2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS)

Detecting Depression in Reddit Posts using Hybrid Deep Learning Model LSTM-CNN

Bhumika Gupta
G.B. Pant Institute of Engineering & Technology
Pauri Garhwal, India
mail2bhumikagupta@gmail.com

Neeraj Pokhriyal
G.B. Pant Institute of Engineering & Technology
Pauri Garhwal, India
npokhriyal12@gmail.com

Kamal Kumar Gola
College of Engineering Roorkee (COER)
Roorkee, India
kkgolaa1503@gmail.com

Mridula
College of Engineering Roorkee (COER)
Roorkee, India
dr.mridula121@gmail.com

***Abstract*— The detection of depression is a critical issue for human well-being. Previous research has shown us that online detection is successful in social media, allowing for proactive intervention for depressed users. It is a serious psychological disorder and it takes hold of more than 300 million people across the globe. A person who is depressed experience anxiety and low self-esteem in their everyday life, which affects their relationships with their family and friends, and can lead to various diseases and, in the most extreme scenario, suicide. With the rise of social media, the majority of individuals now use it to express their emotions, feelings, and thoughts. If a person's depression can be discovered early by analyzing their post, then essential efforts can be taken to save them from depression-related disorders or, in the best scenario, from suicide. The main goal of our work is to inspect Reddit user posts to see whether any factors suggest depression attitudes among relevant internet users. We use sentiment examination and Machine Learning (ML) techniques to train the ML model and assess the efficacy of our suggested strategy for this goal. A lexicon of phrases that are more common in depressed accounts is identified. In this study, we have combined Long Short-Term Memory (LSTM) and Convolution Neural Network (CNN) to build a hybrid model that can predict depression by evaluating user textual messages.**

Keywords*— *NLP, LSTM, CNN, Deep Learning, Sentiment Analysis, LSTM-CNN.

I. INTRODUCTION

Depression is among the most prevalent psychological conditions in the world nowadays. Depression is a mental condition that influences how we think, experience, and behave. In recent years, the frequency of suicides attributed to depression has been increasing. This is a problem that must be talked about. Given the quick rise of numerous social media stages and their impact on the social order and a person's psychosomatic background, a micro-blogging website is a platform for depressed people to express emotions, as well as to study their behavior by tracing their social movements through their posts [1]. The need for AI (Artificial Intelligent) technologies that can effectively handle Early Risk Detection (ERD) concerns on social networks, such as early diagnosis of depression, early misinformation detection, and early sexual molester detection has increased with the growth of the Internet. Early identification and counseling for depressed individuals can significantly increase the likelihood of managing depression and minimizing its negative impact on a user's well-being, healthcare, and socioeconomic life [2]. As per the WHO analysis, 322 million individuals worldwide or 4.4% of the population are expected to have depression. The Western Pacific area, which includes China, India, and South-East Asia, accounts for 27% and 27% of the population at risk. Depression is still underdiagnosed and inadequately treated in many nations, which can result in negative self-perception and, in the worst cases, suicide [3].

