Expert Systems With Applications 213 (2023) 119007

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

An attention-based hybrid architecture with explainability for depressive social media text detection in Bangla

Tapotosh Ghosh [a], Md. Hasan Al Banna [b], Md. Jaber Al Nahian [c], Mohammed Nasir Uddin [c], M. Shamim Kaiser [d,*], Mufti Mahmud [e,f,g,**]

[a] Department of Computer Science and Engineering, United International University, Dhaka, Bangladesh
[b] Department of Computer Science and Engineering, Bangladesh University of Professionals, Dhaka, Bangladesh
[c] Department of Information and Communication Engineering, Bangladesh University of Professionals, Dhaka, Bangladesh
[d] Institute of Information Technology, Jahangirnagar University, Savar, Dhaka, Bangladesh
[e] Department of Computer Science, Nottingham Trent University, Nottingham NG11 8NS, UK
[f] Medical Technologies Innovation Facility, Nottingham Trent University, Nottingham NG11 8NS, UK
[g] Computing and Informatics Research Centre, Nottingham Trent University, Nottingham NG11 8NS, UK

ARTICLE INFO

Keywords:
Depression
Social media
Attention
Mental health
Suicide

ABSTRACT

Mental health has become a major concern in recent years. Social media have been increasingly used as platforms to gain insight into a person's mental health condition by analysing the posts and comments, which are textual in nature. By analysing these texts, depressive posts can be detected. To facilitate this process, this work presents an attention-based bidirectional Long Short-Term Memory (LSTM)- Convolutional Neural Network (CNN) based model to detect depressive Bangla social media texts, which is lighter and more robust than the conventional models and provides better performance. A dataset containing such Bangla texts was also developed in this work to mitigate the scarcity. Different preprocessing stages were followed, and three embeddings were used in this task. Thanks to the attention mechanism, the proposed model achieved an accuracy of 94.3% with 92.63% of sensitivity and 95.12% of specificity. When tested on other languages, such as English, the proposed model performed remarkably. The robustness and explainability of the proposed model were also discussed in this paper. Additionally, when compared with classical machine learning models, ensemble approaches, transformers, other similar models, and existing architectures, the proposed model outperformed them.

1. Introduction

Depression is a mood disorder that involves a persistent feeling of sadness and loss of interest. It is different from the mood fluctuations that people regularly experience as a part of life. Depression comes from two Latin words, "depressare" and "deprimere". A literal translation of "deprimere" is "press down"; "de" means "down" and "premere" means "to press". It implies a feeling of heaviness, one of being 'pressed down,' and can also be called "saddened", "blue" or simply "down" (Kanter, Busch, Weeks, & Landes, 2008). Since the ancient Greek physician Hippocrates, depression has been mentioned in medical literature (Tipton, 2014). Depression can be caused by several different factors. Life events can trigger depressive episodes or aggravate them in the case of unfavourable life circumstances. Furthermore, a sense of unworthiness and negativism about oneself and the world contribute to depression (Addis & Jacobson, 1996). Among the most common causes of depression, there are various types of abuse, sexual orientation, gender, class, financial situation, unemployment, loss of close relationships, and feeling alone.

According to a report by World Health Organization, almost 264 million people are suffering from depression-related disorders, which is 3.4% of the world population (Spencer, Degu, Kalkidan, & Solomon, 2018). The rate of depression is higher among 15–29 years aged adults. This rate is increasing day by day. As depression is increasing, the suicide rate has also become a major concern. Around 800,000 people lose their lives every year in suicide events (WHO, 2014). In every 40 s, a person dies due to suicide. Prevalence of suicidal events is higher in the lower and mid-level income countries (Wang et al., 2007). Suicide has already become the fourth leading cause of death among

* Correspondence to: Institute of Information Technology, Jahangirnagar University, Savar, Dhaka - 1342, Bangladesh
** Corresponding author at: Department of Computer Science, Nottingham Trent University, Nottingham NG11 8NS, UK.
E-mail addresses: tapotoshghosh@gmail.com (T. Ghosh), alifhasan39@gmail.com (Md.H.A. Banna), nahianrism@gmail.com (Md.J.A. Nahian), nasiruddin@bup.edu.bd (M.N. Uddin), mskaiser@juniv.edu (M.S. Kaiser), mufti.mahmud@ntu.ac.uk, muftimahmud@gmail.com (M. Mahmud).

https://doi.org/10.1016/j.eswa.2022.119007
Received 27 October 2021; Received in revised form 10 August 2022; Accepted 9 October 2022
Available online 15 October 2022

15–29 years aged people (WHO, 2021b). Around 2%–6% people in each and every country in the world are depressed (Hannah Ritchie, 2017). Around 1.5% of the total death in 2017 was due to suicide which varies from 0.2% to 5% from country to country. Therefore, depression has already become a major concern in the current world.

Social media has become an integral part of almost everyone nowadays. People love to share their thoughts and emotions through it. According to a report in 2021 (datareportal, 2021; Dean, 2021), the total number of social media users has reached 3.96 billion, which is 50.6% of the world population, and 13.7% more than the previous year. Around 521 million people joined social media between 2020 and 2021. People, on average, spend 142 min on social media per day. Most of the people use Facebook, which is 71.5% of the total social media users (StatCounter, 2021). Twitter and Pinterest come next to it. A debate has been going on for a while about social media's benefits and negative effects on mental health (Berryman, Ferguson, & Negy, 2018). In order to protect mental health, social networking is crucial. A person's health, mental behaviour, physical health, and mortality risk are affected by the quantity and quality of social relationships (Martinsen, 2008). Therefore, there may be a connection between social media and mental health, and the Displaced Behaviour Theory can help to explain it. It has been shown that people who spend more time using social media or performing sedentary activities have less time to participate in face-to-face interactions, both of which are protective against mental illness (Coyne, Rogers, Zurcher, Stockdale, & Booth, 2020; Escobar-Viera et al., 2018). Nevertheless, social theories claim that there is evidence that social media cause mental health problems by influencing how people interact with and maintain their social networks (Rahman et al., 2013). Several researchers have studied the effects of social media on mental health, and they have found that prolonged use of social media platforms such as Facebook, Twitter, Snapchat may be associated with depression, anxiety, and stress symptoms (O'Reilly et al., 2019, 2018). Fear of missing out, feeling less important, insecurities about life, etc., can be formed due to excessive use of social media. In addition, social media can amplify any stereotype that anyone wants to create and enable him to be as popular as others, which is commonly seen nowadays.

Social media can be used to detect and prevent depression as well. As people stay active in social media, they generate a lot of contents. His activity can be a great tool for detecting potential depressive disorder. Depressed people tend to share and generate negative contents. Their writings would be quite different from non-depressive people. Their approach towards the activity of other people will be always a major concern. Depressive people generally suffer from insomnia, so they tend to be active on these social media platforms during that time which is unusual. By detecting or predicting depression in a person, social media can also be used as a platform to fight against this sort of disorder. For example, bots can be developed to talk with them to help them in these tough times. Inspirational and optimistic videos can be flooded to their timeline, which can create positivity among themselves. Depressive contents may also be hidden or given lower priority in case of appearing in their timeline as well, which will help them to be in a good mental state.

