Mental Health Analyzer for Depression Detection Based on Textual Analysis

Pranav Bhat
Electronics and Telecommunication, Maharashtra Institute of Technology, Savitribai Phule Pune University, Pune, Maharashtra, India
Email: pranavdbhat123@gmail.com

Alwin Anuse, Rupali Kute, R. S. Bhadade, and Prasad Purnaye
Vishwanath Karad MIT-World Peace University, Pune, Maharashtra, India
Email: {alwin.anuse, rupali.kute, raghunath.bhadade, prasad.purnaye}@mitwpu.edu.in

***Abstract*—The global coronavirus pandemic and lockdown has had negative impacts on individuals' mental health and well-being. The crisis has generated symptoms of depression in many, which may last even after the lockdown is over. To provide support to individuals in terms of counseling and psychiatric treatment, it is necessary to identify such depressive symptoms in a timely fashion. To address this problem, an artificial intelligence-based system is proposed to assess the changes, if any, in the mental health of an individual as a function of time, starting from the pre-lockdown period (in India from 20 April 2020). A Mental Health Analyzer has been implemented to automatically detect whether an individual is trending toward a state of depression based on his or her tweets over time. The deep learning models of Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Bidirectional LSTM have been implemented and compared for the emotion classification task, specifically to detect the emotions of sadness, fear, anger, and joy present in a person's tweets. The system identifies the emotion of sadness present in tweets to detect depression. An ensemble maximizing model using CNN, LSTM, and Bidirectional LSTM is proposed to maximize the recall metric to improve the performance for the task of depression detection. The implemented system was tested using the dataset provided for the SemEval-2018 semantic evaluation tasks and achieves better results than previous models for the task of emotion classification and, further, can detect depression when tested on real Twitter data.**

***Index Terms*—depression detection, artificial intelligence, deep learning, emotion prediction, mental health analyzer**

I. Introduction

The coronavirus crisis has led countries throughout the world to resort to extreme social measures. As people remain in stay-at-home quarantine and isolation to prevent possible infection, it is important to consider the effects of the pandemic on individuals' mental health. People may experience stress, anxiety, fear, loneliness, and even depression requiring intervention during the lockdown period due to isolation, physical distancing, and the closure of workplaces. Mental health problems developed during the lockdown may persist in the post-lockdown period. It would be beneficial to be able to identify depressive symptoms in individuals in a timely fashion to provide them psychiatric treatment support. The system described in this paper can be used to support such a depression detection process by automatic analysis of individuals' posts on social media platforms.

Twitter, a microblogging social network platform, allows users to publicly express their opinions and emotions on any issue or topic. Users have the freedom to tweet on matters that they feel are important. Hence, users' tweets can be used to decode their emotions. This can be easily done by humans who have a contextual understanding of the tweets.

Many machine learning classifiers use the occurrence and frequency of words for sentiment and emotion classification. As a result, texts that do not use specific words particular to an emotion may be wrongly classified. For machine learning models not to depend solely on statistical measures, deep learning models should be used, which take into consideration a sequence of words and, hence, understand the contextual meaning of the words. This improves the classification accuracy on real-time data wherein the emotion in tweets may require the model to understand hidden or latent meanings by understanding the context of the text.

Tweets by a user can help us understand the emotions of the user, and specifically, the sadness emotion evidenced in tweets can help to detect depression in an individual. The task of classifying tweets into the emotions they represent is a multilabel classification task, as a single tweet may represent more than one emotion, such as fear and sadness. The Mental Health Analyzer system presented here performs multilabel classification using deep learning to classify a user's tweets into the emotions of sadness, anger, fear, and joy. According to the various definitions of the term Depression, available in scientific literature and the research work quoted in the literature survey in Section II 'Sadness' is one of the most critical factors detrimental in predicting Depression. The World Health Organization clearly states while defining Depression, that it is characterized by persistent sadness

Manuscript received July 23, 2021; revised December 2, 2021.

doi: 10.12720/jait.13.1.67-77

[1]. Stress and Fear have also been observed as factors leading to depression as cited in the literature survey [2], [3]. As Sadness is one of the most important and critical factors in determination of Depression, the plot of Sadness emotion can be used to detect a person moving towards the state of depression. Also, the plot of fear emotion can help in further evaluating the mental state of the individual. The proposed system analyzes the mental state of the user based on the person's series of tweets over a period to discern the trend of emotions.

The proposed system uses distributed word representations for encoding the words in a tweet based on their semantic similarity. This further helps the deep learning models to better predict the output and, unlike baseline machine learning models, does not purely rely on the occurrence of specific words or n-grams. Further, the system uses an ensemble maximizing algorithm based on CNN, LSTM, and Bidirectional LSTM, which improves the recall metric and maximizes the chances of depression detection.

Traditional approaches and deep learning approaches for sentiment and emotion classification task are surveyed in Section II. Section III presents the proposed system along with the components, steps, and output of the system. The experimentation carried out and the performance of the deep learning models used in the system are presented in Section IV, and Section V provides the results achieved by the Mental Health Analyzer and depression detection system.

II. Related Work

Various definitions are reported in the open scientific literature, for the term 'Depression' and all of these include Sadness as a critical aspect. Mouchet-Mages *et al.* [2] concludes after studying the relation between sadness and depression, that it is justifiable to consider sadness as a clinical core symptom of depression, and to properly assess and track it. Forbes *et al.* [3] concludes that people suffering from post-traumatic stress disorder, often show high levels of fear and anxiety. Rodríguez-Hidalgo *et al.* [4] studies the scientific literature such as [5], which establishes relationships between stress, anxiety, and depression, and highlights the critical role relationships between fear, stress, and anxiety can play in the development of depression symptoms and how they can be used to track it. Most of the systems that have been proposed for the task of sentiment and emotion detection and classification use machine learning classifiers, such as Support Vector Machine (SVM) and Naïve Bayes. These baseline models rely upon the co-occurrence and frequency of words in texts. Deep learning models tend to perform better than traditional approaches. The following sections survey traditional approaches and deep learning approaches for various sentiment and emotion classification tasks.

A. Traditional Approaches for Sentiment and Emotion Classification

A survey of various approaches for sentiment analysis of Twitter data is presented in [5]. The survey is based on a dataset with positive, negative, or neutral sentiments. The survey compares various machine learning algorithms, such as Naïve Bayes, SVM, and Max Entropy. The paper summarizes the best scores of the models after testing for various cases such as unigrams and bi-grams and their combinations.

SVM is used in [6] by applying categorization only to subjective parts of a document, and an accuracy of 86.4% is achieved. Multiclass SVM and adaptive co-training algorithms are compared in [7] for positive and negative sentiment classification for different topics. The maximum accuracy of 82.52% was obtained using the co-training SVM algorithm. SVM and Naïve Bayes are compared in [8] using sentiment analyzers with machine learning. SVM provided a maximum accuracy of 62.67 when trained with TextBlob, while Naïve Bayes gave a maximum accuracy of 79% when trained with Word Sequence Disambiguation (WSD).

