study, we also use GPT-3 to conduct emotion analysis in addition to LSTM and W2V. An advanced language model called GPT-3 (Generative Pre-trained Transformer 3) uses deep learning methods to produce text that resembles human speech.

In our project, we employ GPT-3 to provide answers that are in line with the user's emotional state. We can give GPT-3 a better knowledge of the user's emotional state by analyzing the input text and assigning a matching emotion using LSTM and W2V. The relevant replies that correspond with the user's emotional state are then generated by GPT-3 using this information. In contrast, if the user displays negative feelings, GPT-3 may produce a more sympathetic response. For example, if the user exhibits good emotion, GPT-3 may produce an uplifting and cheery response.

GPT-3 can also be used to improve our system for accurately detecting emotions. We may assess whether the assigned emotion appropriately reflects the text's context by using GPT-3 to analyze the resulting text output. This can assist us in enhancing the precision of our emotion recognition system as a whole.

## **10.3 Using Output**

In our research, we've created a user-interactive system that lets people submit data, which is then processed to gauge their emotional state. The system gives the user the choice to either interact with it again or quit the program after processing. If the user decides to proceed, they are brought back to the input stage where they can supply additional information to be processed. The processed output is saved in a dataframe after the user selects to quit the program.

Our technology is set up to store user data in a dataframe so that experts can further analyze it. This is especially helpful for mental health specialists who might need to access a patient's data to assist in the diagnosis and treatment of various mental health issues. To make this easier, we added a function that delivers the analyzed data to experts through SMS. The ability to track emotional patterns and trends over time is made possible by the storage of the user's data in a dataframe.


## 11. Algorithm

---

The main motive behind the Data Flow Methodology is the processing of the original data in such a way that it becomes easy for our proposed architecture. In deep learning, LSTM-RNNs (Long Short Term Memory Recurrent Neural Networks) require time-series data, continuous-time data, or in other words sequential data, data that depends on time [4,15,1]. LSTMs are quite advanced architecture that is capable of processing time-series data. We proposed this data flow modeling to convert our original data set into the time-series format. The proposed method for Emotion detection can be explained in the following steps:

**1. After getting the input, the words have been tokenized first. This segregates each word which would help us embed the sentence properly.**

Tokenizing the input words is the first stage. Tokenization, which in this context refers to individual words, is the process of dividing a string of text into smaller units. It is simpler to process and analyze the input text and get it ready for embedding, which is the next stage in the suggested procedure, by breaking it down into its individual words. Tokenization essentially entails the transformation of unstructured text input into structured data that can then be further processed and analyzed using deep learning methods like LSTM-RNNs.

```
 #To input the data to our NN Model we'll need some preprocessing:

#Tokenize our texts and count unique tokens
#Padding: each input (sentence or text) has to be of the same length
#Labels have to be converted to integers and categorized

def clean_text(data):

    # remove hashtags and @usernames
    data = re.sub(r"#[\d\w\.]+", '', data)
    data = re.sub(r"@[\d\w\.]+", '', data)

    # tokenization using nltk
    data = word_tokenize(data)

    return data
```

```
[ ] tokenizer = Tokenizer()
    tokenizer.fit_on_texts(texts)

    sequence_train = tokenizer.texts_to_sequences(texts_train)
    sequence_test = tokenizer.texts_to_sequences(texts_test)

    index_of_words = tokenizer.word_index

    # vocab size is number of unique words + reserved 0 index for padding
    vocab_size = len(index_of_words) + 1

    print('Number of unique words: {}'.format(len(index_of_words)))
```

Fig 4: Tokenizing

2. The length of each sentence being different, becomes difficult if we pass it through the model. The length of each sentence should be equal. So, it is mandatory to convert every sentence to an approximate number of words. For this, we have taken the help of padding.

The second stage entails making sure that each sentence in the incoming data is around the same length. This is significant since the input to LSTM-RNNs must have a set sequence length. Variations in the duration of the input sequence can make it difficult to train the model.

