*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 9*
*DOI: https://doi.org/10.17762/ijritcc.v11i9.8325*
***Article Received:** 09 July 2023 **Revised:** 30 August 2023 **Accepted:** 15 September 2023*

# Enhancing Mental Health Awareness through Twitter Analysis: A Comparative Study of Machine Learning and Hybrid Deep Learning Techniques

**Rohini Kancharapu[1], Sri Nagesh Ayyagari[2]**
[1]Gayatri Vidya Parishad College of Engineering for Women
CSE Department, Kommadhi
Visakhapatnam, India
rohinik3108@gmail.com
[2]Rayapati Venkata Rangarao & Jagarlamudi Chandramouli College of Engineering,
CSE Department, Chowdavaram
Guntur, India.
asrinagesh11@gmail.com

**Abstract**—This study explores the utilization of social media data, specifically tweets and comments, for gaining insights into individuals' mental health conditions. The objective is to enhance mental health awareness and enable early detection and intervention. Twitter data is collected using depression-related keywords, and two models are employed: a Random Forest model with TF-IDF and a hybrid CNN-LSTM model incorporating word2vec. The performance of the CNN-LSTM model surpasses that of the Random Forest model, achieving an accuracy rate of 89.4%. Furthermore, a user interface is developed to analyze users' Twitter profiles based on their tweets, allowing for potential intervention through automated reply messages. By harnessing social media data and advanced machine learning techniques, this research contributes to improving mental health awareness and timely addressing of mental health concerns.

**Keywords**-Convolution Neural Network(CNN); Depressed Keywords; Long Short Term Memory (LSTM); Random Forest (RF); Twitter; VADER.

## I. INTRODUCTION

Depression is a pervasive mental health disorder that affects a significant portion of the global population, leading to substantial personal, social, and economic burdens. According to the World Health Organization (WHO), depression is ranked as the leading cause of disability worldwide [21,24]. Timely detection and intervention are crucial in addressing this public health issue effectively [26]. With advancements in technology, mental health professionals and researchers now have a powerful tool at their disposal – social media data. By monitoring and analyzing negative emotions and attitudes expressed on these platforms, we can identify individuals at risk and provide them with the necessary support and resources.

In recent years, the advent of social media platforms has revolutionized communication and provided individuals with a digital space to express their thoughts, emotions, and experiences [5-7]. Social media platforms, such as Twitter, have become a vast repository of user-generated content, offering valuable insights into people's lives, including their mental well-being [1]. Recognizing the potential of social media data for mental health research, scholars have explored novel approaches to analyze and leverage this data effectively. One promising avenue is sentiment analysis [25], a computational technique that focuses on understanding and extracting emotional signals from text data.

This study aims to harness the power of sentiment analysis and social media data to enhance mental health awareness and early detection of depression. Specifically, we focus on Twitter as a source of data, given its popularity and extensive user base. By extracting data from Twitter using specific keywords associated with depression, we delve into the vast pool of information available on the platform. By analyzing tweets and comments, we can uncover valuable information regarding individuals' mental states and emotional well-being.

To achieve this, we employ advanced machine learning techniques to process and analyze the vast amount of textual data available on Twitter. In our analysis of Twitter data, we utilize the VADER algorithm to determine the polarity score of tweets and categorize users' mental health status based on their content [21]. Additionally, we employ text-to-vector conversion techniques such as Word2vec and TF-IDF, ), enable us to transform text into numerical

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 9*
*DOI: https://doi.org/10.17762/ijritcc.v11i9.8325*
***Article Received:*** *09 July 2023* ***Revised:*** *30 August 2023* ***Accepted:*** *15 September 2023*

representations that can be effectively combined with Deep learning models like CNN-LSTM and Machine learning models like Random Forest, to analyze tweets and classify users as depressed or not. By leveraging the strengths of both models, we aim to achieve accurate classification of Twitter users' mental health status, specifically identifying whether they exhibit signs of depression or not.

Furthermore, we develop a user interface that integrates with the CNN-LSTM model, allowing for real-time analysis of users' Twitter profiles based on their posted tweets. When a person is flagged as potentially depressed, an automated reply message is sent to provide support and relevant resources. Through this research, we strive to contribute to the growing field of mental health analysis using social media data. By effectively leveraging sentiment analysis techniques and advanced machine learning models, we can enhance mental health awareness, facilitate early detection, and ultimately improve the well-being of individuals affected by depression.

To provide a practical application of our research, we developed a user interface that integrates with the CNN-LSTM model. This interface allows us to analyze users' Twitter profiles based on their posted tweets. When a person's tweets suggest signs of potential depression, an automated reply message is sent to them, offering support and resources.

By effectively utilizing social media data and implementing advanced machine learning techniques, our research contributes to enhancing mental health awareness. The analysis of tweets provides valuable insights into individuals' mental states, enabling early detection and intervention. This approach has the potential to make a significant impact on mental health, as it presents an opportunity for timely support and intervention for those who may be struggling.

Taking it a step further, our research emphasizes proactive measures by sending response messages to users who are flagged as potentially depressed. By reaching out and alerting their followers, we aim to create a supportive network and foster a sense of community.

In conclusion, this research highlights the importance of utilizing social media data and advanced machine learning techniques to enhance mental health awareness. By analyzing tweets, we gain valuable insights into individuals' mental states, enabling early intervention and support. This approach has the potential to positively impact the lives of countless individuals, contributing to the overall well-being of society.

## II. LITERATURE REVIEW

Several studies have been conducted in the field of sentiment analysis [25] and mental health detection using social media data [6-16]. In a study by Rinki Chatterjee, Rajeev Kumar Gupta, and Bhavana Gupta [1], a depression detection model was created using social media data, specifically Facebook comments and Twitter tweets. The model utilized a RapidMiner-created workflow and employed a naive Bayes classifier for classification. The researchers identified factors indicating depression in social media users and described the overall workflow of their proposed system.

Ghaida and Danda B. Rawat focused on tracking COVID-19 opinions in Twitter data through a sentiment analysis system [2]. They utilized Natural Language Processing (NLP) techniques and implemented a Recurrent Neural Network (RNN) model with Long-Short Term Memory Networks (LSTMs). Their workflow involved data collection, preprocessing, exploratory data analysis, and classification using the LSTM model. The study aimed to determine the sentiment (positive or negative) of COVID-19-related tweets.

Mohammad Ehsan Basiri, Shahla Nemati, Moloud Abdar, Erik Cambria, and U. Rajendra Acharya proposed an attention-based bidirectional CNN-RNN deep model (ABCDM) for sentiment analysis [3]. The model incorporated two bidirectional LSTM layers and a GRU layer to extract contextual information. Through an experiment with reviews and Twitter datasets, they demonstrated the superior performance of ABCDM compared to other sentiment analysis models. The proposed model also employed the global vector of word representations (GloVe) as embedding layer weights.

Shivam Behl, Aman Rao, Sahil Aggarwal, Sakshi Chadha, and H.S. Pannu developed a sentiment analysis model for disaster relief using Twitter data [4]. They compared supervised learning approaches and utilized a Multi-Layer Perceptron (MLP) network for multi-class classification of tweets related to resource requirements, availability, and neutral categories. The model achieved high classification accuracy on their original COVID-19 dataset, demonstrating its effectiveness in analyzing tweets for disaster relief purposes.