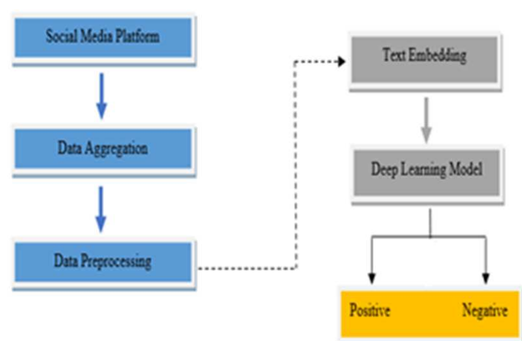

Figure 1. Steps for the Sentiment Analysis

In the domain of natural NLP, text classification has long been a root and well-liked study domain. Since text categorization technology has a diverse range of uses in our everyday lives, including the classification of text, subjective and objective sentences, and movie reviews, it is highly significant in assisting people in making decisions in real life. The two main categories of conventional text categorization algorithms are those that rely on dictionaries and those that use ML. The development of Deep Learning (DL) algorithms has considerably improved text categorization accuracy. When classifying texts using a deep learning approach, CNN and LSTM are widely used. An enhanced variation of the error back propagation network is the multi-layer convolutional neural network. It excels at using photos, especially large ones, to solve connected machine learning problems. The Recurrent Neural Network (RNN) is a neural network structure that features loops. Information can be preserved via it. The production of the concealed layer is dependent on historical data at all times. It has been widely employed in natural language processing applications including text categorization and machine translation. Chain characteristics of RNN show that the model is strongly connected to the problem of sequence annotation. The gradient explosion or disappearance brought on by the long-term reliance and the lengthy sequence of the RNN is a concern, though, because the present output results of the RNN are tied to a long input sequence. An LSTM neural network was developed by researchers to avoid the long-term reliance on RNN [4].

II. Literature Review

Sentiment analysis has long been one of the most important explorations demands since it works with big client assessments in a variety of applications. Within the scope of this review, it was discovered that hybrid algorithms produced superior classification exhibitions. In the wake of these experiments, it was discovered that the hybrid algorithm produced improved classification results. AI and DL were combined by Liu et al. [5] to execute sentiment classification better. According to recent research, when dealing with text-based data, scientists are far more likely to use CNN and LSTM. It is worth noting that sentiment analysis predicts whether a sentiment will be neutral, positive, or both. Furthermore, it may be used to guess a variety of emotions such as love, disdain, grief, astonishment, wrath, despair, and melancholy. Several studies [6,7] evaluate these feelings and demonstrate how much they were transferred by projecting their power. Investigations of medical services information are conducted regularly to examine the therapeutic experiences that patients and their friends and family have discussed. The author compiled 845 tweets from bosom disease sufferers on their own experiences with therapy and perseverance [8].

Conventional machine learning algorithms, according to the study [9], are vague when it comes to training from complex and unstructured datasets. Another advantage of DL is how it adjusts Transfer Learning. Pretrained models on massive amounts of data may be used for a variety of related activities. Using pre-trained models can significantly reduce the quantity of data and processing resources necessary for model training. CNN for text was initially proposed as a solution for text classification in 2014. CNN is extremely successful in object and picture classification applications. CNN is used for text classification problems to represent documents as pictures that the Neural Network can comprehend. Researchers used DL-based techniques to extract more information from images, videos, unstructured text, and emojis. Orabi et al. [10] used CNN and RNN to identify sadness in Twitter dataset. A novel method for detecting depressed users was proposed by Zogan et al. [11] and is dependent on the real user timeline Twitter posts as well as user behavior on a benchmark dataset, the CNN and Bi-GRU hybrid model was evaluated.

Another study [14] examined how people from different cultural backgrounds felt about the ensuing actions taken by various countries and how they responded to the Coronavirus. LSTM model has successfully built using the sentiment140 data to achieve the highest level of accuracy when estimating emotion polarities and emotions from retrieved tweets. An unsupervised progression-based word embedding strategy was proposed by Jianqiang et al. [12]. Storing context-oriented semantic linkages and depiction among words and tweets are used in this suggested technique. The implanted phrases are combined with n-gram and state of mind extremity score features to construct a set of close-to-home highlights that are then incorporated into a profound convolutional brain function. A brand-new multi-step time series forecasting model was proposed by Han et al. [13]. This study attempted to forecast COVID-19 tiredness, anxiety, and severity. The model was tested on replicated and real-life datasets. The measurable and comprehensive assessments were carried out built on expected accuracy and computational cost. An RNN-based DL algorithm was suggested by Jelodar et al. [16] for the sentiment classification of COVID-19-related microblogging texts. They created a model, particularly for mining COVID-19-associated problems. Only a small number of important alterations exist between this work and ours. They began by analyzing Reddit content, the same as our study. Second, they utilized deep learning with LSTM, whereas we used a hybrid DL model (LSTM-CNN) for the categorization of depressed Reddit postings. Lastly, they focused on opinion extraction whereas we focused on how strongly Reddit members feel about sad feelings.

The authors [15] then proposed a new sentiment analysis model called SLCABG that combines the features of CNN and bi-GRU (Bidirectional Gated Recurrent Unit) entities. It is based on a sentiment lexicon and utilizes ML and lexicon to improve product review performance.