As reported in [9], training a recursive neural tensor network on the Stanford Sentiment Treebank corpus achieved an accuracy of 85.4% in positive and negative sentiment classification.

A classifier to predict polarity based on context for subjective phrases is proposed in [10], based on various features. The maximum accuracy achieved on all the features was 84.08%. Use of a tree kernel and POS features are explored in [11] and achieved a maximum accuracy of 75.39% for positive and negative sentiment classification on Twitter data. Table I shows the accuracy scores achieved by different models for the task of sentiment classification.

TABLE I. Accuracy Score of Models for Positive and Negative Sentiment Classification

Model	Accuracy
Naïve Bayes [5]	76.44
Max Entropy [5]	74.93
SVM [5]	77.73
SVM (only subjective part) [6]	86.4
Co-training SVM [7]	82.52
SVM Sentiment Treebank [9]	85.4
SVM and TextBlob [8]	62.67
Naïve Bayes and WSD [8]	79

Mohammad *et al.* [12] presented the task of inferring the emotional state of a person from his or her tweets, including subtasks such as inferring emotional intensity regression and emotion classification. Multiple teams participated in the shared task. The maximum accuracy achieved in the emotion classification task was 58.8% with a maximum F1 score of 0.701.

Abbasi *et al.* [13] applied an entropy weighted genetic algorithm along with SVM for sentiment analysis-based classification of web forums in English and Arabic. This approach yielded accuracies of over 91% on the benchmark dataset.

Lu *et al.* [14] proposed a framework to combine information from different sources to learn a sentiment lexicon based on context given unlabeled texts pertaining

to certain opinions. This learned corpus was then used for the sentiment classification task.

Wilson *et al.* [15] created a dictionary to distinguish between neutral and polar expressions and determine their polarity for phrase-level sentiment analysis.

B. Sentiment and Emotion Classification Using Deep Learning

Bengio *et al.* [16] used a model to learn distributed word representations along with neural networks to determine the probability functions of word sequences and generalize these representations for unseen texts.

Mikolov *et al.* [17] introduced a novel model named skip-gram to improve accuracy and reduce computational cost of learning continuous word vector representations. The same authors [18] implemented extensions of the model to increase the speed of training on large datasets and provided alternative simple techniques.

Tang *et al.* [19] applied sentiment-specific word embedding to the sentiment classification task by creating word representations taking into consideration the words' polarity and sentiment using neural networks.

Glorot *et al.* [20] used high-level features to train a sentiment-based classifier in an unsupervised approach to tackle the problem of domain adaptation and performed emotion classification of online reviews from different domains.

Wehrmann *et al.* [21] used CNN for sentiment analysis of a Twitter dataset involving multiple language datasets for training purposes.

Cambria *et al.* [22] used CNN for extracting visual features for sentiment analysis of multimodal data, while [23] used CNN for high-level feature extraction for the task of semantic labeling of images.

Tian *et al.* [24], [25] used CNN-based models for extracting features. Tian *et al.* [26] used CNN to obtain trained word vectors by training the distributed word embeddings on each word and used the features obtained for emotion analysis.

Kim [27] used CNN over pre-trained word vectors for text classification, achieving higher accuracy than previous methods.

Ren *et al.* [28] used word embedding vectors based on contextual features, taking into consideration only relevant tweets, for the task of Twitter-based sentiment analysis.

Arora *et al.* [29] applied text normalization using a neural network for deep convolutional character level embedding (Conv-char-Emb) for sentiment analysis of unstructured data.

Satapathy *et al.* [30] applied microtext normalization using the deep learning architectures of LSTM, CNN with LSTM, attentive LSTM, and an attentive Gated Recurrent Unit (GRU) and achieved improved results for sentiment analysis.

Wang *et al.* [31] used CNN consisting of convolution, pooling, and concatenation layers for extracting features that were given as sequences to Recurrent Neural Networks (RNNs) to understand long-term dependencies. The RNNs used were LSTM and GRU and both achieved good results.

Previous work offers promising results for deep learning models as compared to traditional machine learning models. Hence, we propose a system using an ensemble maximizing model using CNN, LSTM, and Bidirectional LSTM for emotion classification and depression detection.

III. Proposed System

Fig. 1 portrays the components, inputs, and outputs of the proposed Mental Health Analyzer system. The system block diagram is explained in detail in the following subsections.

A. Dataset

To implement a depression detection system based on tweets, a dataset with tweets labeled with emotions is required. The SemEval-2018 Task 1 [11] dataset for Emotion Intensity Regression provides a dataset with tweets labeled with the following emotions, anger, fear, joy, and sadness, along with the intensity of each emotion.

To use these data for classification of emotions, the implementation uses only the tweets with an intensity score above 0.5, where the maximum score possible is 1. The proposed system has the aim to detect depressing tweets. The number of tweets in the original dataset are considerably high and the aim of the dataset with the intensity label in SemEval-2018 Task 1 was to estimate the level of intensity which was a regression task unlike the system proposed in our manuscript. As the proposed system uses a classifier, removing tweets with emotion intensity below 0.5 ensures that only those tweets which clearly fall in the respective emotions remain, for training the model. The total number of tweets in the dataset after removing the tweets with intensity less than 0.5 is 3535 and the number of tweets categorized to each emotion in the dataset is shown in Table II. The categorized dataset is visualized in Fig. 2. The research work uses SemEval-2018 Task 1 Mohammad *et al.* (2018) dataset, which includes tweets along with its labelled emotions and intensity. The dataset was created without user label when it was created for the task in SemEval-2018 Task, so that training and testing is user-independent and should not overfit or fit only for the users in the dataset. The research work uses TensorFlow and Keras to build and train the models and uses standard train-test split function which randomly split the dataset into given proportions, independent of any conditions.