In another study by Kour1 and Manoj K. Gupta [5], a hybrid approach combining convolutional neural networks (CNN) and bidirectional LSTM was used for sentiment analysis of static Twitter data. The proposed system consisted of modules for data extraction, raw data analysis, preprocessing, feature extraction, depression classification, and parameter evaluation. The hybrid model outperformed other models, achieving high accuracy after optimization.

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 9*
*DOI: https://doi.org/10.17762/ijritcc.v11i9.8325*
***Article Received:** 09 July 2023 **Revised:** 30 August 2023 **Accepted:** 15 September 2023*

Hameed and Garcia-Zapirain [6] proposed a sentiment classification model using a single-layered BiLSTM (Bidirectional Long Short-Term Memory) for analyzing text emotions. They focused on developing a model that can effectively classify sentiments in text data. Alsagri and Ykhlef [7] introduced a machine learning-based approach for depression detection on Twitter using content and activity features. They aimed to identify depressive tendencies in social media posts by analyzing various factors such as the content of the posts and the user's activity patterns.

Shetty et al. [8] utilized deep learning and ensemble algorithms to predict depression using raw Twitter data. They employed a combination of deep learning models and ensemble techniques to accurately classify tweets into depression and non-depression categories. Zheng et al. [9] presented a graph attention model embedded with multi-modal knowledge for depression detection. Their approach incorporated both textual and visual information from social media posts to improve the accuracy of depression detection.

Trotzek et al. [10] focused on utilizing neural networks and linguistic metadata for early detection of depression indications in text sequences. They developed a model that combined the power of neural networks with linguistic metadata to identify early signs of depression in text data. Bhargava et al. [11] conducted research on depression detection using sentiment analysis of tweets. They leveraged sentiment analysis techniques to analyze the emotional content of tweets and identify patterns associated with depression.

Kaur et al. [12] proposed a sentiment analysis deep learning algorithm for analyzing COVID-19 tweets. They developed an algorithm that could effectively analyze tweets related to the COVID-19 pandemic and extract sentiment information to gain insights into people's emotions during the crisis. P. P. A. [13] performed a performance evaluation and comparison of deep learning techniques in sentiment analysis. They assessed the effectiveness of different deep learning models in sentiment analysis tasks and compared their performance to identify the most suitable approaches. Divyapushpalakshmi and Ramalakshmi [14] developed an efficient sentimental analysis approach using hybrid deep learning and optimization techniques for Twitter. They combined deep learning models with optimization techniques to improve the accuracy and efficiency of sentiment analysis on Twitter data.

Joshi and Kanoongo [15] reviewed the use of emotional artificial intelligence and machine learning for depression detection. They provided a comprehensive overview of different approaches and techniques used in the field of depression detection and highlighted the potential of emotional AI and machine learning in this context. Babu and Kanaga [16] conducted a review of sentiment analysis in social media data for depression detection using artificial intelligence. They explored various sentiment analysis methods applied to social media data and discussed their effectiveness in detecting signs of depression.

Guntuku et al. [17] examined the language of ADHD in adults on social media. They investigated the linguistic patterns and characteristics displayed by adults with ADHD on social media platforms to gain insights into their communication behaviors. Almouzini and Alageel [18] focused on detecting depressed Arabic users from Twitter data. They developed a method specifically tailored for the Arabic language to identify users who exhibit signs of depression based on their tweets and online activities.

Yang et al. [19] proposed a mental state knowledge-aware and contrastive network for early stress and depression detection on social media. Their model incorporated knowledge from external sources and utilized a contrastive learning approach to improve the accuracy of stress and depression detection. Kang et al. [20] introduced active learning with complementary sampling for instructing class-biased multi-label text emotion classification. They developed an active learning method that effectively selects informative samples for emotion classification tasks

These studies highlight the growing interest in sentiment analysis and mental health detection using social media data. They demonstrate the effectiveness of various techniques and models in analyzing and classifying users' mental health status based on their social media content. By leveraging advanced machine learning and deep learning techniques, researchers are able to contribute to the development of innovative approaches for early detection and intervention in mental health issues.

### *A. Novelty of our work*

The novelty of this research is that we have used dataset by collecting the tweets from Twitter API with the help of the keys generated from the developer account. We built the website with deep learning model of CNN-LSTM to classify the tweets. The website acts like a depression watch, where by submitting the username as input in the webpage, the model will analyze the tweets, that the user have posted on the present day and detects whether the user has depressed or not, if the person is identified as depressed based on the tweets posted in his/her account, a support message will be sent to the user’s tweet as a reply. We also created a webpage where we can identify user’s thoughts when submitted on our webpage, if person is detected as depressed then personal message will be sent to their account.

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 9*
*DOI: https://doi.org/10.17762/ijritcc.v11i9.8325*
***Article Received:** 09 July 2023 **Revised:** 30 August 2023 **Accepted:** 15 September 2023*

*B. Proposed system*

The system we have developed aims to analyze real-time streams of tweets obtained from the Twitter API using the tweepy library. To access the Twitter API, developers must create an account and obtain access keys. The live stream of tweets is then subjected to the VADER algorithm, which categorizes and assigns labels to tweets as positive, negative, or neutral based on their polarity scores.

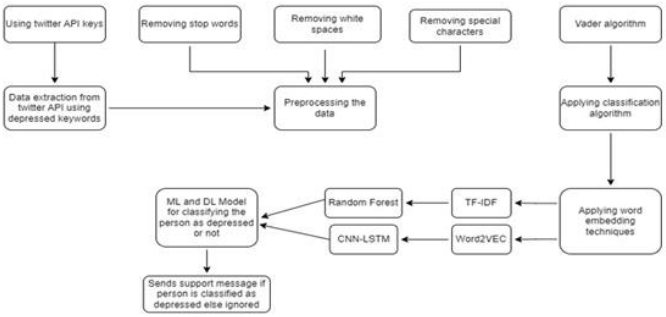


Figure 1. Proposed framework

To delve deeper into the tweet content, we employed embedding techniques such as word2vec and TF-IDF. These techniques convert the text data into a vector representation, enabling analysis and processing by machine learning models. Our model building process involves a hybrid approach that combines the Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models. The CNN-LSTM model is trained to extract relevant features from the text data, while the random forest algorithm is employed to classify tweets as either depressed or non-depressed based on these extracted features shown in Fig.1.

This hybrid approach enhances the accuracy of our model and enables effective classification of Twitter users as depressed or non-depressed based on their tweets. Furthermore, this approach allows us to handle a large volume of tweets in real-time, providing valuable insights into the mental health status of Twitter users. Overall, our proposed system offers a robust tool for mental health professionals and researchers to gain a better understanding of and monitor changes in mental health status through social media.