A novel Persian sentiment analysis technique using context awareness and deep learning was developed by Dashtipour et al. [17]. Persian movie reviews are categorized as good or bad using the recommended deep-learning-driven based feature extraction approach. In addition to their earlier projected, Support Vector Machine (SVM) based methodology, they applied and contrasted two deep learning techniques, CNN and LSTM. LSTM beat the Multilayer Perceptron (MLP), auto encoder, SVM, Logistic Regression, and CNN algorithms, according to the experimental findings. Furthermore, Rehman et al. [18] developed a Hybrid CNN-LSTM Model, which combines LSTM and highly deep CNN models, to address the sentiment analysis problem. To begin, they trained basic word embeddings using the Word to Vector (Word2Vec) technique. Word2Vec computes the distance between words, groups words with similar meanings, and turns text into a matrix of numeric values. The proposed model contains a set of features obtained using convolutional layers and global max-pooling layers, as well as longstanding needs. To boost accuracy, the suggested framework additionally employs dropout technology, normalization, and modified linear units. In regards to accuracy, recall, F-value, and precision, their findings reveal that the proposed hybrid CNN-LSTM model surpasses classic DL and ML approaches.

III. Background

The core concepts of the approaches employed in sentiment analysis and text classification challenges are briefly discussed in this section.

A. *Deep Learning Approaches*

Deep learning is generally three or more layers of a neural network stacked above or beneath other layers. Such neural networks attempt, though imperfectly, to mimic the functioning of the human mind by learning as humans from vast sums of data. A single layer network could give accurate estimates, and the addition of further layers can aid with accuracy refinement. The different DL methods used in this study are discussed below.

1) Convolution Neural Network (CNN): CNNs are made up of three different types of layers. The convolution layer, pool, and fully-connected layers are linked. These layers are layered to create a CNN architecture. A straightforward structure for text categorization with CNN is revealed in Figure 2.

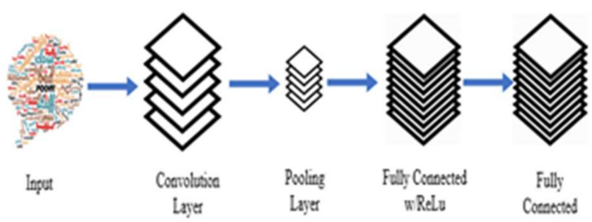

Figure 2. CNN Architecture

The fundamental functionality of the sample CNN above may be divided into four distinct sections.

- The initial layer in CNN architecture is the input layer. This layer receives the data and forwards it to the remainder of the network.
- The convolution layer is also known as the feature extractor layer since the features of the text are separated inside it. A piece of image is linked with the Convolution layer to do convolution activity as previously seen and figure out the dab item between the open field and the filter. The activity's aftereffect is a single whole number of the outcome volume. Then, using a Stride, we glide the filter over the subsequent responsive field of comparable information and repeat a similar action.
- The pooling-layer is used to decrease the spatial quantity of the user input matrix after convolution. It is used to connect two convolution layers. As a result, max pooling is the sole technique to minimize the spatial volume of the input.
- To produce the final output from the activation functions for classification, the hidden layers will then carry out identical operations as standard ANNs. It is also suggested that ReLu be used in between these layers to improve performance.

2) Long Short-Term Memory (LSTM): The LSTM is an enhanced RNN, commonly referred to as a sequence network that enables data storage. It can fix the RNN's disappearing gradient issue. In contrast to traditional feed forward neural networks, LSTMs have feedback connections. Because of this characteristic, LSTMs can process whole data sequences (such as time series) without having to deal with each data point separately. Instead, they may retain crucial information about previous data points in the sequence to help them interpret subsequent data points. LSTMs, perform exceptionally well for processing data sequences like textual, audio, and basic time series. Data entering an LSTM cell goes through three different gates for processing. Output gate, input gate, and a forget gate are the three different gates that make up an LSTM cell. These gates may be regarded as filters because each one is a distinct neural network. Throughout this section, we will go over each one in more detail.