TABLE II. Dataset Emotion Categories

Emotion	Number of tweets
Anger	834
Fear	1115
Joy	821
Sadness	764

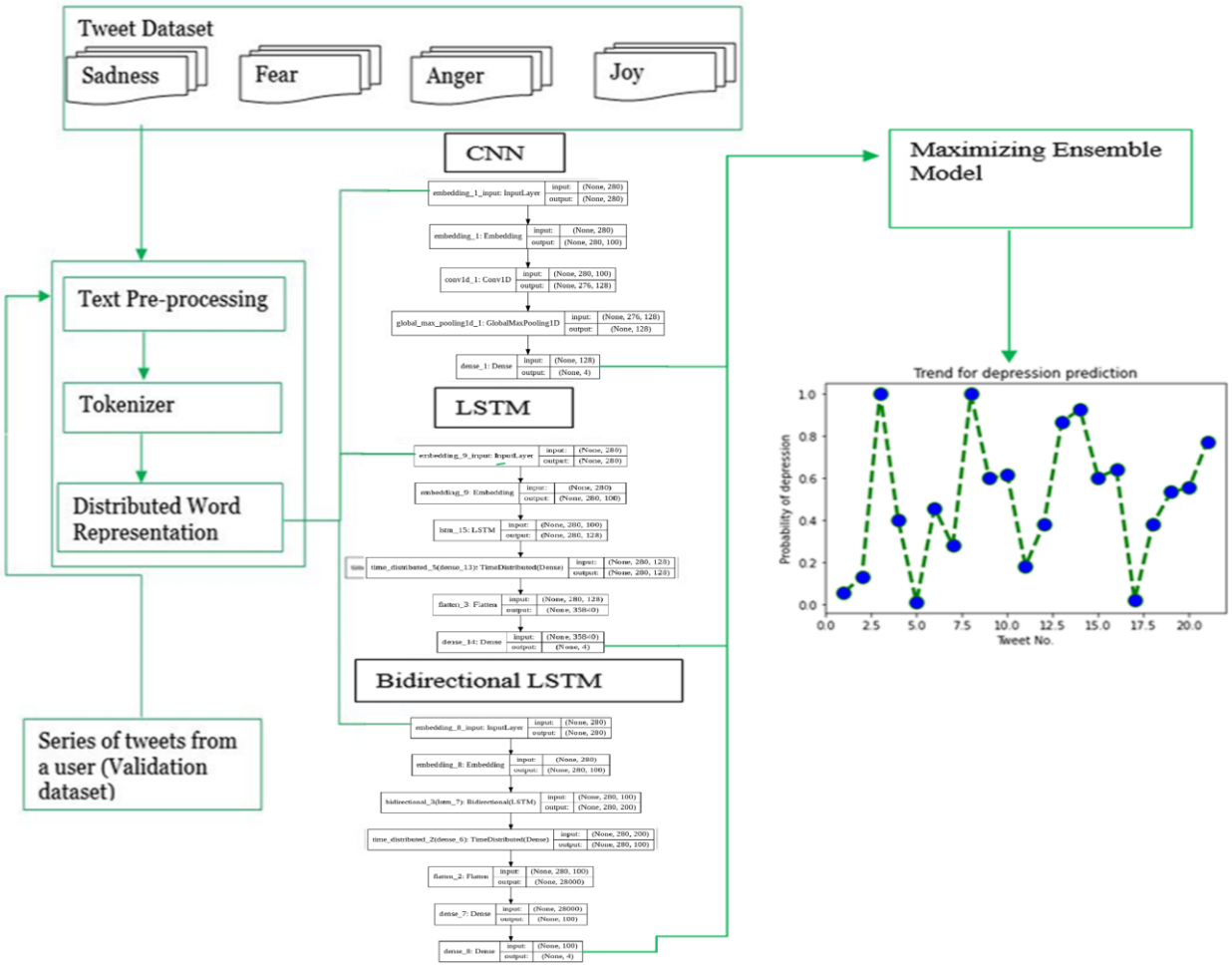

Figure 1. System block diagram of the mental health analyzer.

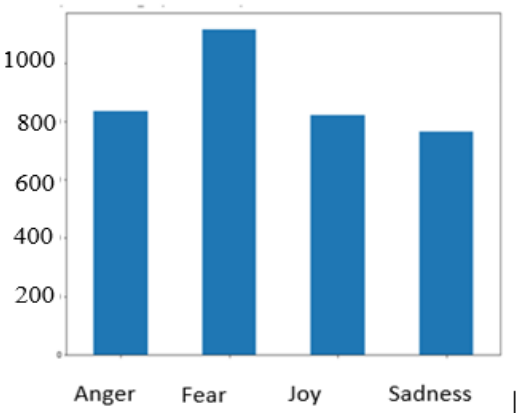

Figure 2. The dataset by category.

B. Text Preprocessing

All the words in the tweets must be tokenized as part of the text preprocessing before using them to extract features. Tweets in the corpus do not belong to a specific topic. The tweets contain punctuation marks and unnecessary characters that need to be removed.

C. Tokenization and Vectorization

Tokenization is the process of converting the text, in this case the tweets, into tokens. The tokenizer class of the Keras Text Processing code package is used to tokenize the tweets. Words are tokenized and indexed as a dictionary. Punctuation marks and other unnecessary characters are removed from the original text. The tokenizer vectorizes and indexes the tokens, returning a sequence for the text present in the training and testing set.

The maximum number of words to be tokenized and indexed is set to 5000. The vocabulary length is set to the number of words indexed + 1.

The sequences are padded to ensure that all sequences are of the same size; that is, sequences that are smaller than the predefined maximum length are padded. The maximum allowed length of any individual tweet is set by Twitter to be 280 characters.

D. Distributed Word Representation

In distributed word representations, each entry in the vector of words represents a hidden feature of the meaning of the word. Distributed word representations are employed to use the semantic and syntactic dependencies of the words in the corpus to better understand the context of a sentence and the relation between words.

GloVe, which stands for global vectors for word representation, is used to create a co-occurrence matrix, and a conditional probability is calculated for each word [32]. Word vectors trained using the GloVe algorithm on two billion tweets were used by the system as input to the

downstream components. The system uses 100-dimensional word representations.

Each word is processed using the tokenizer dictionary and an embedding matrix is created with an index for each word. The embedding matrix stores the GloVe representation of each word at its respective index.

GloVe has been observed to give better results as compared to baseline vectorizers. Also GloVe weights used in the research are pretrained on 2.7 billion tweets and hence GloVe is seen to better suited to gauge the semantic relation between the words for tweet related vectorization and extracting further insights.

E. Deep Learning Models

The problem of depression detection requires the model to understand the context of the words in a tweet. General statistical machine learning classifiers, which depend on statistical measures to perform the classification task, have not been able to achieve a high degree of accuracy. To solve this problem, this system uses CNN, LSTM, and Bidirectional LSTM models to understand and take into consideration the latent features in the tweets for the classification task.

1) CNN model

The CNN model has an embedding layer, a 1D convolutional layer, a global max pooling layer, and a fully connected layer with four outputs. The embedding layer is given the GloVe embedding matrix as weights and is frozen for the training process. The fully connected layer at the end of this process uses a sigmoid activation function, as the problem is a multilabel classification problem because a tweet may represent more than one emotion.

The layers used, the shape of the outputs of each layer, the parameters for each layer of the deep learning model, and the total trainable and non-trainable parameters for the implementation of the CNN model are shown in Table III.

TABLE III. CNN MODEL LAYERS AND PARAMETERS

Layer (type) (Output shape)	Parameter #
Embedding_1 (Embedding) (None, 280, 100)	984700
conv1d_1 (Conv1D) (None, 276, 128)	64128
global_max_pooling1d_1 (Glob (None, 128)	0
dense1 (Dense) (None, 4)	516

Total parameters: 1,049,344
Trainable parameters: 64,644
Non-trainable parameters: 984,700

2) LSTM model

The LSTM model is implemented using the LSTM layer from Keras and is called after the embedding layer. The LSTM layer is set to return sequences to obtain the output for each timestamp.