## III. METHODOLOGY

Within this section, we will outline the different modules employed in our work. Our decision was to utilize the Twitter dataset as our primary source of data. In order to access live tweets from the Twitter API, we proceeded by first setting up a regular Twitter account and subsequently creating a developer account. The access keys generated from the developer account were used to extract live tweets and their associated metadata, such as user names and descriptions. Once the data was collected, we preprocessed it before using it in subsequent steps. We used the VADER algorithm to label the dataset, which assigned a positive, negative, or neutral label to each tweet based on its polarity score. After completing the previous steps, we proceeded to train a model using the dataset that had been cleaned and labeled. We used a hybrid model of CNN-LSTM and Random Forest algorithm for this purpose.

Finally, we used the trained model to categorize a user's tweets as either depressed or non-depressed. Detailed explanations of the operations performed by each module can be found in the subsequent sections. Overall, our work involved collecting live tweets, preprocessing the data, labeling the dataset using the VADER algorithm, training a hybrid model, and categorizing a user's tweets as either depressed or non-depressed based on the trained model.

*A. Algorithms*

This section provides detailed explanation of our algorithms.

*Hybrid CNN_LSTM algorithm*

Input: preprocessed data labeled using VADER

Output: Classify the tweet of the user as positive negative and neutral

1. The data was divided into separate sets for training and testing purposes.
2. We began by initializing a Word2Vec model with CBOW architecture, specifying the desired vector size.
3. The Word2Vec model was trained on the training data for the specified number of epochs.
4. Convert each text document in the corpus into a fixed length vector using the trained Word2Vec model.
5. Train the CNN-LSTM hybrid model on the vectors and labels.
6. Evaluate the trained classifier on the testing set and calculate the classification metrics.
7. Return the trained Word2Vec model and CNN-LSTM hybrid model.
8. Categorize user's tweets as depressed or non-depressed using the trained model

*Random Forest algorithm*

Input: preprocessed data labeled using VADER

Output: Classify the tweet of the user as positive negative and neutral

1. The data was divided into separate sets for training and testing purposes.
2. Initialize a TF-IDF model with the given vector size.
3. Train the TF-IDF model on the training data.

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 9*
*DOI: https://doi.org/10.17762/ijritcc.v11i9.8325*
***Article Received:** 09 July 2023 **Revised:** 30 August 2023 **Accepted:** 15 September 2023*

4. Using the trained TF-IDF model, we converted each text document in the corpus into a fixed-length vector.
5. Train the Random Forest model on the vectors and labels.
6. Evaluate the trained classifier on the testing set and calculate the classification metrics.
7. Return the trained model.
8. Categorize user's tweets as depressed or non-depressed using the trained model

We divided the whole process into several modules and explained how each module works in this section illustrated in Fig.2.

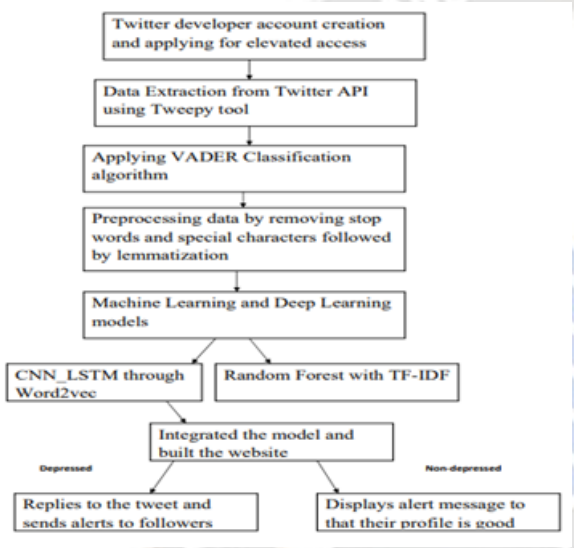


Figure 2. Content diagram

Firstly, to work with we need a dataset for that we choose to use twitter dataset. In order to collect live tweets from the twitter API we need to have access for a developer account. We need to create a normal twitter account first and using that we can create a developer account. After creating a developers account, we request for the access keys and generate the keys. These keys are used to extract live tweets along with other metadata like user name, description etc. this dataset is then preprocessed and used for next steps. The dataset is labeled using the VADER algorithm. The model is trained using the cleaned and labeled dataset. By utilizing the trained model, we are able to classify tweets from users into either depressed or non-depressed categories. The functionality of each module is thoroughly explained in subsequent sections for a clear understanding.

*Step 1: Generating access keys for Twitter*

Before applying for Twitter API access, we must first create a Twitter API developer account; this developer account contains a profile that includes information about the reason for accessing Twitter API as well as our research details. We need API access to extract tweets from Twitter. The Twitter developer account provides two types of access: essential access and elevated access. Only some rights are included in essential access, whereas elevated access has more privileges than essential access. Essential access is granted simply by registering for a developer account, but elevated access is granted only after we have completed the questionnaire. We can request access keys after creating a developer account on Twitter with a Twitter account. To do so, we must complete the questionnaire, which includes questions such as what the data will be used for, among other things. We start with essential access, but once Twitter accepts our request, we can upgrade to elevated access, which gives us more privileges. We can generate and use the keys once the request acceptance email is sent.

The following generated keys are used to extract tweet data.

- API_KEY
- API_SECRET
- ACCESS_TOKEN
- ACCESS_TOKEN_SECRET

*Step 2: Data Extraction*

Using the keys generated in the previous step we can extract data from Twitter API. The tweepy tool was used for extraction. We extract nearly 40,000 tweets as well as other tweet-related information. The tweets are extracted using some search terms related to depression, so only data containing these search terms is extracted. The depression search terms come from a document uploaded by Jonathan Scour [22] field in research gate. The extracted data includes tweets as well as several other attributes such as username, user description, TweetID, favourite count, retweet count, and so on.

*Step 3: Preprocessing the Tweets*

Data preprocessing is an essential step when working with datasets as it aids in reducing unnecessary data and extracting vital features. The initial stage of this process involves data cleaning, where special characters and stop words are removed from the tweets. This ensures that the dataset is sanitized and prepared for labeling. To remove stop words and special characters from the tweets, we utilized the Natural Language Tool Kit (NLTK), a popular Python toolkit for working with human language data. NLTK offers a range of text processing libraries and wrappers for tasks like categorization, tokenization, stemming, tagging, parsing, and

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 9*
*DOI: https://doi.org/10.17762/ijritcc.v11i9.8325*
***Article Received:** 09 July 2023 **Revised:** 30 August 2023 **Accepted:** 15 September 2023*

semantic reasoning. Additionally, it provides access to a discussion forum and interfaces with over 50 corpora and lexical resources, including WordNet.

NLTK includes a predefined list of stop words in 16 different languages, which we employed to eliminate stop words from the generated tweets. Furthermore, regular expressions were used to strip away special characters from the text. These measures ensured that the dataset was thoroughly cleansed and ready for further processing. In summary, preprocessing plays a vital role in improving the model's accuracy. By eliminating unnecessary data and extracting essential features, the model can be trained on relevant data, resulting in more precise predictions.