- The forget gate is the initial stage of the procedure. This stage involves identifying the parts important in light of both the previous hidden state and the brand-new incoming input data. The forget gate determines what information must be remembered and what can be forgotten. The sigmoid function collects the data from X(t) and h(t-1). The values that Sigmoid produces range from 0 to 1. Closer the value to 1 then it will retain the information and if closer the value to 0 it will discard the information.
- The input gate does the subsequent processes to recondition the cell states. The previously concealed state h(t-1) and the present state X(t) are sent as input to the next sigmoid function. The range of transformed values is 0 to 1. The input from the concealed state and present state work as input for the tan-h function. To manage the network, the tan-h operative will create a vector containing each feasible value amid -1 and 1. After that output from both the sigmoid and tan-h functions will be ready for multiplication.

- The LSTM cell has sufficient data from the forget gate and input gate. The subsequent stage is to choose and store the data from the new state in the cell state. The data from C (t-1) gets multiplied with forget gate's output. If the output from this previous cell state is 0 then the cell state will drop this value. Then, the LSTM cell takes the product from the input gate and performs addition with cell state data updating the cell state C(t).

- The output gate decides which information needs to be in a concealed state. This state has data on past inputs. To begin with, the data from the current state and past concealed state are passed into another sigmoid function. After this new cell state will go through the tan-h. Output from these two functions is multiplied. According to the final output, the LSTM cell decides which data to be kept.

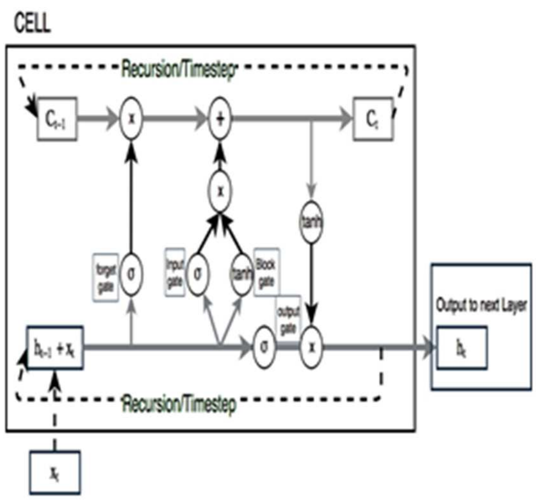

Figure 3. An LSTM cell diagram

To finish this segment, the forget gate picks whatever vital information from the previous procedures is necessary to finish. The output gates create the next h(t), whereas the input gate selects what critical information from the current stage may be delivered.

3) Word Embedding: Word embeddings meaningfully convert human language into numerical form. This enables computers to comprehend the subtleties implicitly hidden in our languages. The basic notion here is that each word may be transformed into a set of integers and an N-dimensional vector. Even though each word is allocated a unique vector/embedding, comparable words end up with values that are closer together. The vectors for the phrases 'Woman' and 'Girl', for example, would be more similar than the vectors for 'Girl' and 'Cat.' To be really effective, these numerical representations must incorporate meanings, semantic links, patterns among words, and the context of distinct words as they are used organically by humans.

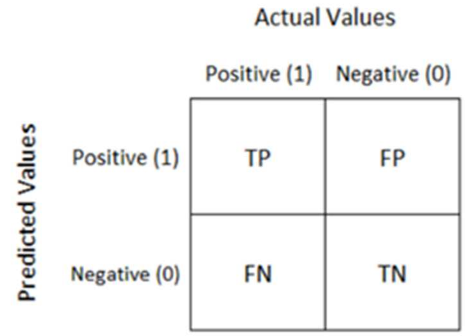

Figure 4. Confusion Matrix

Pretrained Word Embeddings are embeddings acquired in one activity and applied to another similar task. These embeddings are learned On big datasets, saved, and then applied to different problems. Because of this, pre-trained word embeddings are a type of Transfer Learning.

B. Performance Evaluation

The values from the confusion matrix will be evaluated with the categorization achievements from text categorization in related research to demonstrate the accuracy of the technique. The confusion matrix is used to determine measurement values for accuracy, precision, and F1. Table 1 provides the basic confusion matrix for binary classification.

- TP: TP is how many times both the predicted and actual classifications are positive.

- FP: FP is how many occurrences in which the anticipated classification is positive while the predicted classification is negative.

- FN: FN is how many occurrences in which the anticipated classification is negative while the predicted classification is positive.