Several researches have been carried out to detect depressive social media texts. Mumu, Munni, and Das (2021) took a Long Short-Term Memory (LSTM) - Convolutional Neural Network (CNN) hybrid approach to detect depression from Bangla social media posts. The CNN layers were added to perform feature mapping, where LSTM layers were responsible for extracting features. After that, the features were gone through a set of fully connected layers to predict the label of the status. Classical machine learning (ML) algorithms were adopted by Billah and Hassan (2019) to detect depression. They used Facebook statuses and calculated Term Frequency-Inverse Document Frequency (TF-IDF) in order to predict the depression status of a person. SGD classifier achieved the best result. Uddin, Bapery, and Arif (2019a) proposed a 5-layered LSTM model with 128 neurons to classify depressive Bangla social media content. A Gated Recurrent Unit (GRU) based method was adopted by Uddin, Bapery, and Arif (2019b). They created several combinations of GRU and Dense layers and finally found a 5-layered GRU with 512 neurons as the optimal model. Ahmad, Asghar, Alotaibi, and Hameed (2020) took a BiLSTM approach to detect depression from English tweets. BiLSTM model with 215 neurons and 10 filters outperformed several LSTM, GRU, and Recurrent Neural Network (RNN) combinations in this task. Deshpande and Rao (2017) adopted a classical ML approach to detect depression. Naive Bayes (NB) performed the best among the ML models in terms of accuracy, precision, and recall. Zogan, Razzak, Jameel, and Xu (2021) proposed a social media-based depression detection framework that dealt with the posts and also some user behaviour features, which was named as "DepressionNet". They extracted features using BERT-BART models and classified depression using a CNN-GRU stacked model. Reddit users' posts were examined by Tadesse, Lin, Xu, and Yang (2019) to identify any factors that may indicate depression attitudes. They used Linguistic Inquiry and Word Count (LIWC), Latent Dirichlet Allocation (LDA), N-gram features, and classical machine learning algorithms in their framework. Islam et al. (2018) proposed a classical ML approach to detect depressive social media posts. They processed the textual data using LIWC. Then, K nearest neighbour (KNN) was introduced to classify texts. Depression is reflected on the writings of a person. Depressed people tend to use first person, possessive pronouns, personal pronouns, and depressive words more than normal persons. Trotzek, Koitka, and Friedrich (2018) tried to utilise this concept to early detection of depressive text from social media messages. To perform this task, they tried out several word embeddings and also 17 metadata which was fed into a CNN model that performed quite well. Suman, Shalu, Agrawal, Agrawal, and Kadiwala (2020) created a mobile application that detected depression through Twitter using deep learning (DL). RoBERTA was used as the backend model to compare the tweets with inbound tweets that had been earlier labelled. Cong et al. (2018) proposed a 2 layered architecture to detect depression from Reddit posts. At first, the XGBoost algorithm was used to classify text as negative or positive. If the text was negative, it was considered non-depressive. If it was positive, it went through a BiLSTM model. Venkataraman and Parameswaran (2018) proposed a depression detection system from videos of students. They detected the face using a Viola–Jones face detector and extracted features from the face using a Gabor filter. Support Vector Machine (SVM) algorithm classified depression from the extracted features. Chiu, Lane, Koh, and Chen (2021) tried to detect depression from the social media activity of users. They took a multi-modal approach where they considered images, texts, and the behaviour of the user. They used the AlexNet model for image classification, a 2-layered LSTM model in text classification, and Random Forest (RF) classifier for classifying behavioural data. All 3 scores were merged using the Adaboost classifier for final depression prediction. Ghosh et al. (2021) detected depressive English tweets using an LSTM-CNN model. They further used this model to analyse the impact of COVID-19 on mental health. Peng, Hu, and Dang (2019) proposed a framework to predict depression from Sina Weibo, a Chinese micro-blog which has a large number of users. To classify texts, they extracted TF-IDF features. They also extracted features from the user profile, behaviour, and emotional feature of the text. All the features were merged to classify depression. Multi Kernel SVM performed the best in the classification task.

Social media analysis can be effective in fighting depression. But, due to the huge explosion in the usage of social media, it is impossible to analyse social media content by humans. Artificial Intelligence (AI)-enabled systems should be developed for it. Another problem is the diversity in the language of the content. People love to talk and express their feelings in their mother tongue. Using a native language is also very common in social media. So, analysing social media contents of all languages is a crying need. But, most of the research in natural

language processing (NLP) is dealt with the English language. In this work, Bangla, which is a low-resource language, has been chosen. Bangla is one of the most popular languages, with 228 million native speakers. It is also the mother tongue of the Bangladeshis, who hold a very high share of social media users. Long Short Term Memory (LSTM) models are very popular in sequence and time series data, but their performance degrades in longer sequences which is noticeable on social media. An attention mechanism can solve this problem. CNN is quite useful in finding deep feature sets which assist in improving performance. Therefore, attention, LSTM, and CNN-based model have been proposed to detect Bangla depressive social media texts. This model can be adopted in any language, which is also proven in this paper. The main contribution of this research is given below:

- Establishing an effective and robust Attention-LSTM-CNN based architecture for Bangla depressive text detection through searching the optimal model using Keras tuner and deep diving into the performance of the best performing models. To the best of the author's knowledge, attention was never used for Bangla depressive text detection. A significant performance was achieved by this model in this study.
- Creating a dataset of more than 15,000 samples for Bangla depressive text detection. The advancement of Bangla NLP is very slow, and it will surely contribute a lot to the Bangla NLP.
- Comparing Performance with ensemble models, existing architectures, transformers and ML models to establish the supremacy of the proposed model.
- Discussed the impact of preprocessing stages, feature reduction, and different types of embeddings in the performance of the proposed model.
- Testing the robustness of the proposed model. The model was tested in an English depressive social media text detection dataset where it provided a significant result which proved its capability to achieve a very good performance in any language.
- Explaining the reasons behind classifying a sample as depressive or non-depressive is essential as it is related to mental health problems.

The paper is organised in the following manner: Section 2 describes the attention mechanism. The methodology of this work is discussed in Section 3. Results and comparison with related architectures have been provided in Section 4. The Explainability of this model has been discussed in Section 5. Section 6 wraps up this paper with some recommendations and research scopes in this field.

2. Attention mechanism

Bahdanau, Cho, and Bengio (2014) first used the attention mechanism for the machine-to-machine translation-related tasks in 2015. This concept was mainly derived for NLP tasks. But, nowadays, it is useful in different ML applications (Al Nahian et al., 2020). LSTM model was introduced to deal with larger sequences. Still, its performance degrades as the sequence length increases due to the difficulty in providing focus in a relatively larger sequence. It is also impossible for LSTMs to focus on a specific part of a sequence. Therefore, an attention mechanism has come to light to solve this problem. If the input sequence is considered as $X_1, X_2, \ldots, X_T$ and the output is y_i at a given time i, the conditional probability of an output event can be found using Eq. (1).

$$P(y_i|y_1, \ldots, y_{i-1}) = g(y_{i-1}, s_i, c_i) \tag{1}$$

Here, the hidden state is denoted as s_i. It can be found using Eq. (2).

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \tag{2}$$

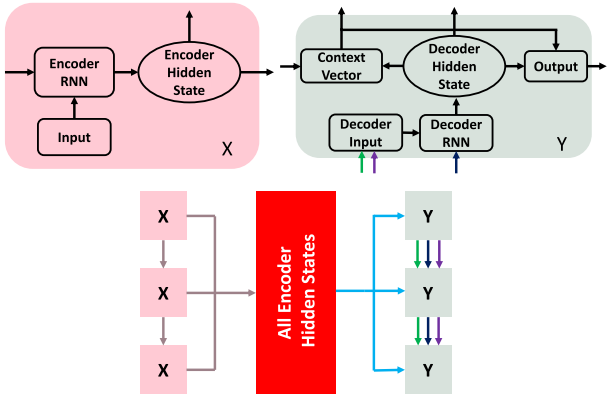

Fig. 1. Attention mechanism architecture. Attention architecture consists of two parts: an encoder and a decoder. X and Y represent the encoder and decoder blocks, respectively. Previous hidden states of the encoder are provided as input to the encoder RNNs. On the other hand, the previous hidden states and the decoder input are used as input to the decoder RNN block. The decoder input is based on the previous context vector's output and the overall output. To generate the context for overall output, encoder hidden states are used.

Here, the context vector denoted as c_i controls the amount of attention that should be provided to a portion of the sequence during calculating the output. It is dependent on the annotations $(h_1, h_2, \ldots .h_{Tx})$, where h_i contains information about the entire sequence, with further emphasis on some parts surrounding the ith position. The context vector c_i is calculated using Eq. (3).

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{3}$$

Here, α_{ij} represents the weights multiplied by each portion of the sequence, which is determined by the softmax operation. Mathematical representation of is provided in Eq. (4).

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{4}$$

Here, e_{ij} represents the model's alignment, which is dependent on s_{i-1}, and h_j. A feed-forward neural network (FFNN) is used to calculate it, which trains automatically during the training of the whole model. With these operations, each output is calculated with the weighted sum of the input sequence, where the weights are elements of the context vector. Fig. 1 illustrates the attention mechanism.

3. Methodology

In this work, an attention-based BiLSTM-CNN approach has been proposed to detect depressive Bangla social media texts. As no publicly available dataset exists, a new dataset has been developed. Then, the textual data went through a preprocessing stage comprised of tokenization, removing punctuation, stemming, removing stop words, and finally creating input tensor phases. Afterwards, features were extracted using pre-trained embeddings from the texts to provide input to the proposed model, which classified the input text as depressive or non-depressive.

This framework was found after searching through various criteria using the Keras tuner from which eight different models, including the proposed model, were found as best performing. To evaluate this framework's performance, performance of these eight models were also considered, and their performances were found by training and testing the models on the same dataset. An ensemble of the three best-performing attention-based models was also built to compare with the proposed model. The robustness of the proposed model was also tested by taking different training and testing sets, evaluating performance

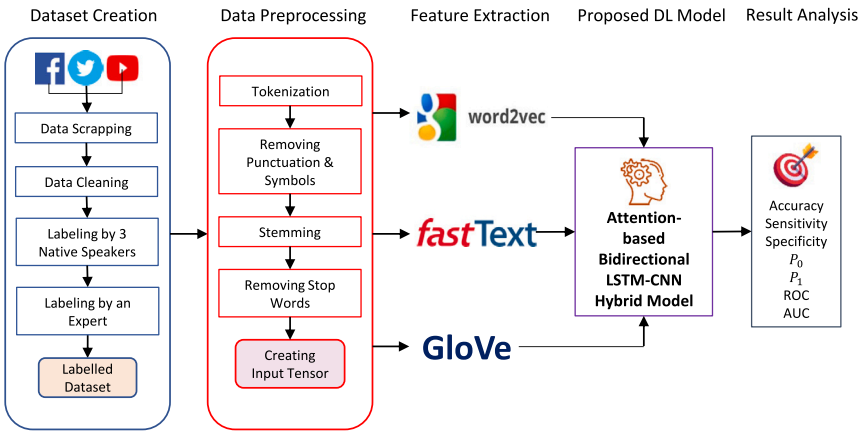

Fig. 2. Overview of the framework. The proposed framework comprises dataset creation, data preprocessing, feature extraction, model creation, and result analysis phases.