*Step 4: Labeling Data for Classification of Tweets Using VADER algorithm*

In the proposed system, tweets are retrieved through the Twitter API using TweepyV2. The tweets are then labeled using the Valence Aware Dictionary and sEntiment Reasoner(VADER) algorithm, which is an effective tool for analyzing and categorizing social media content based on sentiment. The VADER algorithm calculates the polarity score of each tweet and assigns it to one of three categories: negative, neutral or positive. The compound score serves as a metric that calculates the summation of all lexicon scores, which are then normalized to a range between -1 and +1. Tweets with a compound score equal to or greater than 0.5 are classified as having a positive sentiment, while tweets with a compound score equal to or less than -0.5 are considered to have a negative sentiment. Tweets falling within the range of -0.5 to 0.5 are categorized as having a neutral sentiment. The formula to calculate the compound score is given by equation (1).

Compound score = sum of valence scores / sqrt(sum of valence scores^2 + alpha) (1)

where:

- *sum of valence scores*: the sum of the valence scores of each word in the text, where valence scores are positive, negative, or neutral scores of the words. The valence scores are obtained from the VADER lexicon, which contains a list of words along with their associated sentiment scores.
- *alpha*: a normalization factor that is added to the denominator to prevent division by zero and to offset extreme scores. The default value of alpha in the original VADER implementation is 15.

To calculate the compound score:

i. Tokenize the text into individual words.
ii. Look up each word in the VADER lexicon to retrieve its valence score. The valence score is a scalar value that ranges from -4 (extremely negative) to +4 (extremely positive) and is normalized to a range between -1 and +1.
iii. Calculate the sum of the valence scores for all the words in the text.
iv. Calculate the sum of the squared valence scores for all the words in the text.
v. Add the normalization factor alpha to the sum of squared valence scores.
vi. Divide the sum of valence scores by the square root of the sum of squared valence scores plus alpha.
vii. The result is the compound score, which represents the overall sentiment of the text.

The compound score ranges from -1 (extremely negative) to +1 (extremely positive) and indicates the degree of positivity or negativity of the text. The compound score is a useful metric for comparing the sentiment of different texts and for tracking changes in sentiment over time. However, it's important to note that the compound score is just one aspect of sentiment analysis and should be used in conjunction with other measures and tools to obtain a more complete understanding of the sentiment in the text. It's important to note that the VADER sentiment analysis tool may not always accurately capture the true sentiment of a text, especially in cases where sarcasm, irony. Valence scores can be assigned using a sentiment lexicon or dictionary, which is a collection of words with associated positive, negative, or neutral scores. In the case of VADER, it uses a pre-built lexicon that contains over 7,500 lexical features, including words, emoticons, acronyms, and other expressions.

The lexicon contains a list of words along with their associated sentiment scores. The sentiment scores are determined by multiple human judges who rated the words on a scale from -4 (extremely negative) to +4 (extremely positive), and then the scores are normalized to a range between -1 and +1. The lexicon also contains rules and heuristics to handle sentiment intensifiers, negations, and other linguistic features that can modify the sentiment of a text.

When analyzing a piece of text using VADER, the tool tokenizes the text into individual words and then looks up each word in the lexicon to retrieve its sentiment score. The tool also applies the lexicon's rules and heuristics to adjust the sentiment scores based on the context and linguistic features of the text. Finally, the tool calculates the overall sentiment of the text by aggregating the sentiment scores of all the words and generates a compound score that ranges from -1 to +1.

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 9*
*DOI: https://doi.org/10.17762/ijritcc.v11i9.8325*
***Article Received:*** *09 July 2023* ***Revised:*** *30 August 2023* ***Accepted:*** *15 September 2023*

*Step 5 : Feature engineering*

For converting words or sentences into vectors, we used embedding techniques. To create vectors for the input data, we employed the Word2Vec and Term Frequency-Inverse Document Frequency (TF-IDF) word embedding techniques. These techniques allowed us to generate vectors that will serve as inputs to the model. We have tried the model building Word2Vec with CNN_LSTM and TF-IDF with Random Forest model.

*A. Word Embedding*

The utilization of word embeddings has gained significant popularity in the fields of natural language processing and machine learning. Word embedding allows for the representation of words and documents in a numerical vector format. In this format, similar words are grouped together and have similar numeric expressions. This approach enables machine learning models to process natural language by understanding the meaning and context of words in a corpus.

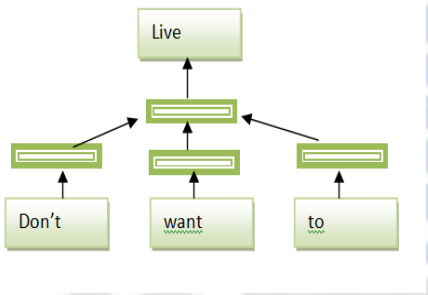


Figure 3. Word embedding

The use of word embeddings helps to reduce the high-dimensional and sparse representation of text data, which helps in extracting better features and improving the performance of machine learning models. These embeddings are learned using techniques such as Word2vec, GloVe, and fastText. The learned embeddings can be visualized to identify underlying patterns in the corpus. Overall, word embeddings have proven to be a useful approach for representing text data and enhancing the performance of machine learning models.

*1. Word2vec*

Word2vec is a highly popular word embedding technique that employs a neural network model to learn associations between words from extensive text corpora. This technique represents each unique word as a vector, capturing the semantic meaning and interrelationships among words. The word embeddings generated by Word2vec can be utilized for visualizing data patterns and clusters, as well as serving as inputs for machine learning models.

Word2vec provides two distinct models: Continuous Bag-of-Words (CBOW) and continuous skip-gram. In the CBOW model, the model takes a context window of adjacent words as input and predicts the current word. Conversely, the skip-gram architecture predicts the context words surrounding a given word. While the skip-gram model performs better with infrequent words, it is comparatively slower than the CBOW model.

For this research, the CBOW model of Word2vec was implemented using the gensim package. This implementation takes the context of each word as input and endeavors to predict the corresponding word. Such an approach facilitates capturing the inherent meaning and relationships within the dataset's words. Fig. 4 illustrates the internal functioning of the CBOW model.

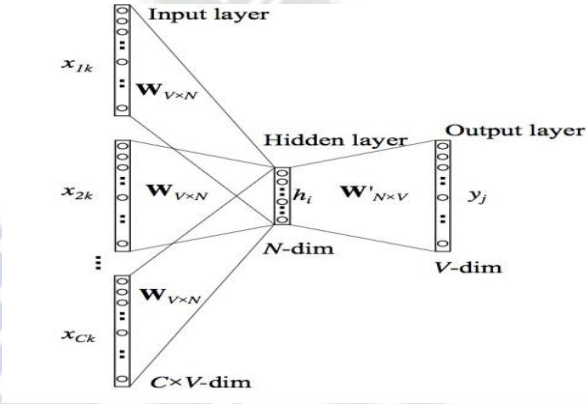


Figure 4. CBOW architecture

Within this architecture, two weight matrices, Wvxn and Wnxv, are employed. These matrices possess dimensions of vn and nv, respectively. The matrix Wvxn facilitates the mapping of the input x to the hidden layer, while Wnxv maps the outputs from the hidden layer to the final output layer. The architecture incorporates C context words, and its objective is to predict the target word. To accomplish this prediction, the encoding of the input word is utilized to calculate the output error in comparison to the encoding of the target word. During the training process, the vector representation of target words is learned. The hidden layer performs a weighted summation of inputs received from the input layer, subsequently copying the result to the output layer.