The TimeDistributed wrapper from Keras is used for the first fully connected layer. The output is then flattened and given to a fully connected layer with four final outputs using the sigmoid activation function.

The layers, the shape of the outputs of each layer, the parameters for each layer of the deep learning model, and the total trainable and non-trainable parameters for the implementation of the LSTM model are shown in Table IV.

TABLE IV. LSTM MODEL LAYERS AND PARAMETERS

Layer (type) (Output shape)	Parameter #
embedding 1 (Embedding) (None, 280, 100)	984700
Lstm_1 (LSTM) (None, 280, 128)	117248
Time_distributed_1 (TimeDist (280, 128)	16512
Flatten_1 (Flatten)	0
dense1 (Dense) (None, 4)	143364

Total parameters: 1,261,824
Trainable parameters: 277,124
Non-trainable parameters: 984,700

3) Bidirectional LSTM model

The Bidirectional LSTM model is implemented using the Bidirectional layer from Keras as a wrapper for the LSTM layer. The outputs from the forward and backward passes of the LSTM layer are concatenated by the Bidirectional layer using the merge mode parameter. The rest of the implementation of the model is similar to that of the LSTM layer. The different layers of the model, along with each respective type, output shape, and the total trainable and non-trainable parameters, are given in Table V.

TABLE V. BIDIRECTIONAL LSTM MODEL LAYERS AND PARAMETERS

Layer (type) (Output shape)	Parameter #
embedding 1 (Embedding) (None, 280, 100)	984700
Bidirectional_1 (Bidirection (None, 280, 200)	160800
Time_distributed_2 (TimeDist (280, 100)	20100
Flatten_1 (Flatten)	0
dense1 (Dense) (None, 100)	2800100
dense1 (Dense) (None, 4)	404

Total parameters: 3,966,104
Trainable parameters: 2,981,404
Non-trainable parameters: 984,700

4) Maximizing ensemble model

Fear and sadness are the two emotions critical to depression detection and analysis. Hence, it is crucial to maximize the detection chances of these two emotions in our model. This is accomplished by using a maximizing ensemble of the CNN, LSTM, and Bidirectional LSTM models wherein the maximum probability for each emotion is selected from the predictions made by the three models. Expressed mathematically, Emotion probability = Maximum (prediction probability by CNN, prediction probability by LSTM, prediction probability by Bidirectional LSTM).

This approach helps to maximize the recall for all the emotion classes and, hence, better detect emotions in all the tweets, including ones that any of the models may wrongly classify while another model may classify correctly.

F. Mental Health Analyzer and Depression Detection

A series of tweets from a user is used for the model that detects the emotions present in the tweets. The user is a

sample user and is not a part of the training or test set. The user was selected randomly from the internet by curating a search for consecutive highly depressing tweets from a random user. These tweets from the user are used to display the end result of the proposed system The depression vs. tweets graph is plotted to visualize and analyze any trend in depression that may occur and thereby understand whether the person is depressed or trending toward a state of depression. A sample graph showing the depression detection probability in each tweet over a series of tweets is shown in Fig. 3. Such graphs can be plotted as the final output of the system for any user for a given series of tweets.

The graphs for the other emotions of fear, anger, and joy, which are also detected by the system, are also plotted as the output of the system. This helps in further assessing the emotions of the user and reaching a conclusion regarding the mental state and health of the user.

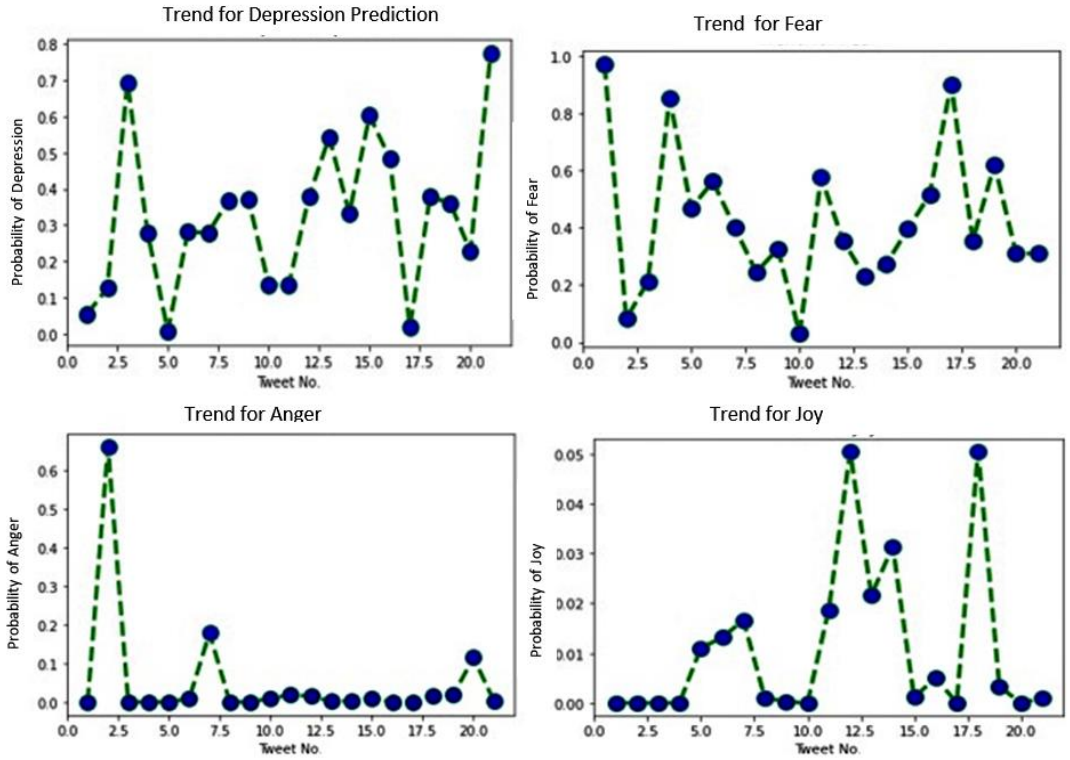

Figure 3. System output – plots of emotions (sadness, fear, anger, joy) over a sequence of tweets.

Fig. 3 shows the graph output of the Mental Health Analyzer for all four emotions detected by the system over a series of tweets.

IV. Experimentation and Performance

The tweet dataset was cleansed, the tweets were tokenized, and distributed word embeddings were used for vectorization and the CNN, LSTM, and Bidirectional LSTM. The models were trained and tested on this preprocessed dataset. The experimental results and performance of the three models are compared in the following sections.

A. Data Split Experimentation

The models were tested for different training-testing data splits. The results achieved for different data splits for the individual models are summarized below.

1) CNN model

Table VI shows the accuracies achieved for different training-testing data splits for the CNN model. The best training accuracy of 98% and validation accuracy of 91.07% were achieved for the 90-10 data split for the CNN model.