The proposed solution uses padding to add extra tokens (often zeros) to the ends of sentences that are shorter than the input sequence's maximum length in order to solve this problem. All phrases are made to have the same length by inserting padding tokens, guaranteeing that the LSTM-RNN model can process them correctly.

```
[ ] X_train_pad = pad_sequences(sequence_train, maxlen = max_seq_len )
    X_test_pad = pad_sequences(sequence_test, maxlen = max_seq_len )

    X_train_pad
```

```
array([[ 0,  0,  0, ..., 119,  51, 345],
       [ 0,  0,  0, ..., 37, 277, 154],
       [ 0,  0,  0, ..., 16,  2, 1210],
       ...,
       [ 0,  0,  0, ..., 876,  4, 909],
       [ 0,  0,  0, ...,  1,  6, 117],
       [ 0,  0,  0, ..., 10259, 173, 13]], dtype=int32)
```

Fig 5. Padding

3. Now each word needs to be embedded in some numeric representation, as the model understands only numeric digits. So for this task, we used Fasttext. It is easy to download and import any pre-trained word embedding as it is available. The 300-dimensional w2v pre-trained on Wikipedia articles has been used here.

The LSTM-RNN model can only process numerical data, thus the third step entails turning each word in the input phrases into a numerical representation. Pre-trained word embeddings, which are vector representations of words in a high-dimensional space, are used in the method to do this.

In this instance, the technique downloads and imports pre-trained word embeddings using FastText, a library for effective learning of word representations. The technique specifically makes use of a word2vec model with 300 dimensions that have already been pre-trained on Wikipedia articles.

The method eliminates the requirement to train its own word embeddings from scratch, which can be computationally expensive and time-consuming, by leveraging pre-trained word embeddings. Instead, it can use the pre-trained embeddings' prior knowledge to represent the input phrases in a meaningful fashion that the LSTM-RNN model can understand.

```
#We can download and import any pre-trained word embeddings.
#We will use 300 dimensional w2v pre-trained on wikipedia articles.
#Free Resource : https://fasttext.cc/docs/en/english-vectors.html

import urllib.request
import zipfile
import os

fname = 'embeddings/wiki-news-300d-1M.vec'

if not os.path.isfile(fname):
    print('Downloading word vectors...')
    urllib.request.urlretrieve('https://dl.fbaipublicfiles.com/fasttext/vectors-english/wiki-news-300d-1M.vec.zip',
                               'wiki-news-300d-1M.vec.zip')
    print('Unzipping...')
    with zipfile.ZipFile('wiki-news-300d-1M.vec.zip', 'r') as zip_ref:
        zip_ref.extractall('embeddings')
    print('done.')

    os.remove('wiki-news-300d-1M.vec.zip')
```

Fig 6. Embedding in numeric representation

4. The first word of each row is the character that is to be embedded. And from the column to the last column, there is the numeric representation of that character in a 300d vector form.

Based on the pre-trained word embeddings, step four entails building an embedding matrix that represents each character in the input sentences as a 300-dimensional vector.

Each row in the two-dimensional array that makes up the embedding matrix represents a different character from the input sentences. Each row's first column contains the character itself, while the following columns provide the character's 300-dimensional vector numeric representation based on the pre-trained word embeddings.

The LSTM-RNN model can process the input data and predict the emotion portrayed in the sentence by using this embedding matrix to look up the relevant vector representation for each character in the input sentence.

```
[ ] #Defining vector space dimension and fixed input size

# Number of labels: joy, anger, fear, sadness, neutral
num_classes = 5

# Number of dimensions for word embedding
embed_num_dims = 300

# Max input length (max number of words)
max_seq_len = 500

class_names = ['joy', 'fear', 'anger', 'sadness', 'neutral']
```