$$J(\theta) = \frac{1}{T}\log\left(L(\theta)\right) = -\frac{1}{T} \quad (2)$$

$$L(\theta) = \pi_{t=1}^{T}\pi_{-m\le j\le m}P(w_{t+J}|w_t;\theta) \quad (3)$$

In this context, L(θ) represents the likelihood of the context words, given a text sequence of length T. The word at a particular time step t is denoted as wt, and m refers to the size of the context window. We will simply take the log of the

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 9*
*DOI: https://doi.org/10.17762/ijritcc.v11i9.8325*
***Article Received:** 09 July 2023 **Revised:** 30 August 2023 **Accepted:** 15 September 2023*

equation and multiply it by -1 to compute the -ve log-likelihood in order to transform this equation into a form that makes it simple to take derivatives and turn it into a minimization problem. Equations (2) and (3) represent word vectorization for word2vec embedding. Word2Vec learns the semantic relationships between words by mapping them to high-dimensional vectors. The algorithm represents each word within a corpus as a vector in a high-dimensional space. This representation is designed in such a way that words with similar meanings are positioned closer to each other in the vector space.

Let us consider the sentence "My life is misery and done with it", To apply Word2Vec to the sentence "My life is misery and done with it", we first need to preprocess the sentence by removing any punctuation and converting all words to lowercase. The preprocessed sentence would be: "my life is misery and done with it".

Next, we would split the sentence into individual words or tokens: ["my", "life", "is", "misery", "and", "done", "with", "it"]. We can then use Word2Vec to learn the vector representations of each of these words based on their co-occurrence patterns in a larger corpus. Once we have successfully trained a Word2Vec model on a substantial corpus, we can leverage the trained model to obtain vector representations for each word within a given sentence. The vector representations can be obtained by simply passing each word through the trained Word2Vec model.

For example, assuming that the vector dimensionality of our Word2Vec model is 300, the vector representation of the word "life" might be: [0.12, 0.45, -0.23, ..., 0.67]. Likewise, we can obtain vector representations for the remaining words in the sentence using the same approach. Once we have acquired the vector representations of all words in the sentence, we can utilize these vectors to undertake diverse tasks. These tasks include calculating the similarity between words, identifying the most similar words to a given word, or clustering words with similar meanings together.

2. *Term Frequency-Inverse Document Frequency(TF-IDF)*

TF-IDF can be described as a method for assessing the relationship between words and a given text or document. The importance of a word is determined by both its frequency within the text and its rarity in the overall corpus or dataset. TF-IDF comprises two main components:

Term Frequency (TF): It quantifies the frequency of a specific term within a document. It is calculated by counting the number of occurrences of the term in the document. The higher the count, the higher the term frequency, is indicating a greater occurrence of the term within the document.

$$TF = \frac{\text{No.oftarget terms in the document}}{\text{Total no.ofterms in the document}} \quad (4)$$

Inverse Document Frequency (IDF): It assesses the significance of a term across the corpus. It is determined by calculating the logarithm of the ratio between the total number of documents in the corpus and the number of documents that contain the term. The TF-IDF score for a term in a document is obtained by multiplying its TF and IDF values together. A high TF-IDF score indicates that the term is important within the document, while a low score suggests that the term holds less importance. The TF-IDF approach finds extensive use in various applications, including information retrieval, text mining, and natural language processing. It aids in ranking documents based on their relevance to a given query and identifies crucial terms within a text.

$$IDF = \frac{\text{Total no.of documents}}{\text{No of documents target word occurs}} \quad (5)$$

Consider an example to work on the TF_IDF procedure
Sentence 1: "My life is misery and done with my life"

Tokenized Sentence 1: ["My", "life", "is", "misery", "and", "done", "with", "my", "life"]

Sentence 2: "I want to end my life"

Tokenized Sentence 2: ["I", "want", "to", "end", "my", "life"]

TABLE I. TF TABLE

| Term | Sentence 1 | Sentence 2 |
|---|---|---|
| My | 02-Sep | 01-Jun |
| life | 02-Sep | 01-Jun |
| is | 01-Sep | 0 |
| misery | 01-Sep | 0 |
| and | 01-Sep | 0 |
| Done | 01-Sep | 0 |
| With | 01-Sep | 0 |
| I | 0 | 01-Jun |
| Want | 0 | 01-Jun |
| to | 0 | 01-Jun |
| end | 0 | 01-Jun |

TABLE II. IDF TABLE

| Term | IDF |
|---|---|
| My | Log(2/1)=0.693 |
| life | Log(2/2)=0 |
| is | Log(2/1)=0.693 |
| misery | Log(2/1)=0.693 |
| and | Log(2/1)=0.693 |
| Done | Log(2/1)=0.693 |

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 9*
*DOI: https://doi.org/10.17762/ijritcc.v11i9.8325*
***Article Received:** 09 July 2023 **Revised:** 30 August 2023 **Accepted:** 15 September 2023*

| | |
|---|---|
| With | Log(2/1)=0.693 |
| I | Log(2/1)=0.693 |
| Want | Log(2/1)=0.693 |
| to | Log(2/1)=0.693 |
| end | Log(2/1)=0.693 |

The IDF value, in combination with the TF value, is multiplied to obtain the TF-IDF value. This value serves as a measure of the term's importance and is utilized for the vector representation of words.

*Step5: Model Building*

*Random Forest*

Random Forest is a machine learning algorithm that belongs to the ensemble learning family. It is composed of multiple decision trees that collaborate to make predictions. The class with the most votes from the individual trees in the random forest determines the final prediction of the model. This approach uses the Random Forest bagging method to avoid over fitting by using subsets of the original data to create predictions. Random Forest algorithm creates multiple decision trees (DTs) with different subsets of observations, making it suitable for both regression and classification problems. This algorithm involves creating subsets of the original data through row and feature sampling, building individual decision trees for each subset, and using Majority Voting for classification and average for regression problems to gain the final output.

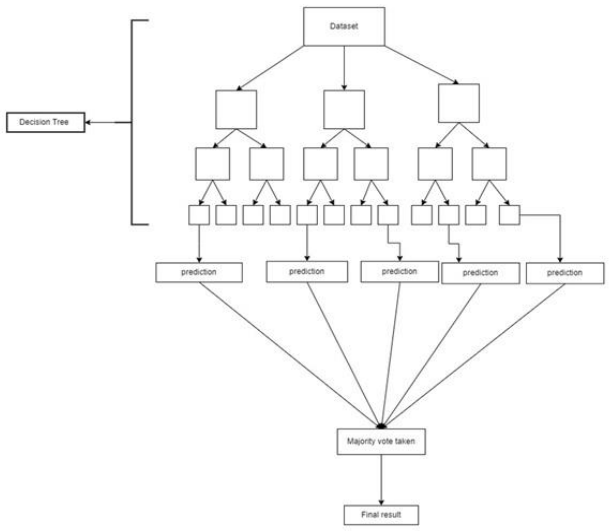


Figure 5. Random Forest

These are the steps involved in implementing the Random Forest Algorithm:

Step 1: Data Sampling- Create subsets of the original data by randomly selecting rows and columns, with replacement, to form subsets of the training dataset.