Table 1
Dataset characterisation table.

Platforms	Facebook, Twitter, YouTube
Type of data	Text
Language	Bangla
Identity of the participants	Anonymous (Only public posts were considered)
Gender biasness	Unknown (Anonymous collection)
Location of posts	Bangladesh, India (Kolkata)
Total scrapped samples	16000
Removed sample before labelling	969
Considered sample for labelling	15031
Conflict among native speaker annotators	1353 (9%)
Conflict among annotators and expert	302 (2%)
Total no of words	36380
Total no of depressive samples (1)	4784
Total no of non-depressive samples (0)	10247
Total no of training samples	12000 (1:3806, 0:8194)
Total no of testing samples	3031 (1:978, 0:2053)

in an English depressive social media text dataset, and classifying real suicidal Bangla texts. The impact of attention layer, embedding, feature reduction and preprocessing is also evaluated in this paper. Finally, the reason behind the performance of the proposed framework was explained (see Fig. 2).

3.1. Dataset preparation

A large dataset was created because there is no publicly available dataset of Bangla depressive social media text. At first, data was scrapped from social media, then labelled as depressive or non-depressive. Finally, data were divided into two portions: train and test set. Characteristics of the developed dataset have been provided in Table 1.

3.1.1. Data scrapping and cleaning

As most of the Bengali-spoken people use Facebook, Twitter, and YouTube, around 16,000 Bangla statuses were scrapped from these platforms. To scrap this, Twitter and YouTube API have been used. Facebook posts were scrapped manually from depression related pages, groups or hashtags. After scrapping data, statuses shorter than 3 words were removed. At the end of this phase, 15,031 social media statuses were there for labelling.

3.1.2. Data labelling

In this phase, the data is labelled as 0 if the status is non-depressive, and it was labelled as 1 in the case of depressive content. This phase can be divided in to 2 stages: labelling by native speakers, and labelling by expert.

Labelling by native speakers. Here, all the statuses were labelled by 3 Bangla native speakers who are university students. They labelled these contents manually. After labelling by them, majority voting was performed. All three annotators provided the same label in 91% of the cases (13,678 samples). The conflict was found in 1353 samples. These labels were provided to an expert for checking.

Labelling by expert. A psychologist labelled these contents manually and checked the labelling by native speakers. Only labelling of 2% contents (302 samples) were overturned from the labelling provided by the native speakers. Labelling provided by the expert was considered as final in this dataset.

3.1.3. Dataset splitting

After labelling, the dataset contained 4784 depressive and 10,247 non-depressive posts. As the non-depressive post is much more present on social media, it is reflected in the dataset. To train and evaluate the proposed model, the dataset was divided at an 80:20 ratio, where 12,000 statuses were considered for training, and 3032 samples were kept aside for testing, which was not revealed in the training phase. The models built for performance comparison were also evaluated using the same training and testing samples.

3.2. Dataset preprocessing

After creating the dataset, the textual data was gone through a preprocessing stage. Here, at first, the text was tokenised, punctuation and words from other languages were removed and stemmed and stop words were removed from the tokens. Afterwards, tokens were converted into a numerical vector and later converted to a fixed 140-sized vector by adding padding 0s.

3.2.1. Tokenization

Tokenization is breaking the text into a list of words. In this work, a tokeniser which is available in the BNLTK library was used.

3.2.2. Removing punctuations, emoticons, and words of other languages

All the characters not included in the Bangla language were removed from the text. Emoticons were also removed in this work.

3.2.3. Stemming

Stemming is a technique used to extract the base form of the words by removing affixes from them. It is just like cutting down the branches of a tree to its stems. For example, if we stem the word "eating", "eaten", we will get "eat". It extracts the base word, which reduces the number of unique words in the dictionary. A pre-trained stemmer Kaium (2021) was used in this work.

3.2.4. Removing stop words

Words that are commonly used in a language and donot have significant meaning are called stop words. The removal of stop words is common in text mining and NLP to get rid of words that are so frequent that they have little to no useful meaning or value. A list of 439 mostly used stopwords in the Bangla language was created and removed from the text in this work.

3.2.5. Creating input tensor

DL models cannot take text as input. So, the input tensor was converted to a list of numerical values. Here, a dictionary of unique words used in the corpus was created, which had a numerical key and the word as the value. Words in the vectors were replaced by the dictionary key. Then, all the vectors were converted to the same size by adding padding 0s.

3.3. Word embedding

Word embedding refers to the learning of similar representations for words that share the same meaning. The words are represented in a predefined vector space as a real-valued vector of floating-point values (Ghannay, Favre, Esteve, & Camelin, 2016; Lai, Liu, He, & Zhao, 2016; Levy & Goldberg, 2014). Since the word mapping is done in a way similar to the creation of a neural network, this technique is often referred to as deep learning. This technique captures the contextual meaning of a word in a document, analyses its semantic and syntactic similarity, and expresses relationships among words.

If we create a one-hot representation of two sentences, "Have a nice time", and "Have a great time", then the vocabulary would be V = Have, a, nice, great, time. One hot representation of "nice" and "great" would be [0, 0, 1, 0, 0] and [0, 0, 0, 1, 0] respectively. Here "nice" and "great" represent quite similar meanings. But due to representation in one-hot encoding, these two words are getting quite different representations. It is not capturing any sort of relationship. Word embedding comes in handy in this sort of situation where similar representation is needed in the case of similar words. It creates a dependency between two words which is missing in word frequency-based representations.

3.3.1. Word2Vec

Word2Vec is a deep neural network that reconstructs linguistic contexts using two layered shallow neural networks. An enormous corpus of words is fed into the system, which generates a vector space, with the individual words of the corpus assigned a corresponding vector in the space. There are two ways to obtain it: Skip Gram and continuous Bag Of Words (CBOW).

In CBOW, one hot vector of multiple words is considered as input, and the neural network predicts the corresponding word based on the context. The model is trained using the error between the predicted one hot encoding and the actual target's one hot encoding vector. In this way, a vector representation of the target word is learned. Skip Gram is just the opposite of the CBOW model. A word is given as the input, and the model provides a probability distribution from which multiple words are predicted.

Skip Gram is useful in the case of a small amount of data and rare words, whereas CBOW is faster and provides better representation in the case of common words. A pre-trained 100-dimensional Word2Vec has been used in this work which was trained on wiki dump dataset (BNLP, 2021a, 2021b).

3.3.2. fastText

fastText is another word embedding method that converts words into vectors. In this embedding, a word is broken into n-gram of characters rather than a single word (Joulin et al., 2016; Liao, Shi, Bai, Wang, & Liu, 2017; Wu & Manber, 1992). For example, we want to get the embedding of the word "artificial" with n=3. In that case, the representation using fastText embedding of this word is <ar, art, rti, tif, ifi, fic, ici, ial, al>, where the angular brackets denote the beginning and end of the word. It also enables embeddings to understand suffixes and prefixes, which helps them capture the meaning of shorter words. Then, a skip-gram training is done to determine the word's embeddings based on its character n-gram representation. Since no internal structure of the word is considered, this model can be considered a bag of words model. Moreover, n-grams need not be in any particular order, as long as they are within this window.

fastText handles rare words well. Even if a word was not used during training, its embedding could still be determined by breaking it down into n-grams. Conversely, Word2vec and GloVe cannot deal with unknown words missing in the model dictionary, making fastText more suitable for low-resource languages. In this work, a pre-trained fastText word embedding has been used, which was trained with 20 million words and for 50 epochs. The dimension was set to 100.

3.3.3. GloVe

By not fully exploiting the statistical information about word co-occurrences, Word2Vec's online scanning approach is suboptimal. GloVe was built to solve this problem (Pennington, Socher, & Manning, 2014). This method was based on two concepts: global matrix factorisation and local context window. A global matrix factorisation method is a way to reduce large-term frequency matrices by the use of matrix factorisation. In these matrices, words are shown as occurrences or absences throughout a document. There are two methods for local context windows: CBOW and Skip-Gram. In GloVe, rather than taking the embeddings using neural networks designed for specific tasks, such as neighbour-word prediction (CBOW) or focus word prediction (Skip-Gram), the embedding vectors are generated in a way in which the dot product of two words equals the log of the number of times the two words occur together. As an example, if the words "cricket" and "badminton" occur 20 times in the document corpus within a 10-word window, then: $Vector(cricket).Vector(badminton) = log(10)$ would be used to generate word vector. The model is compelled to represent the frequency distribution of nearby words in more global contexts. In this work, a pre-trained GloVe has been used, which was trained on 39M token and contained 0.18 million vocabularies (Sarker, 2021).