- TN: TN is how many times both predicted and actual classifications are negative.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Precision = \frac{TP}{TN + TP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

IV. Proposed Methodology

In this part, we will discuss in depth the dataset used, text preprocessing, and the architecture of the proposed model.

A. Dataset

Reddit is a well-known microblogging network that provides easy access to information. It takes a long time to define and confirm terms used as a vocabulary for persons with mental diseases to access data, this is where Reddit comes into the picture. Reddit is community-based social media where users post about their experiences in a particular community. The dataset consisting of 231,943 unlabeled posts was obtained for this investigation using the Reddit API PRAW. This is quite simple to utilize and may provide us with useful information about the user such as postings, date, URLs, topic, username, and so on. To label this dataset, we utilized the NLTK tool VADER Sentiment. It aided in LSTM + CNN distinguishing the attitude expressed in the posts as positive and negative. The dataset statistics are shown in Table 1.

TABLE I. DEMOGRAPHY OF DATASET

Total number of Posts	Depressed Posts	Non-depressed Posts
231,943	116,014	115,929

B. Text Pre-processing

Data pre-processing has always been a requisite component of data mining procedures. Real-world data is obtained using several ways that are not domain-specific, resulting in inadequate, unorganized, and uncertain data with errors. When such data is analyzed directly, it produces inappropriate and incorrect predictions. Various strategies are used in the current framework throughout the pre-processing period which is shown in Figure 5. The primary approach minimizes user-specified text arrangements. This technique discards all URLs in the Reddit post and thoroughly cleans each post in the dataset. URLs are not considered because they are irrelevant for prediction, and removing them reduces computation difficulty. Date and time are also not important for the classification of depressed individuals from non-depressive individuals, thus they have been removed from the posts. Similarly, numerals are not a viable trait intended for estimation, and hashtags can be utilized for prediction. It had been found that prediction based on hashtags has very low accuracy. Because we don't need to depart from the concept, hashtags were eliminated as well. The succeeding step remains to eliminate emojis as well as excessive spaces from the statement. Stopwords such as are was, at, if, and so on add nothing to the sense of the statement. The Re package, which consists of a group of stopwords, is utilized to eliminate stopwords from the transcript. Following that, stop words are removed, and lemmatizing is conducted. Lemmatization is the process of bringing all of a term's grammatical forms together so that they may be analyzed as a single item. Lemmatization is similar to stemming in that it provides meaning to words. As a result, it combines words with similar origins into a single word. Text preprocessing incorporates both stemming and lemmatization. Many individuals are perplexed by these two words. Some people confuse the two. Lemmatization is recommended over stemming since it does linguistic analysis of the words. Following the cleaning of all posts, the cleaned posts are retrieved and fed into the tokenizer. Tokenizing raw text input is a critical stage in the pre-processing of NLP algorithms. Tokenizers break the sentences in tokens. Token helps us in building the vocabulary and to process the data easily.

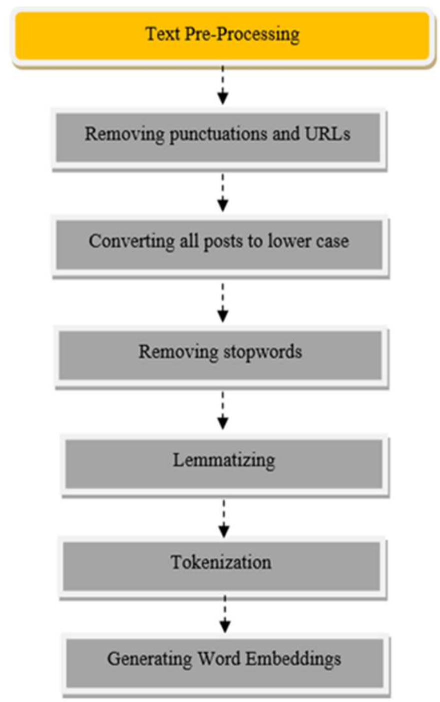

Figure 5. Flowchart of Text preprocessing

The Re package is imported to use the various tokenization functions. The text data after the cleaning of positive and negative posts are fed to Tokenizer.fit_on_texts () in the first stage of tokenizing. It produces the vocabulary index based on word frequency and refreshes the interior vocabulary using a record of texts. As a result, the term with the lowest index value is the one with the highest frequency. As a result, this function provides the greatest possible amounts of words, each with its index. The next phase is the texts_to_sequences () method. It gets information from the previous functions, this consists of the maximum number of words with an index. Its objective is to transform each word in a post into an integer series and replace it with the the corresponding numerical value from the word index dictionary. After this transformation posts which are smaller than 500 words are padded with zeroes. Then word embeddings are created with the help of pre-trained word embedding GloVe.