Step 2: Decision Tree Creation- Construct a decision tree for each subset of the data.

Step 3: Decision Tree Prediction- Each decision tree generates its own output or prediction based on the input data.

Step 4: Ensemble Prediction- For classification problems, the final output is determined through majority voting, where the most common prediction among the decision trees is chosen. In regression problems, the final output is determined by taking the average of the outputs from all the decision trees. By combining the predictions of multiple decision trees, Random Forest leverages their collective knowledge to make more accurate predictions. In Random Forest, parameters need to be defined for building decision trees:

- n_estimators: Number of trees in the forest.
- max_depth: Maximum depth of each decision tree node

*Hybrid model of CNN_LSTM*

*CNN*

Convolutional Neural Network (CNN) is a type of neural network that was first developed for image processing, and has shown excellent performance in classifying objects into pre-defined categories. When applied to NLP tasks, instead of images, text is represented as a one-dimensional array. The images, as in NLP tasks, a one-dimensional array is used to represent the text. Here, the CNNs' design is modified to use 1D convolutional and pooling procedures. Sentence classification, which involves categorizing a sentence into a set of predetermined categories by taking into consideration its n-grams that is, its words or sequence of words, as well as its characters or sequence of characters is one of the most common NLP tasks where CNN are used.

Convolutional Neural Networks involve two operations that can usually be thought of as feature extractors: convolution and pooling.When an input sequence is presented in the convolution layer, each element is linked to a dimension-based embedding vector. A kernel of size k is chosen, which is a sliding window of size k that moves across the sentence and applies the same convolutional filter as it moves, that is, it is the dot product between the concatenation of the embedding vectors and the weight vectors in a given window. In a Pooling layer a single, two-dimensional vector is created by pooling the vectors produced by many convolution windows. This is repeated by taking the maximum or average value noted in the vector that results from the convolutions.

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 9*
*DOI: https://doi.org/10.17762/ijritcc.v11i9.8325*
***Article Received:** 09 July 2023 **Revised:** 30 August 2023 **Accepted:** 15 September 2023*

This vector should ideally include all of the important elements of the sentence or document. This process is illustrated by Fig.6.

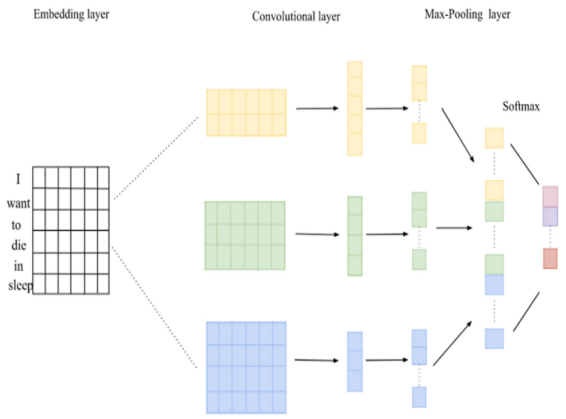


Figure 6. Convolutional Neural Networks

*Long Short Term memory (LSTM)*

To address the problem of latency in RNNs, LSTM networks were introduced as a solution to the vanishing gradient problem. Unlike traditional feed-forward neural networks, LSTMs incorporate feedback connections, allowing them to process entire sequences of data, such as time series, without analyzing each data point individually. Instead, LSTMs leverage prior sequence data stored as meaningful information to analyze new data points. They are particularly effective in handling data sequences like text, audio, and time-series.

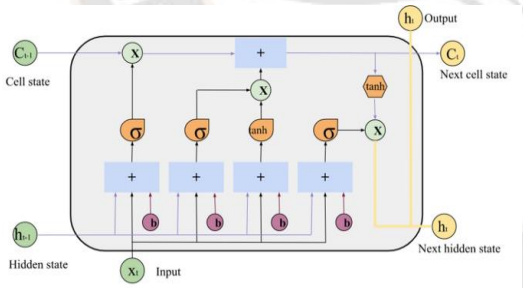


Figure 7. LSTM

At its core, an LSTM's output at any given moment is influenced by three key factors: the cell state (representing the network's long-term memory), the previous hidden state (output from the previous time point), and the current input data. LSTMs utilize several "gates" to control the flow of data into, out of, and within the network. These gates, implemented as separate neural networks, act as filters. The forget gate initiates the process by determining the relevance of parts of the cell state based on the previous hidden state and new input data. The input gate and the new memory network identify new information to be stored in the network's long-term memory. Finally, the output gate calculates the updated hidden state based on the revised cell state, previous hidden state, and new input data.

*Hybrid architecture of CNN –LSTM*

In the first step we will be sending our input to the embedding layer which has parameter of top words, dimensions of the matrix, and input length. Top words: it describes the top words which are present in the corpus. Fig. 8 provides an illustration of the model's workflow.

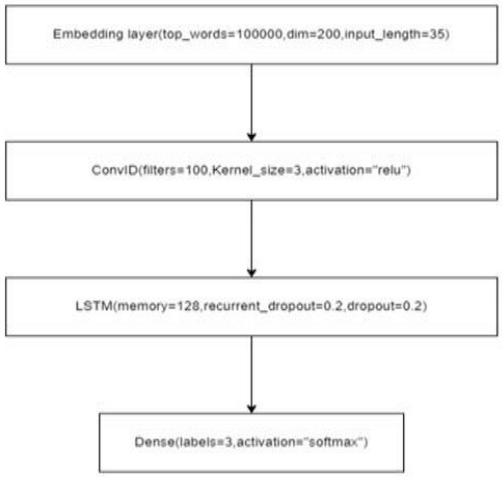


Figure 8. Hybrid architecture of CNN -LSTM

Layers description for hybrid model of CNN-LSTM

- Dimensions: represents the matrix size, it can be 100,200,300,400 etc
- Input length represents the tweet length, it does padding up to length of the tweet if the tweet size is minimum. In the second layer of model represents Conv1D layer which includes parameters like filters, kernel size, and activation function . During the training process of a CNN, the values for multiple filters in each convolution layer are learned from a specific training set. By the end of training, these filters have distinct values that enable them to identify specific features in the dataset. Kernel size describes the how many inputs are looked at once while training
- Activation function ReLU (Rectified Linear Unit) understands the relationship between input and output, it understands the output of neuron and ignores positive label and represents 0 for negative.
- In the third layer we add LSTM layer where we have parameters like memory size and dropout layer ,Memory size represents the size of the memory used for storing the important information Dropout layer used for reducing the over fitting the data.

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 9*
*DOI: https://doi.org/10.17762/ijritcc.v11i9.8325*
*Article Received: 09 July 2023 Revised: 30 August 2023 Accepted: 15 September 2023*

- The fourth layer is the dense layer where actual classification takes place .We adds the number of labels we have and also represent activation function; we have used softmax activation function.