TABLE VI. CNN Model Accuracy Results

Dataset split	Accuracy	Validation accuracy
40 – 60 Train-Test	99.49	87.898
60 – 40 Train-Test	99.51	90.116
80 – 20 Train-Test	96.10	78.269
90 – 10 Train-Test	**98**	**91.077**

2) LSTM model

Table VII shows the accuracies achieved for different training-testing data splits for the LSTM model. The best training accuracy of 94.84% and validation accuracy of 93% are achieved for the 90-10 data split for the LSTM model.

TABLE VII. LSTM Model Accuracy Results

Dataset split	Accuracy	Validation accuracy
40 – 60 Train-Test	97.52	90.05
60 – 40 Train-Test	97.63	92.69
80 – 20 Train-Test	96.74	92.58
90 – 10 Train-Test	**94.84**	**93**

3) Bidirectional LSTM model

Table VIII shows the accuracies achieved for different training-testing data splits for the Bidirectional LSTM model. The best training accuracy of 97.10% and validation accuracy of 92.77% were achieved for the 90-10 data split for the Bidirectional LSTM model.

TABLE VIII. BIDIRECTIONAL LSTM MODEL ACCURACY RESULTS

Dataset split	Accuracy	Validation accuracy
40 – 60 Train-Test	93.67	87.49
60 – 40 Train-Test	98.10	91.39
80 – 20 Train-Test	97.92	91.61
90 – 10 Train-Test	**97.10**	**92.77**

B. Performance

The training and validation accuracy of different data splits was recorded, and the accuracy vs. epoch curve and loss vs. epoch curve for the best results for each model were plotted as described below.

1) CNN model

Fig. 4 shows the convergence curves of accuracy vs. epoch and model loss vs. epoch for the CNN model. As the best accuracy was obtained for the 90-10 training-testing data split, the convergence curves recorded for that split are shown.

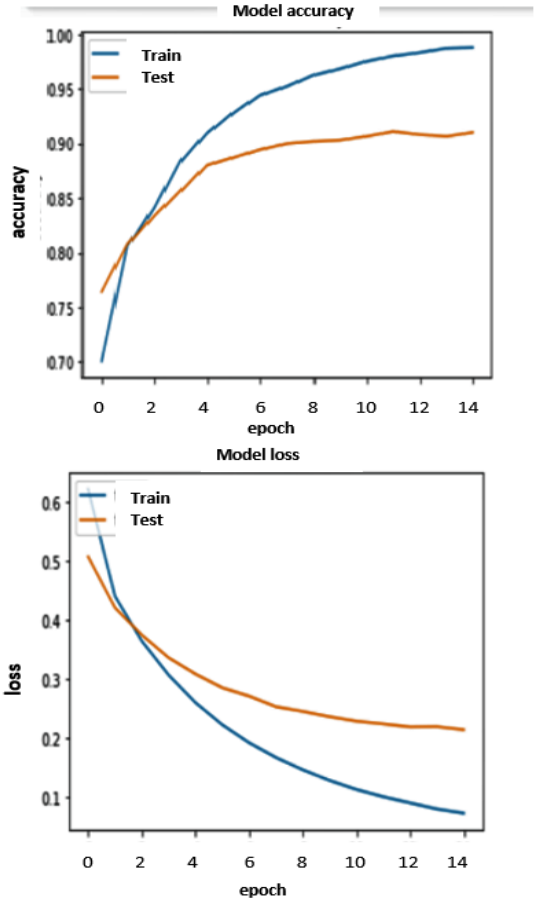

Figure 4. Model accuracy vs. epoch curve and model loss vs. epoch curve for the CNN model for 90-10 data split.

2) LSTM model

Fig. 5 shows the convergence curves of accuracy vs. epoch and model loss vs. epoch for the LSTM model. As the best accuracy was obtained for the 90-10 training-testing data split, the convergence curves recorded for that split are shown.

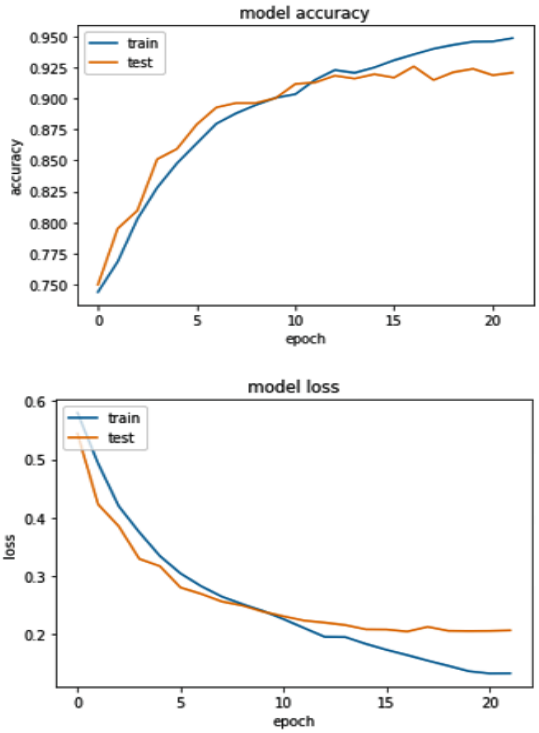

Figure 5. Model accuracy vs. epoch curve and model loss vs. epoch curve for the LSTM model for 90-10 data split.

3) Bidirectional LSTM model

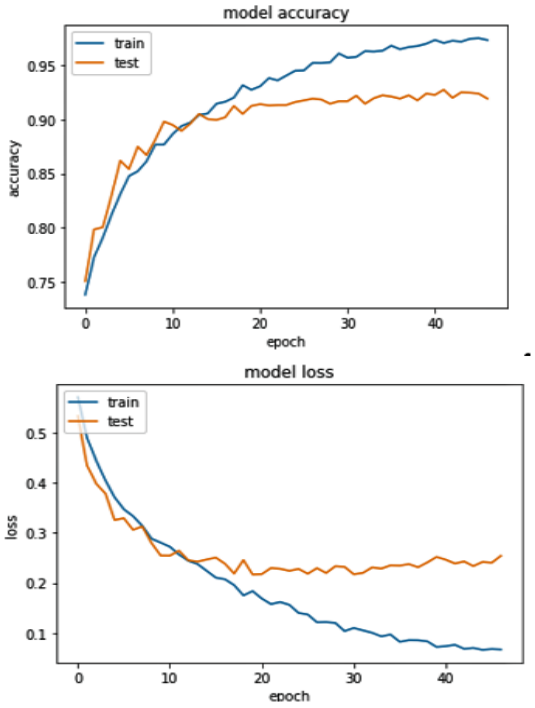

Figure 6. Model accuracy vs. epoch curve and model loss vs. epoch curve for the Bidirectional LSTM model for 90-10 data split.