Fig 7. Character embedded in Table



5. Now, the preprocessing part is over and now we need to perform the following things:

- **Do one-hot encoding of each emotion:** This phase entails encoding each emotion label in the dataset as a binary vector with a length equal to the number of distinct emotions in the dataset. For instance, if the dataset contains four emotions, they might be encoded as "happy"=[1,0,0,0], "sad"=[0,1,0,0], "angry"=[0,0,1,0], and "neutral"=[0,0,0,1].
- **Split the dataset into train and test sets:** The dataset is divided into two different sets: a training set and a testing set—in order to train and evaluate the LSTM-RNN model. The model is trained on the training set, and its performance is assessed on the testing set.
- **Train the model on our dataset:** The LSTM-RNN model is trained on the training set using backpropagation and gradient descent after the data has been preprocessed and divided into training and testing sets. The model gains the ability to predict the appropriate emotion label for each input sentence during training.
- **Test the model on the test set:** The model's performance is assessed on the testing set after it has been trained on the training set. The model is asked to forecast the associated emotion label given input sentences from the testing set. The number of accurate predictions the model makes is then used to determine its accuracy.

```
[ ] #Categorizing the labels

encoding = {
    'joy': 0,
    'fear': 1,
    'anger': 2,
    'sadness': 3,
    'neutral': 4
}

# Integer labels
y_train = [encoding[x] for x in data_train.Emotion]
y_test = [encoding[x] for x in data_test.Emotion]
```

```
[ ] y_train = to_categorical(y_train)
    y_test = to_categorical(y_test)

    y_train
```

Fig 8. Performing OHE on the dataset.

```
#Importing the pretrained words

def create_embedding_matrix(filepath, word_index, embedding_dim):
    vocab_size = len(word_index) + 1 # Adding again 1 because of reserved 0 index
    embedding_matrix = np.zeros((vocab_size, embedding_dim))
    with open(filepath) as f:
        for line in f:
            word, *vector = line.split()
            if word in word_index:
                idx = word_index[word]
                embedding_matrix[idx] = np.array(
                    vector, dtype=np.float32)[:embedding_dim]
    return embedding_matrix
```

Fig 9. Importing the pre-trained words.

```
[ ] batch_size = 128
    epochs = 15

    hist = model.fit(X_train_pad, y_train,
                     batch_size=batch_size,
                     epochs=epochs,
                     validation_data=(X_test_pad,y_test))
```

Fig 10. Training the model

## ALGORITHM USING GPT3 PROGRAM

1. **Setting up resources:** Importing necessary libraries such as OpenAI, pandas, and speech\_recognition. Then, it sets up the API key for OpenAI by assigning the API key to the 'api\_key' attribute of the OpenAI module. This key will be used later to make requests to the OpenAI API. Next, it sets up an Excel file named 'user\_emotions.xlsx' to store the emotional data of users. The path of the file is specified as '/content/user\_emotions.xlsx'.

```
# Importing necessary libraries
import openai
import pandas as pd
import speech_recognition as sr

# Setting up OpenAI API key
openai.api_key = "sk-3jxkvldVPICoWEqu7I8mT3B1bkFJmZ6gQtq8ujNNHtjVcWQL"

# Setting up Excel file for saving emotions
excel_file = "/content/user_emotions.xlsx"
```

Fig 14. Setting up resources.

2. **Interacting with the user to take voice input:** The chatbot first prompts the user for their name and then prompts them to speak for a specified number of seconds, which is converted to a WAV audio file. The WAV file is then converted to the required format for OpenAI's language model. The chatbot then provides a response to the user based on the emotional content of their speech. If the user responds with 'y', they are prompted to speak again, and the process is repeated. If the user responds with 'n', the program terminates. If the user responds with an invalid input, they are prompted to select 'yes' or 'no' again.

```

# Defining main function
if __name__ == "__main__":

    user = input("May I know your name please?: ")

    # obtain audio from the microphone
    r = sr.Recognizer()
    n=int(input("Enter the number of seconds:"))
    print("Speak Anything :")
    print("Time Remaining:")
    wave=record(n)
    audio_file = convert_audio_for_model(wave)
    audio_file_path = '/content/converted_audio_file.wav'
    lang = "en"
    response = chatbot_response(audio_file_path, lang, user)
    while True:
        if response.lower() == "y":
            r = sr.Recognizer()
            n=int(input("Enter the number of seconds:"))
            print("Speak Anything :")
            print("Time Remaining:")
            wave=record(n)
            audio_file = convert_audio_for_model(wave)
            audio_file_path = '/content/converted_audio_file.wav'
            lang = "en"
            response = chatbot_response(audio_file_path, lang, user)