*Step 6: Building the website*

We created a website which acts as a depression watch system where person submits his/her Twitter username, we will try to analyze the tweets posted by the person, and check for negative tweets if any, We will be sending a reply message to the tweet posted by them and also the alerts the followers of the person stating the person is struggling with depression and also detect the thoughts of the user posted in our website and sends personal message if person finds to be depressed. We also add about us page where the details of our team will be displayed and query page to share their queries to us. After training our collected data using a hybrid model of CNN_LSTM and Random Forest, we obtained an accuracy of 89.4%. The hybrid model of CNN_LSTM outperformed the Random Forest model in terms of accuracy.

## IV. RESULTS

To assess the performance of a machine learning model during training and validation, accuracy and loss graphs are commonly used visualizations. The accuracy graph presents the model's ability to correctly classify data throughout the training process. It is usually plotted as a line graph, with training accuracy on the y-axis and the number of training epochs on the x-axis. Typically, the accuracy increases during training, reaching an optimal level at which it levels off, which is shown in Fig.9.

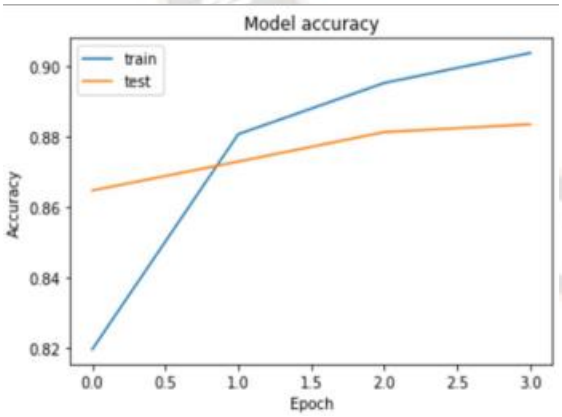


Figure 9. Accuracy graph for Hybrid CNN_LSTM

In contrast, the loss graph depicts the degree to which the model is minimizing the difference between its predicted values and the actual values of the training data.

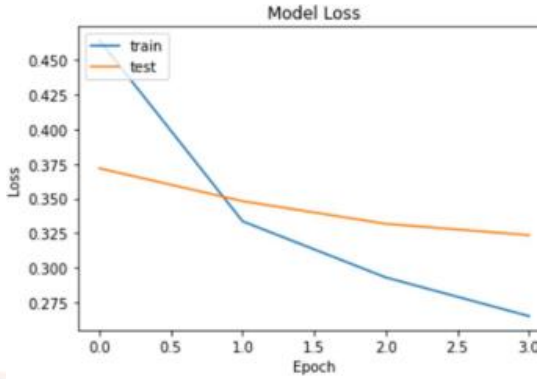


Figure 10. Loss graph for Hybrid CNN_LSTM

The loss function quantifies the discrepancy between predicted and true values, and the goal of the model is to minimize this function. The loss graph is shown in Fig.10, is also presented as a line graph, with training loss on the y-axis and the number of training epochs on the x-axis. The ideal scenario is for the training loss to decrease over time as the model becomes more skilled at making accurate predictions. The Fig.11 shows a comparison graph of accuracies of the models implemented.

By comparing the accuracy scores of the two algorithms presented in Fig. 11, it can be observed that the Hybrid model of CNN-LSTM surpasses the TF-IDF with Random Forest approach. The Hybrid model achieves an accuracy of 90%, while the TF-IDF with Random Forest achieves 85% accuracy. This suggests that the Hybrid model exhibits better accuracy in predicting the target variable compared to the TF-IDF with Random Forest approach. The higher accuracy of the Hybrid model indicates its ability to more effectively classify and analyze the data, leading to improved predictions.

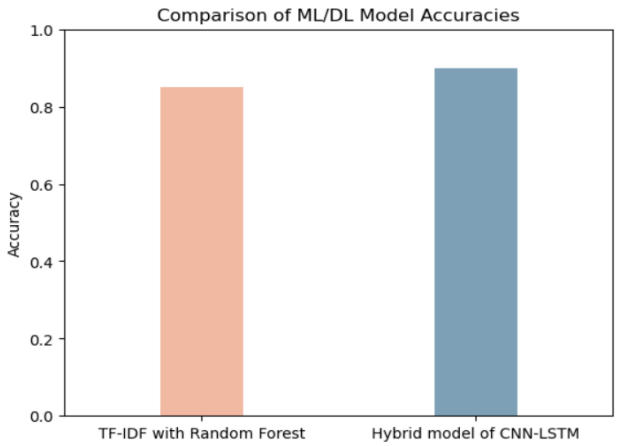


Figure 11. Comparison Graphs

The 5% difference in accuracy between the two algorithms emphasizes the superior performance of the Hybrid model.

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 9*
*DOI: https://doi.org/10.17762/ijritcc.v11i9.8325*
***Article Received:** 09 July 2023 **Revised:** 30 August 2023 **Accepted:** 15 September 2023*

This difference can be significant, particularly in applications where accurate predictions are crucial. The higher accuracy rate of the Hybrid model highlights its potential to provide more reliable and precise results.

Table 3Comparison table

| Algorithms | Accuracy | F1-Score | Precision | Recall |
|---|---|---|---|---|
| TF-IDF with Random Forest | 85% | 79% | 87% | 75% |
| **Hybrid model of CNN-LSTM** | **90%** | **90%** | **91%** | **90%** |

Based on the given Table3, we can observe that the hybrid model of CNN-LSTM outperforms TF-IDF with Random Forest in terms of accuracy, F1-Score, precision, and recall. The hybrid model achieved an accuracy of 90%, while TF-IDF with Random Forest achieved an accuracy of 85%. Similarly, the hybrid model demonstrated higher values for F1-Score, precision, and recall compared to TF-IDF with Random Forest. This indicates that the hybrid model of CNN-LSTM is more effective in accurately predicting the target variable compared to TF-IDF with Random Forest.

## V. FUTURE WORK

We will also be working on hybrid models to improve the model's efficiency and performance. Building the Bot and creating a profile for the user based on their thoughts for easy monitoring. To effectively raise awareness and facilitate the early diagnosis of individuals facing depression, it is important to send support messages to their family and friends. This approach aims to minimize the impact on the affected person's mental health while maximizing the chances of timely intervention. By leveraging data from various social media platforms like Instagram and Facebook, along with the utilization of genetic algorithms for model training, a comprehensive and effective system can be developed. This system would enable the identification and support of individuals at risk, ultimately improving their overall well-being. We will also investigate how to detect depression using other types of data in order to assist people outside of social media platforms.

## VI. CONCLUSION

In this study, we focused on utilizing tweets and comments from social media platforms to understand people's mental health conditions. We developed two models, a Random Forest model with TFIDF and a hybrid deep learning model called CNN-LSTM that used word2vec. The CNN-LSTM model performed better, achieving an accuracy rate of 89.4% and also created a user interface to analyze Twitter profiles based on posted tweets, identifying individuals who may be depressed. If flagged, an automated reply message is sent to offer help. This research contributes to raising awareness about mental health by effectively using social media data and advanced machine learning techniques to detect and intervene early. We examined 149,000 depressed tweets collected through keyword searches and found 45,706 posts from individuals predisposed to depression. The developed models can predict the likelihood of self-harm and serve as a depression monitor. Additionally, this analysis can be used to monitor Twitter users and identify those who express their emotions, such as depression, on the platform, providing them with online consultation from Twitter psychiatrists to help overcome their depression.