3.4. Proposed depressive social media text detection model

Fig. 3 depicts the proposed model for detecting depressive Bangla social media texts. This model was found after trying out several combinations of LSTM, CNN, Max Pooling, Attention, and Dense layers using the Keras tuner. We tried different number of layers (0–3) of BiLSTM, LSTM, Convolution, Attention, and dense layers in different units/filter/neuron range (shown in Table 2). After tuning in this range, eight different models were shortlisted, which is mentioned in Table 3. These models were trained for 30 epochs, and a separated testing set was used to evaluate the performance. Based on this performance, the model illustrated in Fig. 3 was considered optimal.

Table 2
Model tuning range.

Layer	No of layers tried	Tuning range
BiLSTM	0–3	25–500 units
LSTM	0–3	25–500 units
Convolutional layer	0–3	25–500 filters
Attention	0–1	Self/Global, L1/L2 regularisation
Dense layers (ReLU)	1–4	25–1000 neurons
Dense layers (Softmax activation)	1	2 neurons

Table 3
Architecture of the considered models after model searching phase.

Model	BiLSTM (100)	LSTM (100)	LSTM (100)	Conv1D (50)	Attention	Max pooling	Flatten
BiLSTM-CNN	✓	✓	✗	✓	✗	✓	✗
Attention-LSTM-CNN	✗	✓	✓	✓	✓	✓	✗
LSTM-CNN	✗	✓	✓	✓	✗	✓	✗
Attention-BiLSTM	✓	✓	✗	✗	✓	✗	✓
BiLSTM	✓	✓	✗	✗	✗	✗	✓
Attention-LSTM	✗	✓	✓	✗	✓	✗	✓
LSTM	✗	✓	✓	✗	✗	✗	✓

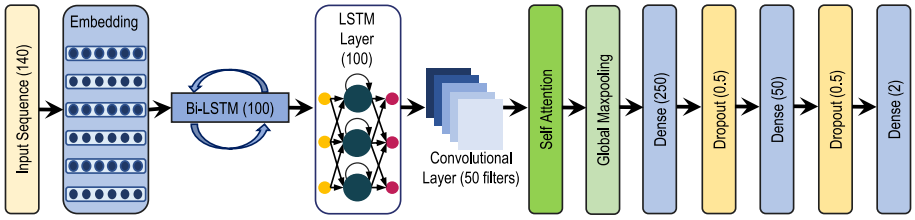

Fig. 3. The proposed Attention-based bidirectional LSTM-CNN hybrid model. This model consists of a bidirectional LSTM layer, an LSTM layer, a Convolutional Layer, a Luong attention layer and three dense layers.

The proposed model took the tensor found after preprocessing stage as input. Then an embedding layer replaced the tensor contents with an embedding matrix which was predefined in the pre-trained embeddings. The output of this layer was the input of a bidirectional LSTM layer with 100 neurons and a sigmoid activation function to grasp both present and past knowledge. Then, a 100-neuron LSTM layer with a sigmoid activation function was used. After that, a convolutional layer with 50 filters and a ReLU activation function was used to extract higher-level features. Then, a multiplicative attention layer was introduced to provide importance to significant features. L1 and L2 regularisation was used with this layer to reduce overfitting. The activation function of the attention layer was ReLU. Then, a 1D max pooling layer was introduced to reduce dimensionality and converted the output of the attention layer to a 1D feature vector that was used as an input to a fully connected dense layer with ReLU activation function and 250 neurons. Another dense layer with 50 neurons was stacked on it. Finally, a dense layer with two neurons and a softmax activation function was used to predict the label of the input text. 50% dropouts were utilised between the dense layers to reduce overfitting.

Let, the input tensor is I, where $I = [152, 12033, \ldots., 27000]$. After the embedding layer, it would be,

$$x_t = \begin{bmatrix} [-0.52, 0.9, \ldots .., 1.52], \\ [0.78, 5.1, \ldots .., -3.70], \\ . \\ . \\ . \\ [0.13, -0.8, \ldots .., 3.30] \end{bmatrix} \tag{5}$$

I_e will be the input to the BiLSTM and an LSTM layer. LSTMs were developed to solve gradient vanishing and gradient exploding problems so that the model can cope with a large context while dealing with large gaps between the present and past. During forward propagation, it processes information as well. Still, its cell structure is quite different, which allows it to retain important information and transmit it through the long chain of sequences (Hochreiter & Schmidhuber, 1997). LSTM layers are mainly constructed using three gates: input gate, output gate, forget gate and a cell state. The memory of the network is stored in the cell state, which is capable of carrying relevant information to the sequence. The gates can recognise the relevant information to keep and forget the rest of the information. The forget gate is used to erase irrelevant information. The input gate selects the relevant information to add, and the output gate sets the following hidden state information.

At first, current input will be passed through the forget gate. Values from current input and previous hidden state propagate through a sigmoid function, which translates to a value between 0 and 1 where the values which are close to 1 are kept in the memory and close to 0 are forgotten from the network. If the previous hidden state information is h_{t-1}, and x_t is the current input. The output of the forget gate is,

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f), \tag{6}$$

where the weights of the forget gate and bias are W_f and b_f, respectively. The cell state is the memory of LSTM, and the input gate is used to update it. At first, the recent inputs and hidden state information are passed through a sigmoid function to transform it to be within 0 to 1, which indicates the importance of the information. The current input and hidden state information also go through a tanh function, which regulates the network. The output of these functions is multiplied, and when the output of the sigmoid is 0, the information is thrown away, and when it is 1, it is stored in the cell. If the tanh activation output is $\tilde{C}_t$ and sigmoid function output is i_t then,

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i), \tag{7}$$

$$\tilde{C}_t = tanh(W_C \times [h_{t-1}, x_t] + b_C), \tag{8}$$

where the weights of the input gate and cell state are W_i and W_C respectively, the biases of the input gate and cell state are b_i and b_C respectively. The cell state gets updated through a point-wise multiplication of the forget gate's output and adding it with the input gate's

output through point-wise addition. Values are dropped in point-wise multiplication if it is multiplied by 0. If the previous state information is C_{t-1} and the current state information is C_t, then,

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t. \tag{9}$$

In the structure, the output gate determines the next hidden state. Prior hidden information and current inputs are converted to a value between 0 and 1 using sigmoid activation. A tanh activation function is applied to pass the updated cell state information, and its output is multiplied by the output of the sigmoid activation function. The hidden state carries the information that is multiplied. If the sigmoid output is o_t in this gate and the output of this gate is h_t, then,

$$o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o) \tag{10}$$

$$h_t = o_t \times tanh(C_t) \tag{11}$$

where W_o is the weight of the output gate, and b_o is the bias of the output gate. Here, h_t will be the output of the LSTM layer. This model used one BiLSTM layer and a forward LSTM layer. Using forward and backwards passes, bi-directional LSTM extends LSTM networks to retain past and future knowledge. The bi-directional LSTM has a better grasp of function meaning than general LSTM. The same LSTM layer is used in the BiLSTM network layout, but one of the LSTM layers has the reverse input direction. The performance of bi-directional LSTM networks is generally better than that of one-way networks.

After the LSTM networks, a convolutional layer was used to extract deep features from the output h_t. The output of the convolutional layer was,

$$Z = h_t * f \tag{12}$$

Here, f is the filter of this layer which performs convolution operation with the input h_t. In this model, the filter size was 50, and the stride was 1. The output of this layer is Z, which was the input of the attention layer.

This work uses Luong attention as the attention layer due to faster runtime than additive attention. L1 and L2 regulariser were used with it. Then, a maxpooling layer was there to reduce the dimensionality of the feature matrix. The output of this layer was the input of the three dense layers, which predicted an input text as depressive or non-depressive.

Several other models were also created for performance comparison. Seven other attention, LSTM, and CNN-based models were built. An ensemble of the proposed and 2 of the seven other models were developed to evaluate performance. Several existing models found in literature and classical ML classifiers were also developed to compare with the performance of the proposed model. The robustness of the proposed model, the impact of preprocessing, the impact of word reduction, and the impact of the selection of embedding layer were also evaluated in this paper.

3.5. Training and testing

The models were trained and tested using the previously separated training and testing sets. The training set contained 12,000 samples, whereas the testing set had 3031 samples. The models were trained with a learning rate of 0.001, where batch size was 32, the optimiser was Adam, and the loss function was categorical cross-entropy. The model was trained for 30 epochs. All the experiments were conducted on Kaggle Kernel. It provides 4 CPU cores, 16 Gigabytes of RAM, and NVIDIA Tesla P100 GPU.