```

Fig 15. Interacting with the user to take voice input

3. **Recording Audio:** A function named 'record' is defined that records audio for a default duration of 5 seconds. It uses the Javascript function 'RECORD' to prompt the user to record their audio. The function takes an optional parameter 'sec' which is the duration for recording in seconds. The recorded audio is stored in a variable 's' after being encoded in base64 format. The base64-encoded audio data is then decoded using the 'b64decode' function from the 'base64' module. The decoded audio data is then written to a file named 'audio.wav'. The function returns the path to the recorded audio file ('audio.wav').

```

#record function that will record the audio for a default of 5 secs
def record(sec=5):
    display(Javascript(RECORD))
    s = output.eval_js('record(%d)' % (sec*1000))
    b = b64decode(s.split(',')[1])
    with open('audio.wav','wb') as f:
        f.write(b)
    return 'audio.wav'

```

Fig 16. Recording Audio

4. **Using GPT3 to analyze the audio:** The 'chatbot\_response' function is defined, which takes in three parameters: 'audio\_file\_path' (the file path of the audio file to be transcribed), 'lang' (the language of the audio file), and 'user' (the name of the user). Within the function, the speech is converted to text using Google's speech recognition, and the resulting text is cleaned using regular expressions to remove any non-alphanumeric characters. Then, OpenAI's language model (Davinci) is used to analyze the emotional content of the text, by prompting the model with the text and asking it to generate the emotion behind the statement.

```

# Load the audio file
with sr.AudioFile(audio_file_path) as source:
    # Read the audio
    audio_text = r.record(source)

try:
    # Convert speech to text
    text = r.recognize_google(audio_text, language=lang)

    # Clean the text
    text = re.sub('[^A-Za-z0-9]+', ' ', text)

```

```

emotion = openai.Completion.create(
    model="text-davinci-003",
    prompt=f'''give me the emotion behind the statement
    "{text}"
    ''',
    temperature=0.7,
    max_tokens=256,
    top_p=1,
    frequency_penalty=0,
    presence_penalty=0
)
response = input("Do you want to talk further? (y/n): ")
emotion = emotion['choices'][0]['text'].strip() #put in excel sheet
user_emotions = {'name': user, 'text' : text, 'emotion': emotion} ##
save_to_excel(user_emotions)

```

Fig 17. Using GPT3 to analyze the audio.

- 5. Sending the message using Twilio API:** The Twilio API is used to send a text message to a phone number. The necessary libraries are imported, including the Twilio client. The Twilio client is set up using the account SID and authentication token. Next, an Excel file is read using Pandas. The data in the file are grouped by "Name" and the count of the "Emotion" column for each user is calculated. The resulting dataframe is then converted to a string. The user is prompted to enter a phone number in the format "+1234567890". The Twilio API is then used to send a message to that phone number with the counts string as the body of the message. Finally, the message SID is printed as confirmation that the message was sent.

```

!pip install twilio

# Import the required libraries
from twilio.rest import Client

# Set up the Twilio client
account_sid = 'ACe9a184c899c1a4d0a5ab37c8cdcb8ccf'
auth_token = '453095700da050ecc5f617347dee2fa9'
client = Client(account_sid, auth_token)

# Read the Excel file
df = pd.read_excel('user_emotions.xlsx')

# Group the data by "Name" and count the number of the "Emotion" column for each user
counts = df.groupby(['Timestamp', 'Name', 'Text', 'Emotion']).size().reset_index(name='Count')

# Format the counts into a string
counts_str = counts.to_string(index=False)

# Get the user's phone number as input
phone_number = input("Enter the phone number in the format +1234567890: ")

```

Fig 18. Sending the message using Twilio API

## 12. Outcomes

---

The goal of the work is to develop a system for emotion analysis using human interaction. To achieve this, the samples have been collected from available datasets. The architecture works initially by tokenizing every word after getting input. With padding, every sentence has been converted into several approximate words. Then each word has been embedded into a numeric representation with the Fasttext dataset. The performance of the model is around 86%, which is reported in Table 2. The confusion matrix of the model is shown in Figure 11.