C. Proposed Model

We proposed combining CNN and LSTM for improved classification performance in predicting depressed Reddit users. LSTM is an enhanced RNN, also known as a sequential network that authorized information to be stored. It helps in solving the disappearing gradient problem of RNN. Because of the diminishing gradient, RNNs is not in remember information in long sentences. These are deliberately avoided with LSTMs and we have discovered that CNN is efficient at removing dimensional characteristics as well as performing fine once background statistics from the former arrangement are not necessary after running numerous experiments. RNNs, on the other hand, are useful for retrieving data when the context of nearby inputs is critical for categorization. Initially, multi-dimensional data is utilized as a low-level insert to CNN. Every layer mines critical features throughout the pooling and convolution phase. In contrast to classic CNN, the output layer is completely linked to the hidden layer. A variety of convolutional layers, pooling layers, and hidden layers are used to extract depression posts in depth and improve accuracy. An intricate network may form due to over fitting risk exposure.

The initial layer in this hybrid model is the embedding layer which contains word embedding for our text data. After that embedding is fed into the LSTM layer as an input, and the output of this LSTM layer is used as an input by the convolution layer. To reduce the feature extraction matrix given by the convolution layer a max-pooling1d layer is used. To flatten the output flattening layer is used after the max-pooling layer. The flattening layer is connected to a fully connected layer and this fully connected layer is linked to the output layer.

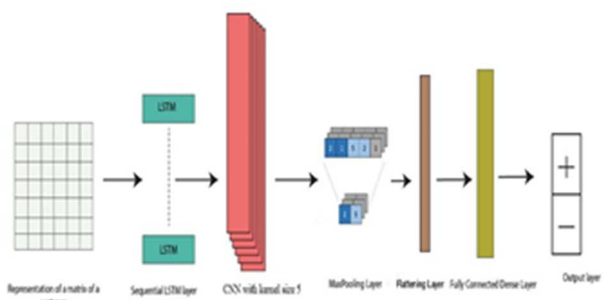

Figure 6. LSTM-CNN component of the network

Before the data was fed into the model, the text underwent preprocessing. This entails eliminating blank space and pointless words, changing other word forms to approximations, and eliminating redundant terms. Each word/token in the preprocessed dataset has a distinct identity and is presented in a distinct and meaningful order. Every preprocessed input token's scattered representation is erudite by the embedding layer. The way the tokens are represented reflects the hidden connections between words that are probably going to appear in the same context. The GloVe model's pre-trained vectors were what we use. Utilizing LSTM networks may have a unique property of its own. We explained how LSTM includes a feature that allows it to recall the order of the data earlier in the section. The text data always contains a lot of unwanted information, which can be removed by the LSTM to shorten calculation times and lower costs, therefore this function also works on the eradication of useless information. The LSTM is a strong tool for text categorization and other text-based tasks since it has the ability to remove unnecessary information and recall the order of the information.

In this experiment, the loss function is binary cross entropy. The optimization approach is the ADAM optimizer, and the learning rate is 0.0001. Additionally, word embeddings have a dimension of 100. A 64 mini-batch is used. The CNN layer has 128 filters and a kernel size of 5 X 5, and the LSTM layer has 50 hidden nodes. After five iterations, the final categorization model is achieved. We compared our hybrid proposed sentiment analysis model with the basic LSTM and basic CNN model trained on the same dataset. The comparison results are shown in Table 2 and Figure 7.

V. Result And Discussion

The classification model's performance is assessed by showing the dispersal of classification ratios for different classes on test data for a given reality is considered as a confusion matrix. Table 3 displays the comparison for CNN, LSTM, and hybrid LSTM-CNN in terms of Accuracy, Recall, Precision, and F1-Score.