4. Results and discussion

In this section, the result of the proposed model and comparison of the performance with related models will be discussed. The robustness of the model, the impact of preprocessing, the impact of embedding, and the impact of the attention layer will also be described.

Table 4
Performance of the attention-based BiLSTM-CNN hybrid model.

Metrics	Embedding		
	fastText	Word2Vec	Glove
Accuracy	**0.94325**	0.9244	0.9267
TN	**1953**	1914	1911
FP	**100**	139	142
FN	**72**	90	80
TP	**906**	888	898
Sensitivity	**0.9263**	0.9079	0.9182
Specificity	**0.9512**	0.9322	0.9308
P_0	**0.9644**	0.9551	0.9598
P_1	**0.9005**	0.8646	0.8635

4.1. Result of the proposed model

3 types of embedding were used during classifying depressive texts. Among them, by using the fastText embedding, the proposed Attention-based BiLSTM-CNN hybrid model performed the best in every evaluation metrices. This model achieved 94.33% accuracy using fastText embedding, while it achieved less than 93% accuracy using Word2Vec and GloVe embedding. The model also achieved a very good sensitivity of 92.63%, where it achieved 90.79% and 91.82% sensitivity by using Word2Vec and GloVe embedding respectively. The specificity of this model is significant (95.12%) while using fastText embedding. Table 4 provides the detailed result of this model.

Fig. 4(a) shows the confusion matrix of this model by adopting fastText embedding. This model correctly classified 906 out of 978 samples of depressive posts, and 1953 out of 2053 non-depressive posts. This model provided a very low number of false alarms. The sensitivity of the model was 92.63% which indicates that the model performed better in the case of non-depressive samples compared to depressive samples. This model classified 1006 samples as depressive, whereas 906 samples were actually depressive (90.05%). From the ROC curve (Fig. 4(b)), we can see the model can classify both the non-depressive and depressive texts with a very high percentage, and AUC was also significant (0.94). This model provided very good performance in terms of all the metrics, but there is room for improvement in terms of recognising depressive posts.

4.2. Robustness of the proposed model

Robustness refers to the capability of the model to perform in the case of more noisy data or data from other sources. To prove the robustness, the proposed model was tested using several training–testing sets, real depressive Facebook posts, and a dataset of English depressive texts.

4.2.1. Performance of the proposed model in different training–testing set combination

3 different random training–testing combinations of the dataset were created to test the performance of the proposed model. In all cases, 12,000 samples were used for training, and 3031 samples were kept aside for testing. Table 5 describes the performance in different combinations of testing sets. In all the cases, the accuracy was in the range of 94.29% and 94.65%. The Sensitivity was over 91.50%, and the specificity was more than 94.5% in all the cases. So, different combinations of training–testing sets did not have a significant impact on the performance of the proposed model.

4.2.2. Performance of the proposed model in English depressive text detection

To prove the capability of detecting depressive texts of other languages, the proposed Attention-based BiLSTM-CNN hybrid model has been used in detecting depressive English social media texts. For this,

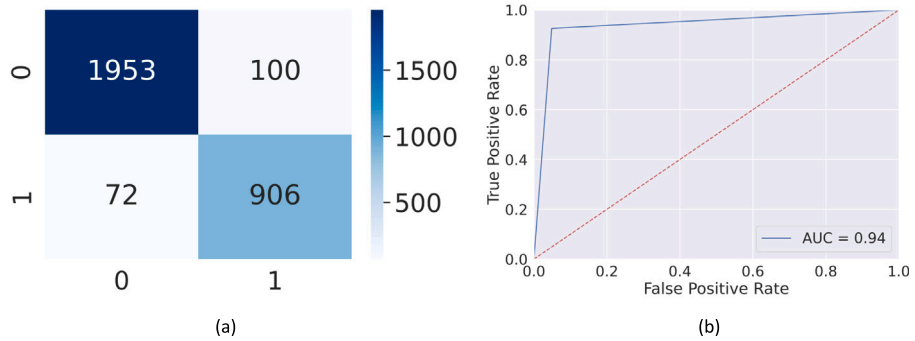

Fig. 4. (a) Confusion matrix and (b) ROC curve of Attention-based BiLSTM-CNN hybrid model using fastText embedding. The proposed model correctly classified 906 out of 978 depressive samples. The AUC was 0.94 in this model.

Table 5
Performance of the proposed model in different test sets.

Set#	Accuracy	Sensitivity	Specificity
Set1	0.9458	0.9151	0.9605
Set3	0.9429	0.9314	0.9483
Set4	0.9465	0.9304	0.9542
Considered	0.9433	0.9263	0.9512

Table 6
Performance in depressive English text detection.

Metrics	Attention-BiLSTM-CNN
Accuracy	0.9602
TN	1935
FP	44
FN	72
TP	869
Sensitivity	0.9234
Specificity	0.9776
P0	0.9641
P1	0.9518

a dataset was built using 10,000 non-depressive tweets from the Sentiment 140 dataset (Go, Bhayani, & Huang, 2009), and 4920 depressive tweets from the Depressive Tweets Processed dataset (P, 2021). The dataset was divided into two parts: train and test. 12,000 samples were used for training, and 2920 samples were kept aside for testing the performance of the model. Then, the texts went through a preprocessing stage, where punctuation and stop words were removed. The texts were lemmatised to get the base form of the words. Then, the texts were converted to 140-sized input tensors after performing one hot encoding and padding. Then, a pre-trained Word2Vec embedding was used to map the words into vectors, which was the input to the proposed model. The result obtained by the proposed Attention-based BiLSTM-CNN is described in Table 6.

The proposed attention-based BiLSTM-CNN model achieved an accuracy of 96.02% in English depressive text detection. This model acquired 92.35% sensitivity and 97.76% specificity in this task. This model correctly classified 869 out of 941 depressive samples, and 1935 out of 1979 non-depressive samples. It predicted 913 samples as depressive, whereas 869 samples (95.18%) were actually depressive. This model marked 2007 samples as non-depressive, and 1935 (96.41%) were true non-depressive. Fig. 5 shows the confusion matrix and ROC curve of the model. AUC score of this model was 0.95, which indicates this model, on average, performs well for 95% of the samples. The above result proves that the proposed model can be applied in other languages as well.

4.2.3. Performance of the proposed model in real suicidal texts

DL models sometimes provide good performance in the experimental dataset, but performance degrades when these models are used in real scenarios. To test this, some real Bangla suicidal Facebook statuses were collected from newspaper articles which were posted before suicidal incidents. There were, in total, 106 Bangla statuses of 15 people, and these statuses were labelled as depressive by an expert. These samples were not used during the training or testing phase of the proposed model nor included in the dataset. As depression is the leading cause of suicide (WHO, 2021a), and suicidal posts tend to be depressive (Pisani et al., 2022), it was a necessity to investigate the performance of the proposed method in this scenario. In this experiment of 106 samples, no non-depressive posts were considered, and so, all the labels of the suicidal samples considered in this experiment were one (1). All the preprocessing stages mentioned in the methodology section were followed in this case. The proposed Attention-based BiLSTM-CNN model with fastText embedding correctly classified **99 out of 106 (93.39%)** posts as depressive. So, the performance of the proposed model, in this case, matches the performance obtained during the experimental dataset ($sensitivity = 92.63\%$).

Table 7
Performance obtained after tweaking the proposed model.

Architecture	Accuracy	Sensitivity	Specificity
Adding a dense layer	0.9254	0.9110	0.9322
Reducing a dense layer	0.9343	0.9182	0.9420
Adding a LSTM layer	0.9214	0.9212	0.9215
Adding a convolutional layer	0.9343	0.8905	0.9551
Proposed combination	**0.9433**	**0.9263**	**0.9512**

4.3. Structure of the proposed model

The layers of the proposed model were selected by performing fine-tuning. Several structures have been tried out by tweaking the Convolutional, LSTM, and Dense layers of the proposed Attention-BiLSTM-CNN model. Table 7 describes the performances obtained during trying out several combinations of the proposed Attention-based BiLSTM-CNN architecture. From the table, it can be clearly said that adding or removing the LSTM, Convolutional or Dense layer could not improve the performance, but rather degraded it. So, the proposed architecture can be considered the optimal one.