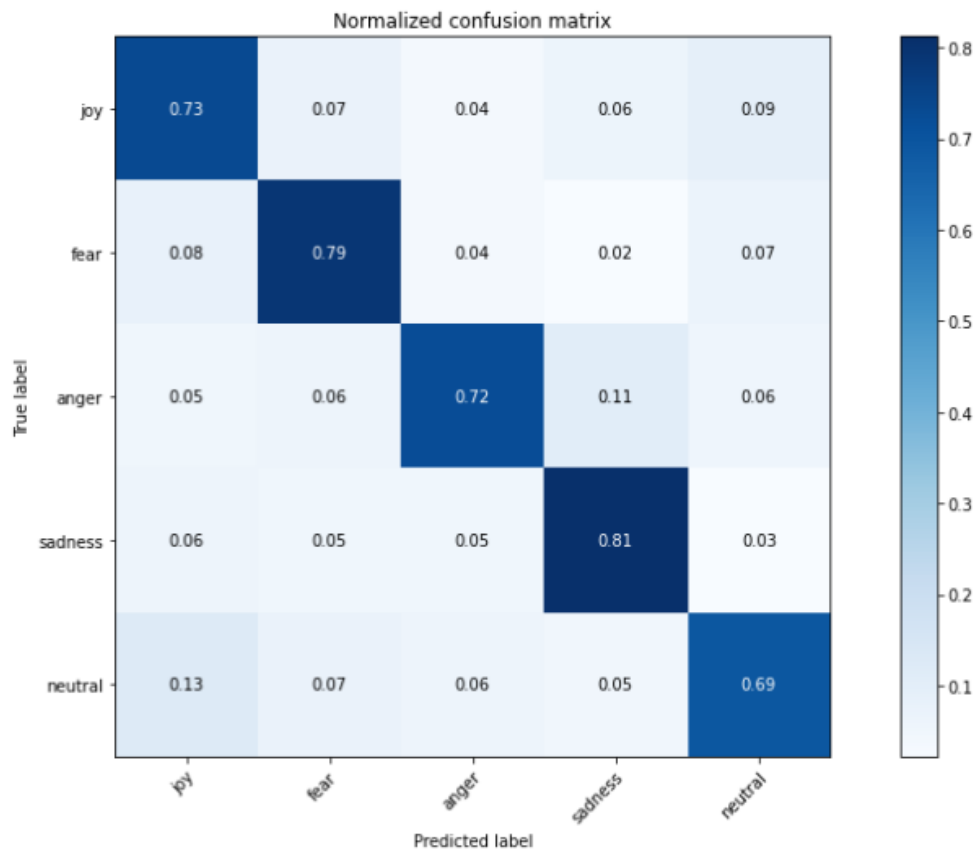


Fig 11. Confusion matrix with 30 epochs.

Table 2.Experimental results based on some random messages.

<b>Message</b>	<b>Prediction</b>	<b>Time (in a sec)</b>
Oh no, someone is following me.	Fear	0.22
People are dying because of Covid-19.	Sadness	0.22
Delivery was an hour late and my pizza was cold.	Sadness	0.06
I hate everyone around me.	Anger	0.20
I don't like going to the doctor.	Fear	0.21
Why are you not paying attention to me?	Anger	0.22
Bro! I got really good marks.	Joy	0.20
I am confused about my life.	Sadness	0.21
I am confused about my life.	Neutral	0.23
Why do we give exams?	Anger	0.20
This is a fine day.	Neutral	0.00
Congratulations, that's big news.	Joy	0.00
that room is so creepy	Fear	0.00
that man over there is running fast	Neutral	0.00



Advancement: This tool involves a chatbot that records the user's speech, translates it to text using Google's speech recognition, and then analyzes the text for emotional content using OpenAI's language model. The data, including the user's name and the analyzed emotion, is then stored in an Excel file and can be sent via SMS to the concerned individual. The user is asked to speak about a personal experience that evoked strong emotions, and the chatbot records their speech. The chatbot then transcribes the speech into text and uses OpenAI's language model to analyze the emotional content of the text. Further, Twilio is implemented to share the data via SMS to the input phone number.