TABLE II. Experimental Results

Performance Metrics	CNN	LSTM	LSTM+CNN
Accuracy	92.60%	92.62%	93.98%
Precision	0.936	0.933	0.942
Recall	0.915	0.918	0.937
F1 Score	0.926	0.926	0.940

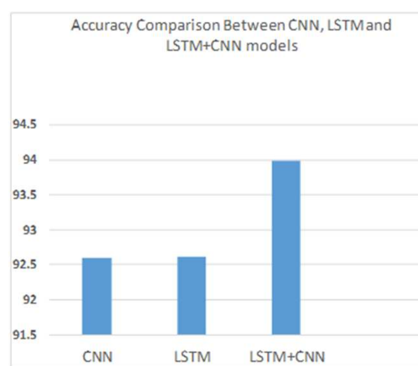

Figure 7. Accuracy Comparison

A. *Confusion Matrix*

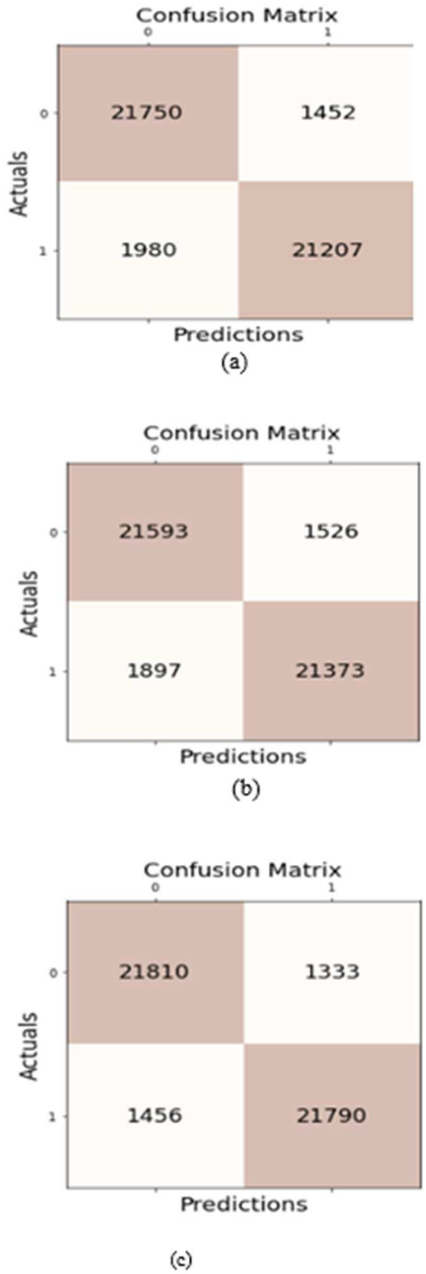

Figure 8. Confusion Matrix for (1) CNN, (2) LSTM, (3) LSTM+CNN

VI. CONCLUSION

In this study, we have collected a dataset of around 2,00,000 posts from Reddit using PRAW Reddit API. 80% of the dataset was used for training and the rest for testing purposes. Now, the data containing irrelevant terms were removed. After that, we extracted the features in word embeddings and for word embedding, we used Stanford's GLOVE which is the most basic method to examine the co-occurrence matrix. Now, this dataset was processed for three different models (CNN, LSTM, and LSTM-CNN) and the results were analyzed and compared for each of them.

After analyzing the data, we have compared the F1 score, Recall, Precision, and Accuracy for these three different models used in this study. The conclusions regarding the results obtained in this study are discussed below:

- The precision for the given dataset when processed in three models namely CNN, LSTM, and LSTM+CNN came out to be 0.936, 0.933, and 0.942 respectively.
- The resulting F1 score for three models (CNN, LSTM, LSTM+CNN) were 0.926, 0.926, and 0.940 respectively.
- The Recall for different models used in this study (CNN, LSTM, LSTM+CNN) were found to be 0.915, 0.918, and 0.937 respectively.
- At last the accuracy reported for all the models (CNN, LSTM, LSTM+CNN) resulted in 92.60%, 92.62%, and 93.98% respectively.
- A significant improvement can be seen in terms of accuracy, recall, and F1 score for proposed hybrid LSTM+CNN models in comparison to the other two models namely CNN and LSTM without having any decrement in precision for the same. Thus hybrid LSTM+CNN models prove to be the best among all three according to the investigation conducted in this study.