4.4. Performance comparison with the Attention-LSTM-CNN models and impact of attention layer

To detect depressive social media posts, the Attention-based BiLSTM-CNN hybrid model has been proposed in this paper. It was found after a model searching phase, from which seven other models were shortlisted. These seven other models were also tried out in this paper. Table 3 describes the structure of these models. A combination of three dense layers and two dropouts which is similar to the proposed model, has been used in these models after the feature extraction

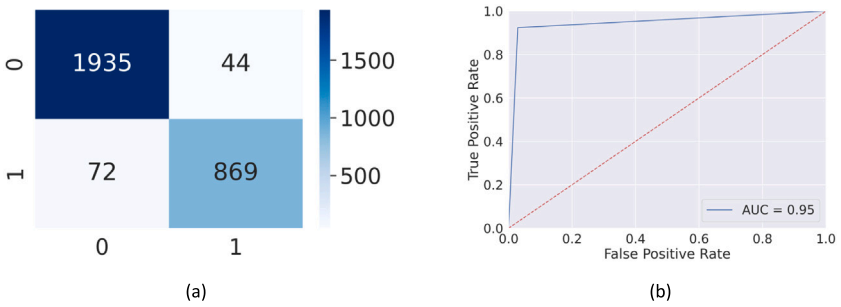

Fig. 5. (a) Confusion matrix and (b) ROC curve of Attention-based bidirectional LSTM-CNN approach, which were obtained during depressive English text detection. The proposed model correctly classified 869 out of 941 depressive posts. The AUC of the model was 0.95.

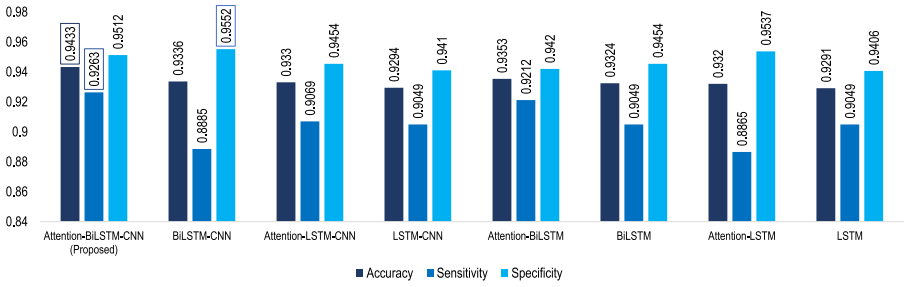

Fig. 6. Performance Comparison of the Attention, LSTM, CNN-based models. The proposed Attention-based BiLSTM-CNN model outperformed all the other models in terms of accuracy and sensitivity.

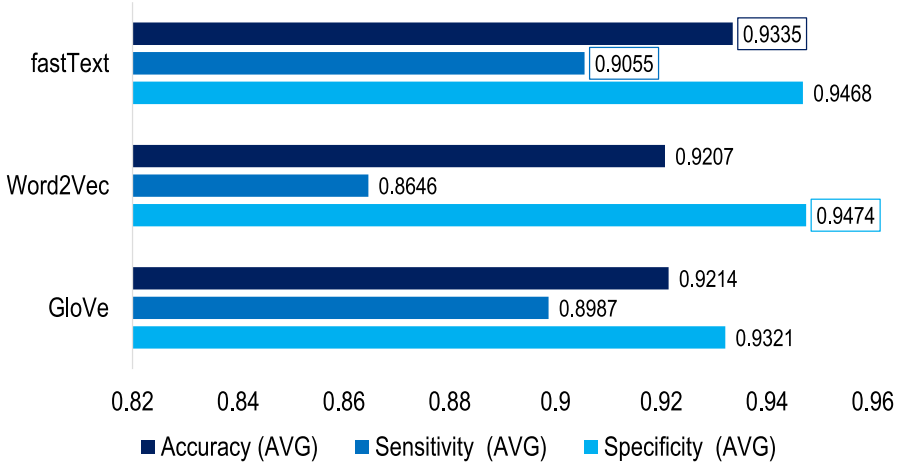

Fig. 7. Performance comparison of the word embedding techniques. This result is the average of the obtained results by 8 Attention, LSTM, CNN based models using these pre-trained word embeddings. These models performed the best while using fastText embedding.

layers. Fig. 6 illustrates the comparison of these models in terms of accuracy, sensitivity, and specificity in the best condition, where Table 8 shows the result in all three embeddings. The models performed better while using fastText embedding. The Proposed Attention-based BiLSTM-CNN model achieved the highest accuracy (94.33%) and sensitivity (92.63%) using fastText embedding. Attention-based BiLSTM model achieved 93.53% accuracy and 92.12% sensitivity using fastText embedding, which is the second highest among these models. The BiLSTM-CNN model achieved the highest specificity (95.52%), but it performed poorly in terms of sensitivity (88.85%). The proposed model achieved 92.44% and 92.67% accuracy, which were the best while using the word2vec and GloVe embedding. Comparing all the metrics, the Attention-based BiLSTM-CNN hybrid model was the best model to detect depressive Bangla social media posts.

From Fig. 6, it is evident that performance was boosted greatly after adopting the attention mechanism. The accuracy of the BiLSTM-CNN model was improved from 93.36% to 94.33% after adding an attention layer. Improvement of accuracy after adding an attention layer is also noticeable in LSTM-CNN (92.94% to 93.30%), BiLSTM (93.24% to 93.53%), and LSTM (92.91% to 93.20%) models. So, the attention mechanism helped the models greatly to achieve a reasonable detection accuracy and sensitivity.

Table 8
Performance of the selected models based while using different embeddings.

Embedding	Model	Accuracy	Sensitivity	Specificity
fastText	Attention-BiLSTM-CNN (Proposed)	**0.94325**	**0.9263**	0.9512
	BiLSTM-CNN	0.9336	0.8885	**0.9552**
	Attention-LSTM-CNN	0.933	0.9069	0.9454
	LSTM-CNN	0.9294	0.9049	0.941
	Attention-BiLSTM	0.9353	0.9212	0.942
	BiLSTM	0.9324	0.9049	0.9454
	Attention-LSTM	0.932	0.8865	0.9537
	LSTM	0.9291	0.9049	0.9406
Word2Vec	Attention-BiLSTM-CNN (Proposed)	**0.9244**	**0.9079**	0.9322
	BiLSTM-CNN	0.9218	0.9018	0.9313
	Attention-LSTM-CNN	0.9182	0.8425	0.9542
	LSTM-CNN	0.9172	0.8139	**0.9663**
	Attention-BiLSTM	0.9241	0.8538	0.9576
	BiLSTM	0.9231	0.8916	0.9381
	Attention-LSTM	0.9188	0.8558	0.9488
	LSTM	0.9182	0.8496	0.9508
GloVe	Attention-BiLSTM-CNN (Proposed)	**0.9267**	**0.9182**	0.9308
	BiLSTM-CNN	0.9221	0.9182	0.924
	Attention-LSTM-CNN	0.9221	0.9182	0.924
	LSTM-CNN	0.9201	0.8251	**0.9654**
	Attention-BiLSTM	0.9241	0.8946	0.9381
	BiLSTM	0.9211	0.9202	0.9215
	Attention-LSTM	0.9161	0.8895	0.9288
	LSTM	0.9185	0.9059	0.9245

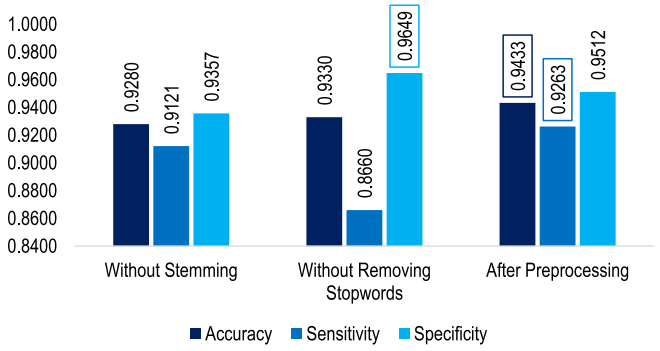

Fig. 8. Impact of the preprocessing in Bangla depressive text detection. The proposed model performed the best while both stemming and stop word removal was performed.

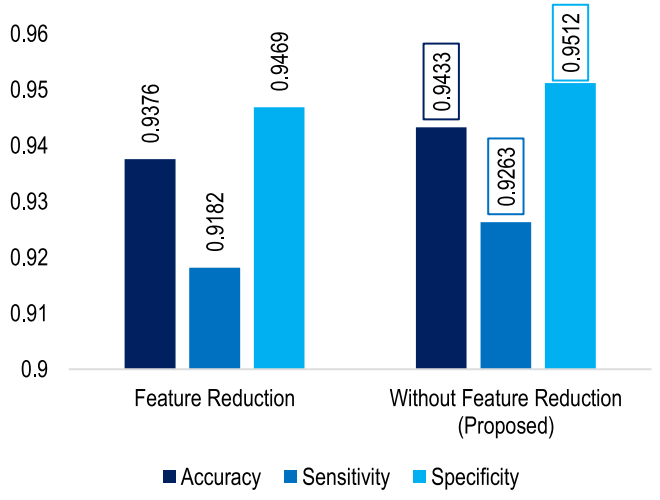

Fig. 9. Impact of feature reduction in Bangla depressive text detection. The feature reduction process could not improve the performance of the proposed model.