Fig. 6 shows the convergence curves of accuracy vs. epoch and model loss vs. epoch for the Bidirectional LSTM model. As the best accuracy was obtained for the 90-10 training-testing data split, the convergence curves recorded for that split are shown.

V. Results

The tweets were classified into four emotions, namely, anger, fear, joy, and sadness. In the following discussion, the models are compared and evaluated using the training data including validation accuracy, precision, recall, and F1 scores.

A. Accuracy

The experiment for training and testing accuracy was carried out on all three models for different data splits for the task of multiclass classification, classifying the tweets into the four emotion classes of anger, fear, joy, and sadness. The best results for the three models were achieved for the 90-10 training-testing data split and are compared in Table IX.

TABLE IX. Accuracy Results for All Models

Model	Accuracy	Validation accuracy
CNN	98	91.077
LSTM	94.84	93
Bidirectional LSTM	97.10	92.77

All three models performed better than the models proposed in the SemEval-2018 emotion classification task [1]. The LSTM model obtained the highest accuracy of 93%, closely followed by the Bidirectional LSTM and CNN models.

B. Precision and Recall

Hossin *et al.* [33] present a survey of the evaluation metrics used for evaluating multiclass data classification tasks. Accuracy serves as an overall evaluation metric by evaluating the performance of the model in correctly classifying the tweets for all the classes. Precision is defined as the ratio of true positives to the sum of true and false positives, while recall is defined as the ratio of true positives to the sum of true and false negatives. The F1 score is the harmonic mean of precision and recall and thus gives equal weighting to precision and recall. To understand the performance of the three individual models for each individual emotion class, precision, recall, and F1 scores are used as the evaluation metrics.

Tables X, XI, and XII show the precision, recall, and F1 scores for the different emotion classes achieved by the CNN, LSTM, and Bidirectional LSTM models, respectively.

In Tables X-XII, Support is the number of comments which were present in the testing set for each of the respective emotion. The support has been calculated automatically by a function wile calculating metrics.

TABLE X. Precision, Recall, F1 Score for CNN Model

Class	Precision	Recall	F1 score	Support
Anger	85.9	89	82.3	100
Fear	81.4	86	83.6	107
Joy	88.8	100	94	79
Sadness	81.7	72.1	76.6	68

TABLE XI. Precision, Recall, F1 Score for LSTM Model

Class	Precision	Recall	F1 score	Support
Anger	83.7	82	82.8	100
Fear	80.4	84.1	82.2	107
Joy	87.6	98.7	92.9	79
Sadness	87.3	70.6	78	68

TABLE XII. Precision, Recall, F1 Score for Bidirectional LSTM Model

Class	Precision	Recall	F1 score	Support
Anger	84.4	81	82.7	100
Fear	81.8	84.1	82.9	107
Joy	91.8	98.7	95.1	79
Sadness	79.4	73.5	76.3	68

C. Comparison of Models for the Sadness Emotion Class

The emotion class of sadness is the most important class for the objective of depression detection. Table XIII compares the three models on the metrics of precision, recall, and F1 score on a scale of 100 (100 being the maximum) for the sadness emotion class in the task of emotion classification.

TABLE XIII. Comparison of Models for the Sadness Class

Model	Precision	Recall	F1 score
CNN	81.7	72.1	76.6
LSTM	87.3	70.6	78.0
Bidirectional LSTM	79.4	73.5	76.3

The highest precision of 87.3% was achieved by the LSTM model and the highest recall of 73.5% was achieved using the Bidirectional LSTM model. The LSTM model achieved the highest F1 score among the three models for the sadness emotion class.

To increase the chances of depression detection, the recall score needs to be increased. Hence, an ensemble maximizing model is proposed that uses the maximum probability for classification of each emotion obtained from the CNN, LSTM, and Bidirectional LSTM models, respectively.

D. Depression Detection Graph Results

The system classifies each tweet into one or more of the four classes of sadness, fear, anger, and joy. The sadness emotion class is used to detect depression. The depression detection results are obtained by plotting a depression vs. tweets graph for a series of tweets. The graph helps to visualize and analyze whether the user is in a depressive state or moving toward that state by considering the presence of sadness in the given series of tweets.

A total of 21 tweets from a random user that were classified in the sadness emotion class were collected for validation. As mentioned in Section III F, the user is selected randomly and is does not belong to either the training or test state. The aim of plotting these graphs based on these tweets is to showcase the application of the research work. As all the tweets belong to the sadness class, the model for which the graph shows the maximum number of tweets as depressed performs better than the rest.

1) CNN model

Fig. 7 shows the depression vs. tweets graph when the CNN model was tested for validation on the series of depressed tweets. The non-zero values show the tweets that the model has correctly classified as indicating depression, and the values represent their respective classification probabilities.

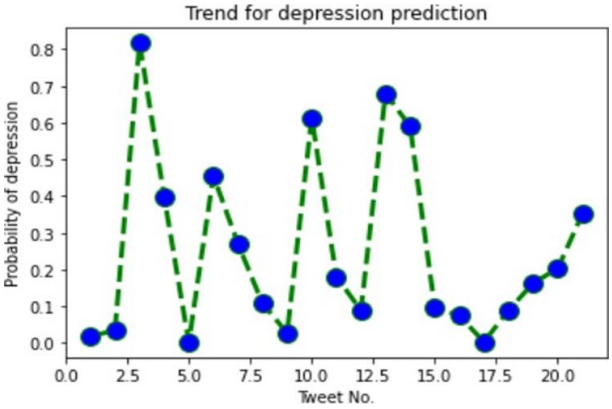

Figure 7. Depression detected by CNN model on a set of depressed tweets by a single user.

2) LSTM model

Fig. 8 shows the depression vs. tweets graph when the LSTM model is tested for validation on the series of depressed tweets. The non-zero values show the tweets that the model has correctly classified as depressed tweets and represent their respective probabilities.

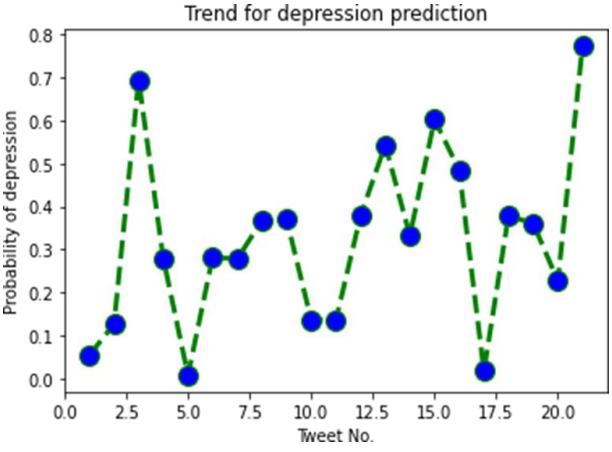

Figure 8. Depression detected by LSTM model in a set of depressed tweets by a user.