REFERENCES

[1] Kour, H., Gupta, M.K. An hybrid deep learning approach for depression prediction from user tweets using feature-rich CNN and bi-directional LSTM. Multimed Tools Appl (2022).

[2] Amanat, A.; Rizwan, M.; Javed, A.R.; Abdelhaq, M.; Alsaqour, R.; Pandya, S.; Uddin, M. Deep Learning for Depression Detection from Textual Data. Electronics 2022, 11, 676.

[3] Debasish Bhattacharjee Victor, Dewan Mamun Raza, "Depression Detection from Facebook using Machine Learning Techniques", 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), pp.1447-1454, 2022.

[4] Hochreiter, Sepp , and Schmidhuber, Jürgen. "Long Short-Term Memory." Neural Computation 9.8(1997):1735-1780

[5] G. Liu, X. Xu, B. Deng, S. Chen, and L. Li, ''A hybrid method for bilingual text sentiment classification based on deep learning,'' in Proc. 17th IEEE/ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput. (SNPD), May 2016, pp. 93–98.

[6] Pitsilis, Georgios K., HeriRamampiaro, and Helge Langseth. "Effective hate-speech detection in Twitter data using recurrent neural networks." Applied Intelligence 48.12 (2018): 4730-4742.

[7] A.S. Imran, S. M. Daudpota, Z. Kastrati and R. Batra, "Cross-Cultural Polarity and Emotion Detection Using Sentiment Analysis and Deep Learning on COVID-19 Related Tweets," in IEEE Access, vol. 8, pp. 181074-181090, 2020, doi: 10.1109/ACCESS.2020.3027350.

[8] Clark, Eric M., et al. "A sentiment analysis of breast cancer treatment experiences and healthcare perceptions across twitter." arXiv preprint arXiv:1805.09959 (2018).

[9] Yao, Hannah, et al. "Detection of suicidality among opioid users on reddit: machine learning–based approach." Journal of medical internet research 22.11 (2020): e15293.

[10] Orabi AH, Buddhitha P, Orabi MH, Inkpen D (2018, June) Deep learning for depression detection of twitter users. In: Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic, pp 88–97.
[11] Zogan H, Wang X, Jameel S, Xu G (2020) Depression detection with multi-modalities using a hybrid deep learning model on social media. arXiv preprint arXiv:2007.02847. https://arxiv.org/ftp/arxiv/papers/2003/2003.04763.pd
[12] Jianqiang, Z., Xiaolin, G., & Xuejun, Z. (2018). Deep convolution neural networks for twitter sentiment analysis. IEEE Access, 6, 23253–23260.
[13] Han, J., Qian, K., Song, M., Yang, Z., Ren, Z., Liu, S., & Li, X. (2020). An early study on intelligent analysis of speech under COVID-19: Severity, sleep quality, fatigue, and anxiety. arXiv preprint arXiv: 2005.00096.
[14] A.S. Imran, S. M. Daudpota, Z. Kastrati and R. Batra, "Cross-Cultural Polarity and Emotion Detection Using Sentiment Analysis and Deep Learning on COVID-19 Related Tweets," in IEEE Access, vol. 8, pp. 181074-181090, 2020, doi: 10.1109/ACCESS.2020.3027350.
[15] L. Yang, Y. Li, J. Wang and R. S. Sherratt, "Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning," in IEEE Access, vol. 8, pp. 23522-23530, 2020, doi: 10.1109/ACCESS.2020.2969854.
[16] Jelodar Hamed, Yongli Wang, Rita Orji, Hucheng Huang, Deep sentiment classification and topic discovery on novel coronavirus on covid19 online discussions: Nlp using lstm recurrent neural network approach, IEEE J. Biomed. Health Inf. 24 (10) (2020) 2733–2742.
[17] Dashtipour, K.; Gogate, M.; Adeel, A.; Larijani, H.; Hussain, A. Sentiment Analysis of Persian Movie Reviews Using Deep Learning. Entropy 2021, 23, 596.
[18] Rehman, A.U., Malik, A.K., Raza, B. et al. A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis. Multimed Tools Appl 78, 26597–26613 (2019).