4.5. Discussion on pre-trained embeddings

In this work, 3 embeddings were used. To evaluate the impact of embedding, the average of the obtained results by 8 attention, LSTM, and CNN-based models were considered, which is illustrated in Fig. 7. The average accuracy obtained using fastText embedding was 93.35%, where the average sensitivity was 0.9055, and specificity was 0.9468. The average accuracy of Word2Vec and GloVe were 92.07% and 92.14%, respectively. The average sensitivity and specificity obtained by using Word2Vec were 86.46% and 89.87%. GloVe performed better than Word2Vec in terms of sensitivity (89.87%). The fastText embedding obtained the best average accuracy and sensitivity among these 3 embeddings. Word2Vec obtained the highest specificity, but the improvement over fastText was not significant enough. Clearly, the models performed the best while using fastText embedding.

Bangla is a very complex language with a very large number of characters and symbols. Even native speakers sometimes find it difficult to write proper spelling, which is more prevalent in social media. fastText breaks the words into characters, which empowers it to perform better in terms of new or misspelt words that might be out of the dictionary. It performs better in complex languages compared to other embeddings. These could be the possible reasons behind achieving very high performance using fastText embedding.

4.6. Discussion on preprocessing

Textual data contains a lot of symbols and words which does not contain any significant importance. To reduce the impact of the curse of dimensionality, the texts went through a stop word removal and stemming phases. These texts were then converted into input tensors after performing one hot encoding and padding sequence. The proposed model took this vector as input and predicted it as depressive or non-depressive. To check the impact of preprocessing, the proposed model was trained and tested without these two preprocessing phases. At first, stemming was not performed on the text. Without stemming, the proposed model classified texts at 92.80% accuracy, 91.21% sensitivity, and 93.57% specificity. Then, the stemming was performed, but stop words were not removed from the texts. The model achieved 93.3% accuracy, 86.6% sensitivity, and 96.49% specificity in this case. In both of the cases, the performance of the proposed model degraded drastically, which is illustrated in Fig. 8. So, the preprocessing stages had a huge impact on the significant performance of the proposed model.

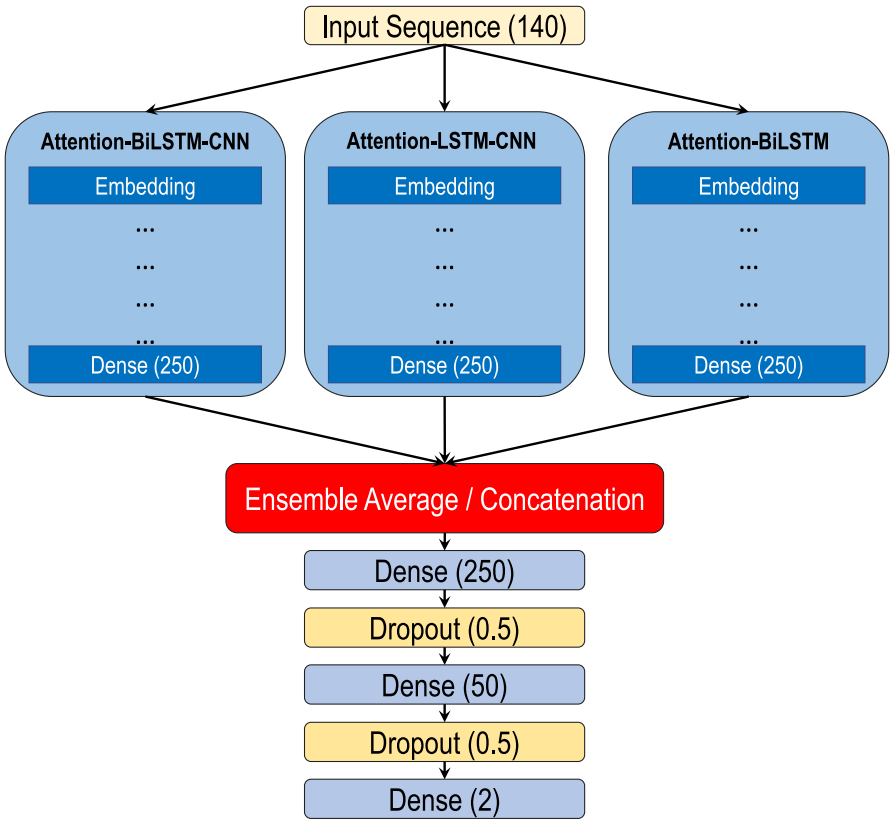

Fig. 10. Ensemble process. Feature extracted by the 3 models were concatenated or averaged, which was the input to the dense layers that provided prediction.

4.7. Discussion on feature reduction

In NLP, sometimes reducing less significant words can lead to performance improvement (Sharmin & Chakma, 2021). So, in this paper, a word reduction technique using F-test has been carried out. After performing all the preprocessing steps, the dataset contained more than 27,000 words. To find the most important words, the TF-IDF of the dataset was calculated at first. Then, F-test was performed on this feature set. By considering the significant scores, 14,785 words were kept in the dataset. Then, the preprocessed texts were given input to the proposed model with fastText embedding. This feature reduction method obtained 93.76% accuracy with 91.82% sensitivity and 94.69% specificity. The proposed model with all the words performed a lot better than the feature reduction method in terms of accuracy, sensitivity and specificity, which is depicted in Fig. 9. Therefore, it can be said that the feature reduction technique could not improve the proposed model's performance in the depressive Bangla social media text detection task.

4.8. Performance comparison with ensemble approaches

In this work, 8 different LSTM, CNN, and Attention-based architectures were developed. Among these models, the attention-based BiLSTM-CNN hybrid model (proposed), Attention-based LSTM-CNN hybrid model, and Attention-based BiLSTM model performed better than the other models. So, these 3 models were selected for the ensemble. The input was given to these 3 models individually, and the output was taken from the first dense layer of these 3 models, which provided a feature vector of length of 250. To ensemble these vectors, two ways were adopted.

- Concatenation of three feature vectors.
- Taking the average of three feature vectors.

Table 9
Performance of the ensemble approach.

Metrics	Ensemble	
	Concatenation	Average
Accuracy	0.9452	0.9459
TN	1989	1979
FP	64	82
FN	102	82
TP	876	896
Sensitivity	0.8957	0.9162
Specificity	0.9688	0.9601
P_0	0.9512	0.9601
P_1	0.9312	0.9162

After performing an ensemble of the feature vectors, the new feature vector was provided as input to a 3-dense layer and 2-dropout layer network. This network provided prediction as output based on the input feature set. The fastText embedding was used in both of the cases as the proposed DL models performed the best while using fastText embedding. Fig. 10 illustrates the ensemble process. The detailed result of the ensemble approach is provided in Table 9.

By concatenating the feature vectors of the 3 models, the ensemble model achieved 94.52% accuracy, 89.57% sensitivity, and 96.88% specificity. It correctly classified 876 out of 978 depressive and 1941 out of 2053 non-depressive samples. 93.12% of the predicted depressive samples were actually depressive, whereas 95.12% of the predicted non-depressive samples were true non-depressive.

By taking the average of the feature vectors of the 3 models, the ensemble model achieved 94.59% accuracy, 91.62% sensitivity, and 96.01% specificity. It correctly classified 896 out of 978 depressive, and 1979 out of 2053 non-depressive samples. 91.62% of the predicted depressive samples were actually depressive, whereas 96.01% of the predicted non-depressive samples were true non-depressive.

Table 10
Size, training time, and parameters of the ensemble and the proposed model.

Model	Size (MB)	Training time (s/epoch)	Parameters
Concatenating ensemble	53	379	4,681,805
Average ensemble	50.8	389	4,556,805
Proposed Attention-BiLSTM-CNN	35.7	158	4,014,153

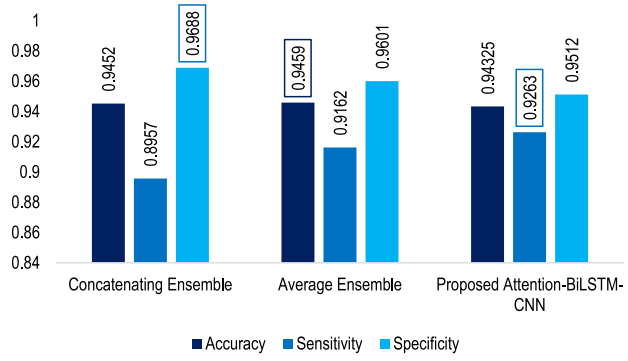

Fig. 11. Performance comparison between ensemble approach and the proposed model. Ensemble approaches achieved better accuracy and specificity, but the proposed model was better considering the overall performance.