### Taking User Input:

```
Enter the number of seconds:5
Speak Anything :
Time Remaining:
Do you want to talk further? (y/n): n
n
Thank you, hope you are feeling good now.
```

```
May I know your name please?: Adarsh
Enter the number of seconds:5
Speak Anything :
Time Remaining:
```

```
May I know your name please?: Ada
Enter the number of seconds:5
Speak Anything :
Time Remaining:
Do you want to talk further? (y/n): y
```

```

Enter the number of seconds:5
Speak Anything :
Time Remaining:
Do you want to talk further? (y/n): n
n
Thank you, hope you are feeling good now.

```

Fig 12. Taking user voice input along with the name

The chatbot was able to accurately analyze the emotional content of the participant's speech. The data collected in the Excel file showed the participants' names, the time and date of the recording, and the emotions detected by the language model.

The information may be used by mental health experts to develop a deeper knowledge of their patient's emotional health and enable more individualized treatment. In order to provide real-time monitoring of the patient's mental condition and early assistance when necessary, the chatbot also gives the option of messaging the data to a worried doctor or human.

Timestamp	Name	Text	Emotion	Count
2023-05-07 09:28:47	Ada	I m feeling sad today	Sadness.	1
2023-05-07 09:29:05	Ada	I want to eat an ice cream	Excitement	1
2023-05-07 09:34:26	Adarsh	I want to go	Longing	1
2023-05-07 09:34:39	Adarsh	I don t want to go to the hospital	Anxiety.	1

Fig 13. Output Dataframe.

Timestamp	Name	Emotion	Text
2023-05-07 09:28:47	Ada	Sadness.	I m feeling sad today
2023-05-07 09:29:05	Ada	Excitement	I want to eat an ice cream
2023-05-07 09:34:26	Adarsh	Longing	I want to go
2023-05-07 09:34:39	Adarsh	Anxiety.	I don t want to go to the hospital

Table 3. Data stored in Excel

### Message generated:

Timestamp	Name	Text	Emotion	Count
2023-05-07 09:28:47	Ada	I am feeling sad today.	Sadness	1
2023-05-07 09:29:05	Ada	I want to eat ice cream.	Excitement	1
2023-05-07 09:34:26	Adarsh	I want to go.	Longing	1
2023-05-07 09:34:39	Adarsh	I don't want to go to the hospital.	Anxiety	1

Table 4: Message sent to specialist

## 13. Conclusion

---

Our research and implementation aimed to provide solutions for emotional analysis using various methods, including AI bots. The system can determine the level of emotion and provide remedial solutions without the need for human intervention, which is often beneficial. Our platform offers easy accessibility and aims to improve emotional well-being. We have primarily implemented RAVDESS and FastText datasets for the prediction and collection of data. FastText illustrates and classifies texts, by comparison with larger datasets and provides precise outcomes, adaptable with the time required. RAVDESS, analyzes the individual's psychological state and severity of an expression, expressed through a statement. In the LSTM, Long short-term memory we applied activation functions namely, Sigmoid Activation Function and Hyperbolic Tangent Activation Function, for determining whether the input sequences should be stored in long-term or short-term memory. Lastly, we provided the end result of the formative development, to provide an insight into our ultimate objective by using GPT-3.

We have made significant progress on the project thanks to the addition of GPT-3. A huge language model called GPT-3 was created by OpenAI and is capable of producing text that is both coherent and contextually relevant. We were able to **increase the number of emotions the model can recognize beyond the five emotions** by introducing GPT-3 into the study. This made it possible to comprehend human emotions more thoroughly, which is crucial in practical applications like sentiment analysis and customer service.