## CONFLICT OF INTEREST

The authors have no conflicts of interest to disclose.

## REFERENCES

[1] Chatterjee R, Gupta RK, Gupta B. Depression detection from social media posts using multinomial naive theorem. IOP Conf Ser Mater Sci Eng. 2021;1022(1). doi: 10.1088/1757-899X/1022/1/012095.

[2] Alorini G, Rawat DB, Alorini D. LSTM-RNN Based Sentiment Analysis to Monitor COVID-19 Opinions using Social Media Data. IEEE International Conference on Communications. 2021. doi: 10.1109/ICC42927.2021.9500897.

[3] Basiri ME, Nemati S, Abdar M, Cambria E, Acharya UR. ABCDM: An Attention-based Bidirectional CNN-RNN Deep Model for sentiment analysis. Future Gener Comput Syst. 2021;115:279-294. doi: 10.1016/j.future.2020.08.005.

[4] Behl S, Rao A, Aggarwal S, Chadha S, Pannu HS. Twitter for disaster relief through sentiment analysis for COVID-19 and natural hazard crises. Int J Disaster Risk Reduct. 2021;55. doi: 10.1016/j.ijdrr.2021.102101.

[5] Kour H, Gupta MK. An hybrid deep learning approach for depression prediction from user tweets using feature-rich CNN and bi-directional LSTM. Multimed Tools Appl. 2022;81(17):23649-23685. doi: 10.1007/s11042-022-12648-y.

[6] Hameed Z, Garcia-Zapirain B. Sentiment Classification Using a Single-LayeredBiLSTM Model. IEEE Access. 2020;8:73992-74001. doi: 10.1109/ACCESS.2020.2988550.

[7] Alsagri HS, Ykhlef M. Machine learning-based approach for depression detection in twitter using content and activity features. IEICE Trans Inf Syst. 2020;E103D(8):1825-1832. doi: 10.1587/transinf.2020EDP7023.

[8] Shetty NP, Muniyal B, Anand A, Kumar S, Prabhu S. Predicting depression using deep learning and ensemble algorithms on raw twitter data. Int J Electr Comput Eng. 2020;10(4):3751-3756. doi: 10.11591/ijece.v10i4.pp3751-3756.

[9] Shetty NP, Muniyal B, Anand A, Kumar S, Prabhu S. Predicting depression using deep learning and ensemble algorithms on raw twitter data. Int J Electr Comput Eng.

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 9*
*DOI: https://doi.org/10.17762/ijritcc.v11i9.8325*
***Article Received:** 09 July 2023 **Revised:** 30 August 2023 **Accepted:** 15 September 2023*

2020;10(4):3751-3756. doi: 10.11591/ijece.v10i4.pp3751-3756.

[10] Trotzek M, Koitka S, Friedrich CM. Utilizing Neural Networks and Linguistic Metadata for Early Detection of Depression Indications in Text Sequences. IEEE Trans Knowl Data Eng. 2020;32(3):588-601. doi: 10.1109/TKDE.2018.2885515.

[11] Sarah Taher Y. Al- lami, Ali A. F. Al- Hamadani. (2023). Systematic Review for Comparison Type of Pulse Tube Refrigerator. International Journal of Intelligent Systems and Applications in Engineering, 11(4s), 625–633. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2740

[12] Bhargava C, Poornima S, Mahur S, Pushpalatha M. Depression Detection Using Sentiment Analysis of Tweets. Turk J Comput Math Educ. 2021;12.

[13] Kaur H, Ul Ahsaan S, Alankar B, Chang V. A Proposed Sentiment Analysis Deep Learning Algorithm for Analyzing COVID-19 Tweets. Published. doi: 10.1007/s10796-021-10135-7.

[14] P. P. A. Performance Evaluation and Comparison using Deep Learning Techniques in Sentiment Analysis. J Soft Comput Paradigm. 2021;3(2):123-134.

[15] Divyapushpalakshmi M, Ramalakshmi R. An efficient sentimental analysis using hybrid deep learning and optimization technique for Twitter using parts of speech (POS) tagging. Int J Speech Technol. 2021;24(2):329-339. doi: 10.1007/s10772-021-09801-7.

[16] Joshi ML, Kanoongo N. Depression detection using emotional artificial intelligence and machine learning: A closer review. Mater Today Proc. 2022;58:217-226. doi: 10.1016/j.matpr.2022.01.467.

[17] Kwame Boateng, Machine Learning in Cybersecurity: Intrusion Detection and Threat Analysis , Machine Learning Applications Conference Proceedings, Vol 3 2023.

[18] Babu NV, Kanaga EGM. Sentiment Analysis in Social Media Data for Depression Detection Using Artificial Intelligence: A Review. SN Comput Sci. 2022;3(1). doi: 10.1007/s42979-021-00958-1.

[19] Guntuku SC, Ramsay JR, Merchant RM, Ungar LH. Language of ADHD in adults on social media. J Attention Disord. 2019 Oct;23(12):1475-1485.

[20] Almouzini S, Alageel A. Detecting Arabic depressed users from Twitter data. Procedia Comput Sci. 2019;163:257-265.

[21] Yang K, Zhang T, Ananiadou S. A mental state Knowledge-aware and Contrastive Network for early stress and depression detection on social media. Inf Process Manag. 2022 Jul 1;59(4):102961.

[22] Kang X, Shi X, Wu Y, Ren F. Active learning with complementary sampling for instructing class-biased multi-label text emotion classification. IEEE Trans Affect Comput. 2020 Nov 16.

[23] World Health Organization. The World Health Report 2001: Mental health: new understanding, new hope.

[24] Elbagir S, Yang J. Twitter sentiment analysis using natural language toolkit and VADER sentiment. InProceedings of the international multiconference of engineers and computer scientists 2019 Mar 13 (Vol. 122, p. 16).

[25] Goyal P, Hossain KSM, Deb A, Tavabi N, Bartley N, Abeliuk A, Ferrara E, Lerman K. Discovering Signals from Web Sources to Predict Cyber Attacks. 2018.

[26] Liu Q, He H, Yang J, Feng X, Zhao F, Lyu J. Changes in the global burden of depression from 1990 to 2017: Findings from the Global Burden of Disease study. Journal of psychiatric research. 2020 Jul 1;126:134-40.

[27] Kwon HJ, Ban HJ, Jun JK, Kim HS. Topic modeling and sentiment analysis of online review for airlines. Information. 2021 Feb 12;12(2):78.

[28] Wangsan K, Chaiear N, Sawanyawisuth K, Klainin P, Simajareuk K. Pattern of shiftwork and health status among nurses in a university hospital in Northeastern Thailand. Asia-Pacific Journal of Science and Technology. 2019 Jun 18;24(2).