3) Bidirectional LSTM model

Fig. 9 shows the depression vs. tweets graph when the Bidirectional LSTM model is tested for validation on the series of depressed tweets. The non-zero values show the tweets that the model has correctly classified as depressed, along with their respective probabilities.

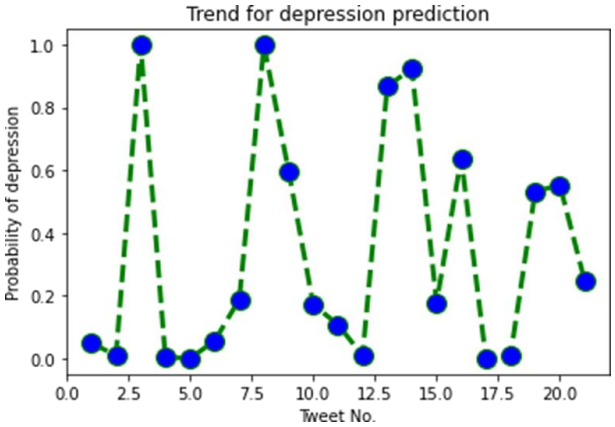

Figure 9. Depression detected by Bidirectional LSTM model in a set of depressed tweets by a user.

4) Maximizing ensemble model

Fig. 10 shows the depression vs. tweets graph when the maximizing ensemble model is tested for validation on the series of depressed tweets. The non-zero values show the tweets that the model has correctly classified as depressing tweets and their respective probabilities.

The collected series of tweets consisted of all tweets belonging to the sadness emotion class. It can be seen from the graphs that the CNN, LSTM, and Bidirectional LSTM models were not able to individually detect all the tweets as belonging to the sadness class. One model may wrongly classify a tweet as not belonging to the sadness class, while another might correctly classify.

The maximizing ensemble model, on the other hand, can correctly detect the presence of depression by correctly classifying almost all the tweets into the sadness class. This shows that the maximizing ensemble model, using a combination of all three models, CNN, LSTM, and Bidirectional LSTM, maximizes the chances of depression detection.

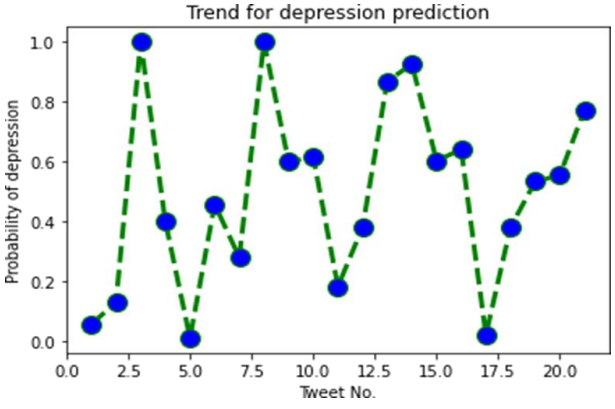

Figure 10. Depression detected by maximizing ensemble model in a set of depressed tweets by a user.

Many of the tweets used for this test require contextual understanding to be correctly classified as belonging to the sadness class. Sample tweets and their depression probabilities as predicted by the maximizing ensemble

model are given in Table XIV to demonstrate the effectiveness of the proposed model.

TABLE XIV. SAMPLE VALIDATION RESULTS FOR DEPRESSION DETECTION AND CLASSIFICATION BY THE PROPOSED ENSEMBLE MAXIMIZING MODEL

Tweet	Emotion
I try so hard to ignore and fill the void with anything I can think of, and for a little while it works, but at the end of the day, it all comes back.	Depression probability = 99.79087%
Do you ever just sit there and realize that you mean nothing to anyone and you start feeling lost, alone, unloved, and truly unwanted.	Depression probability = 86.67954%
Every night I think about all the mistakes I have made and how much I regret making those decisions because my life could have been different.	Depression probability = 63.87678% Fear probability = 73.694164%
She might be laughing but deep inside she's hurting, she's trying to get her mind off things that make her upset by pretending to be fine.	Depression probability = 92.4308%

VI. CONCLUSION

The implemented Mental Health Analyzer uses CNN, LSTM, and Bidirectional LSTM models for emotion classification and depression detection. The task of depression detection requires the models to understand the context and find latent features, which makes deep learning models suitable for the task.

The maximum validation accuracy of 93%, F1 score of 78.0, and precision of 87.3 are achieved by the LSTM model, while the maximum recall of 73.5 is achieved by the Bidirectional LSTM model. There are tweets for which one of the models is not able to identify depression, while another model is able to correctly classify the tweet into the sadness emotion class and thus detect depression. As recall is inversely dependent on the false negatives, recall needs to be maximized so that no depressed tweet goes undetected.

To tackle this, an ensemble model has been implemented, which considers the maximum probability of each emotion among the predicted probabilities derived by each individual model. This improves the chances of depression detection by increasing the recall for all the emotion classes including sadness and fear. This can be seen from the results wherein the ensemble model resulted in higher probabilities of depression compared to the individual models and detected most of the depressed tweets in the validation dataset of sequential tweets.

The validation results of the ensemble model on a set of tweets collected from a real user show that the model is effective on real-time data. The graph of depression prediction vs. tweets will prove helpful to analyze and assess whether a user is in a state of depression or is moving toward a state of depression. Other emotion labels for the tweets, viz., fear, anger, and joy, along with the graphs of these emotions plotted vs. the series of tweets, will help in better assessment of the mental health of the user.

VII. APPLICATIONS AND SCOPE

- This model is helpful in analyzing the mental health of a user by finding the latent meaning of the user's tweets and understanding the tweets in context. The ensemble model can be used to detect depression in people to provide them with the required help. This can be helpful in monitoring and automating a need-based help response for millions of users, which cannot be done manually.
- The system also detects fear, anger, and joy in the tweets, which can be used to analyze the sentiments of people on various matters. The opinions of people on public matters can be made known, which can help authorities to work for the public welfare.
- It is important to filter out hate comments and violent threats on microblogging sites, which can be done by analyzing the emotions present in the tweets. Hate can often be expressed in the form of anger, which can be detected by the model.

CONFLICT OF INTEREST

The authors declare there is no conflict of Interest.

AUTHOR CONTRIBUTIONS

Pranav Bhat and Alwin Anuse worked on Ideation of the work; Pranav Bhat, Alwin Anuse, Rupali Kute, R.S. Bhadade and Prasad Purnaye worked on the literature survey; Pranav Bhat and Rupali Kute worked on Dataset creation; Pranav Bhat, Alwin Anuse and Rupali Kute worked on Data analysis and interpretation; Pranav Bhat and Alwin Anuse worked on the Research methodology, Experimentation, Results and Conclusion; Pranav Bhat and Alwin Anuse worked on drafting the article; Pranav Bhat, Alwin Anuse and Rupali Kute worked on the Critical revision of the article; All authors had approved the final revision.