The project has incorporated the Twilio API to send information about the user's emotions via message to the relevant person. Twilio is a cloud communications platform that offers a range of APIs for creating messaging and phone applications. By integrating the Twilio API into the project, we can easily transmit the user's data to the appropriate parties, which can help raise awareness of the user's health or other relevant concerns. In the future, the system will be upgraded with the model for depression detection. If the person is found to have depression, our AI-bot will provide necessary remedies, based on the severity of depression.

## 14. References

---

1. Al Banna, M.H., et al.: Attention-based bi-directional long-short term memory network for earthquake prediction. *IEEE Access* 9, 56589–56603 (2021)
2. Anagnostopoulos, C.N., Iliou, T., Giannoukos, I.: Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011. *Artificial Intelligence Review* 43(2), 155–177 (2015)
3. Dini, L., Bittar, A.: Emotion analysis on Twitter: the hidden challenge. In: *Proc. LREC'16*. pp. 3953–3958 (2016)
4. Fabietti, M., et al.: Artifact detection in chronically recorded local field potentials using long-short term memory neural network. In: *Proc. AICT 2020*. pp. 1–6 (2020)
5. Ghosh, T., et al.: An attention-based mood controlling framework for social media users. In: *Proc. Brain Informatics*. pp. 245–256 (2021)
6. Ghosh, T., et al.: A hybrid deep learning model to predict the impact of covid-19 on mental health from social media big data. *Preprints* 2021(2021060654) (2021)
7. Humphrey, E.J., Bello, J.P., LeCun, Y.: Moving beyond feature design: Deep architectures and automatic feature learning in music informatics. In: *ISMIR*. pp. 403–408 (2012)
8. Kahou, S.E., et al.: Emonets: Multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces* 10(2), 99–111 (2016)
9. Livingstone, S.R., Russo, F.A.: The Ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in North American English. *PloS one* 13(5), e0196391 (2018)
10. Mikolov, T., Grave, E., Puhersch, C., Joulin, A.: Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*, 1–4 (2017)
11. Mohammad, S.M., Bravo-Marquez, F.: Emotion intensities in tweets. *arXiv preprint arXiv:1708.03696*, 1–13 (2017)
12. Poria, S., Cambria, E., Howard, N., Huang, G.B., Hussain, A.: Fusing audio, visual and textual clues for sentiment analysis from multimodal content. *Neurocomputing* 174, 50–59 (2016)
13. PS, S., Mahalakshmi, G.: Emotion models: a review. *International Journal of Control Theory and Applications* 10, 651–657 (2017)

14. Sailunaz, K., Dhaliwal, M., Rokne, J., Alhadj, R.: Emotion detection from text and speech: a survey. *Social Network Analysis and Mining* 8(1), 1–26 (2018)
15. Satu, M., et al.: Towards improved detection of cognitive performance using bidirectional multilayer long-short term memory neural network. In: *Proc. Brain Informatics*. pp. 297–306 (2020)
16. Satu, M.S., et al.: Tclustvid: A novel machine learning classification model to investigate topics and sentiment in covid-19 tweets. *Knowledge-Based Systems* 226, 107126 (2021)
17. Semwal, N., Kumar, A., Narayanan, S.: Automatic speech emotion detection system using multi-domain acoustic feature selection and classification models. In: *Proc. ISBA*. pp. 1–6 (2017)
- 18. Bhagat, D., Ray, A., Sarda, A., Dutta Roy, N., Mahmud, M., De, D. (2023). Improving Mental Health Through Multimodal Emotion Detection from Speech and Text Data Using Long-Short Term Memory. In: Mandal, J.K., De, D. (eds) Frontiers of ICT in Healthcare. Lecture Notes in Networks and Systems, vol 519. Springer, Singapore. [https://doi.org/10.1007/978-981-19-5191-6\\_2](https://doi.org/10.1007/978-981-19-5191-6_2)**