REFERENCES

[1] Depression. [Online]. Available: https://www.who.int/health-topics/depression#tab=tab_1

[2] S. Mouchet-Mages and F. J. Baylé, "Sadness as an integral part of depression," *Dialogues Clin. Neurosci.*, vol. 10, no. 3, pp. 321-327, 2008.

[3] D. Forbes, *et al.*, "A longitudinal analysis of posttraumatic stress disorder symptoms and their relationship with fear and anxious-misery disorders: Implications for DSM-V," *J. Affect. Disord.*, vol. 127, pp. 147-152, 2010.

[4] A. J. Rodríguez-Hidalgo, Y. Pantaleón, I. Dios, and D. Falla, "Fear of COVID-19, stress, and anxiety in university undergraduate students: A predictive model for depression," *Frontiers in Psychology*, 2020.

[5] V. A. Kharde and S. S. Sonawane, "Sentiment analysis of Twitter data: A survey of techniques," *International Journal of Computers and Applications*, April 2016.

[6] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. 42nd Meeting of the Association for Computational Linguistics*, 2004, pp. 271-278.

[7] S. Liu, F. Li, F. Li, X. Cheng, and H. Shen, "Adaptive co-training SVM for sentiment classification on tweets," in *Proc. the 22nd ACM International Conference on Conference on Information & Knowledge Management*, 2013, pp. 2079-2088.

[8] A. Hasan, S. Moin, A. Karim, and S. Shamshirband, "Machine learning-based sentiment analysis for twitter accounts," *Mathematical and Computational Applications*, vol. 23, no. 1, 2018.
[9] R. Socher, *et al.*, "Recursive deep models for semantic compositionality over a sentiment Treebank," in *Proc. the Conference on Empirical Methods in Natural Language Processing*, 2013.
[10] A. Agarwal, F. Biadsy, and K. R. Mckeown, "Contextual phrase-level polarity analysis using lexical affect scoring and syntactic n-grams," in *Proc. the 12th Conference of the European Chapter of the ACL*, 2009.
[11] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of Twitter data," in *Proc. the Workshop on Language in Social Media*, 2011.
[12] S. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, "Semeval-2018 task 1: Affect in Tweets," in *Proc. International Workshop on Semantic Evaluation*, June 2018.
[13] A. Abbasi, H. Chen, and A. Salem, "Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums," *ACM Transactions on Information Systems*, vol. 26, no. 3, pp. 1-34, 2008.
[14] Y. Lu, *et al.*, "Automatic construction of a context-aware sentiment lexicon: An optimization approach," in *Proc. the 20th international Conference on World Wide Web*, 2011, pp. 347-356.
[15] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proc. the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 2005, pp. 347-354.
[16] B. Yoshua, *et al.*, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137-1115, 2003.
[17] M. Tomas, *et al.*, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, 2013.
[18] M. Tomas, *et al.*, "Efficient estimation of word representations in vector space," arXiv Preprint ArXiv:1301.3781, 2013.
[19] D. Tang, *et al.*, "Learning sentiment-specific word embedding for twitter sentiment classification," in P*roc. the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014, pp. 1555-1565.
[20] G. Xavier, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proc. the 28th International Conference on Machine Learning*, 2011.
[21] J. Wehrmann, *et al.*, "A character-based convolutional neural network for language-agnostic Twitter sentiment analysis," in *Proc. International Joint Conference on Neural Networks*, 2017, pp. 2384-2391.
[22] S. Poria, H. Peng, A. Hussain, N. Howard, and E. Cambria, "Ensemble application of convolutional neural networks and multiple kernel learning for multimodal sentiment analysis," *Neurocomputing*, vol. 261, pp. 217-230, 2017.
[23] M. Volpi and D. Tuia, "Dense semantic labeling of subdecimeter resolution images with convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 881-893, 2017.
[24] Z. Tian, M. Li, M. Qiu, Y. Sun, and S. Su, "Block-DEF: A secure digital evidence framework using blockchain," Information Sciences, vol. 491, pp. 151-165, 2019.
[25] Z. Tian, *et al.*, "Real time lateral movement detection based on evidence reasoning network for edge computing environment," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4285-4294, 2019.
[26] D. Xu, Z. Tian, R. Lai, X. Kong, Z. Tan, and W. Shi, "Deep learning based emotion analysis of microblog texts," *Information Fusion*, vol. 64, pp. 1-11, 2020.
[27] Y. Kim, "Convolutional neural networks for sentence classification," ArXiv Preprint ArXiv:1408.5882, 2014.
[28] Y. Ren, *et al.*, "Context-sensitive twitter sentiment classification using neural network," in *Proc. Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
[29] M. Arora and V. Kansal, "Character level embedding with deep convolutional neural network for text normalization of unstructured data for Twitter sentiment analysis," *Social Network Analysis and Mining*, vol. 9, no. 1, p. 12, 2019.
[30] R. Satapathy, L. Yang, S. Cavallari, and E. Cambria, "Seq2Seq deep learning models for microtext normalization," in *Proc. International Joint Conference on Neural Networks*, 2019, pp. 1-8.
[31] X. Wang, W. Jiang, and Z. Luo, "Combination of convolutional and recurrent neural network for sentiment analysis of short texts," in *Proc. COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 2428-2437.
[32] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. the Conference on Empirical Methods in Natural Language Processing*, 2014.
[33] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining and Knowledge Management Process*, vol. 5, no. 2, pp. 1-11, 2015.

Pranav Bhat was born in Nashik, Maharashtra, India on 9th September 1998. He received his B.E. degree in Electronics and Telecommunication from Savitribai Phule Pune University, India in 2020. He is a Business Technology Analyst at Deloitte Consulting USI. His current research interests include Deep learning, NLP, Computer Vision and Artificial Intelligence.

Dr. Alwin Anuse was born in Pune, Maharashtra, India on 2nd May 1980. He earned his Ph.D. in Electronics and Telecommunication from Savitribai Phule Pune University in 2017. He is an associate professor at Vishwanath Karad MIT - World Peace University. His current research interests include AI, Neural Networks and Signal Processing.

Dr. Rupali Kute was born in Nashik, Maharashtra, India on 20th August 1985. She earned her Ph.D. in Electronics and Telecommunication from Savitribai Phule Pune University in 2019. He is an assistant professor at Vishwanath Karad MIT - World Peace University. Her current research interests include AI, Neural Networks and Biometrics.

R. S. Bhadade was born in Karad, Maharashtra, India on 6th June 1976. He earned his Ph.D. in Electronics and Telecommunication from Savitribai Phule Pune University in 2000. He is an associate professor at Vishwanath Karad MIT - World Peace University. His current research interests include AI and Massive MIMO Antenna.

Prasad Purnaye was born in Washim, Maharashtra, India on 17th December 1991. He earned his M.Tech. in Computer Science from VJTI, Mumbai in 2017. He is an assistant professor at Vishwanath Karad MIT- World Peace University. His current research interests include AI and Cloud Forensics using ML.