The proposed Attention-based BiLSTM-CNN model achieved 94.33% accuracy, 92.63% sensitivity, and 95.12% specificity. Both the ensemble models performed better than the proposed model in terms of accuracy and specificity. But for a .25% accuracy, there was a trade-off of around 1% sensitivity, which is crucial in depressive social media text detection tasks. Fig. 11 depicts the performance comparison between the proposed models and the ensemble approaches.

From Table 10, it can be seen that the proposed model was 15 MB smaller in size than the ensemble models. It is also clear that the proposed model contained less number of parameters and almost took three times lesser time to train than the ensemble approaches, which is a necessary factor as the depressive text detection model needs to deal with a huge number of samples at a shorter time. After considering model size, training time, parameter, and performance trade-off, it can be said that the Attention-based BiLSTM-CNN is a better choice than the ensemble approach in Bangla depressive text detection.

4.9. Performance comparison with transformer models

Pre-trained transformer models can achieve a very good performance in classification tasks. We have tried four different pre-trained transformers in Bangla depressive text detection. Fig. 12 depicts the comparison between transformers and the proposed model. BanglaBERT performed the best (94.16% accuracy, 91.82% sensitivity, and 95.27% specificity) among these models. Bangla-Electra also achieved more than 93% accuracy. The proposed attention-based BiLSTM-CNN outperformed these models with 94.33% accuracy, 92.63% sensitivity, and 95.12% specificity. These pre-trained models are too big and require much time to train and test. Considering all these factors, the proposed model can be a better choice than the transformers.

4.10. Performance comparison with classical ML models

ML models such as Support vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), Naive Bayes (NB), K Nearest Neighbor (KNN) etc. require comparatively less time and resources to train and test. Several classical ML algorithms have been tried with TF-IDF, Word2Vec, fastText, and GloVe embedding. Detail result is shown in Table 11. Fig. 13 illustrates the performance comparison between the best performing classical ML approaches and the proposed model. SVM achieved the best accuracy among the ML models by using the TF-IDF feature. The sensitivity obtained by this approach was not up to the mark (82.5%). LR achieved better specificity than the proposed model, but its accuracy was 91.19%, and sensitivity was 78.63%, which is very low compared to the proposed model. By achieving better accuracy (94.325%), sensitivity (92.63%), and significant specificity (95.12%), the proposed attention-based BiLSTM-CNN model outperformed all the classical ML approaches.

Table 11
Performance of the classical ML models using different embeddings and features.

Embedding	Model	Accuracy	Sensitivity	Specificity
fastText	DT	0.7146	0.5726	0.7822
	KNN	0.6796	0.8803	0.5840
	LR	0.8835	0.8128	0.9171
	NB	0.3375	0.9867	0.0282
	RF	0.7815	0.3466	0.9887
	SVM	0.9155	0.8425	0.9503
Word2Vec	DT	0.6981	0.5204	0.7827
	KNN	0.7387	0.4754	0.8641
	LR	0.8143	0.6973	0.8699
	NB	0.3559	0.9519	0.0721
	RF	0.7516	0.2781	0.9771
	SVM	0.8139	0.5951	0.9181
GloVe	DT	0.6760	0.4734	0.7725
	KNN	0.7004	0.6870	0.9537
	LR	0.8050	0.7218	0.8446
	NB	0.3431	0.9550	0.0516
	RF	0.7347	0.1912	0.9937
	SVM	0.8350	0.6257	0.9347
TF-IDF	DT	0.8426	0.7842	0.8704
	KNN	0.6592	0.1247	0.9138
	LR	0.9119	0.7883	0.9707
	NB	0.8007	0.7781	0.8115
	RF	0.8940	0.8292	0.9249
	SVM	0.9221	0.8251	0.9683

4.11. Performance comparison with the existing architectures

There is a limited number of existing DL architectures for detecting depressive Bangla social media texts. To compare the performance of these models with the proposed model, three existing architectures were recreated, trained and tested using this dataset (see Table 12).

Uddin et al. (2019b) developed a GRU-based model for Bangla depressive post detection. They used 5 GRU layers with 512 neurons. This model was recreated and run for 25 epochs with Word2Vec embedding. This model achieved 91.22% accuracy, where sensitivity was 86.7%, and specificity was 93.37%. This model correctly classified 848 out of 978 depressive posts and 1917 out of 2053 non-depressive posts.

Uddin et al. (2019a) developed a five-layered LSTM architecture with 128 neurons. They used a learning rate of 0.0001, batch size was 25, and the model was trained for 25 epochs. This model was re-developed, trained and tested on our dataset with the same hyperparameters. This architecture achieved 91.15% accuracy with 85.88% sensitivity and 93.66% specificity. This model successfully classified 840 out of 978 depressive posts and 1923 out of 2053 non-depressive posts.

Mumu et al. (2021) took an LSTM-CNN hybrid approach where one Convolutional layer, one max pooling layer, two LSTM layers with 512 neurons and tanh activation function, and a single dense layer with Softmax activation function were used. They used Word2Vec

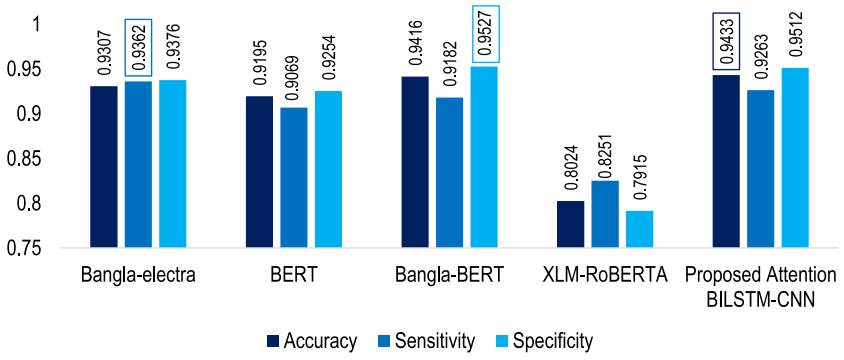

Fig. 12. Performance comparison between transformers and the proposed model. The proposed model performed better than transformer-based models.

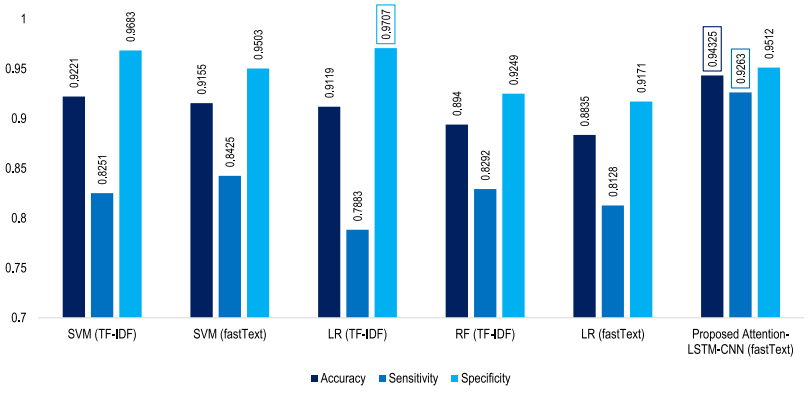

Fig. 13. Performance comparison between best performing classical ML models and the proposed model. The proposed model achieved better accuracy and sensitivity than all the classical ML models.

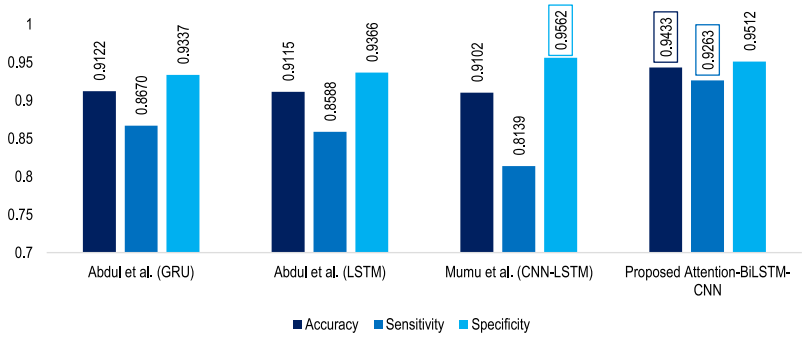

Fig. 14. Performance comparison between the existing architectures and the proposed model. The proposed model outperformed the existing models in terms of accuracy and sensitivity.

Table 12
Performance of existing models.

Metrics	Uddin et al. (2019b) (GRU)	Uddin et al. (2019a) (LSTM)	Mumu et al. (2021)
Accuracy	0.9122	0.9115	0.9102
TN	1917	1923	1963
FP	136	130	90
FN	130	138	182
TP	848	840	796
Sensitivity	0.867	0.8588	0.8139
Specificity	0.9337	0.9366	0.9562
P_0	0.9365	0.933	0.9152
P_1	0.8618	0.8659	0.8984

Fig. 15. Word cloud generated from training depressive and non-depressive samples, true positive testing samples, true negative testing samples, false positive testing samples, and false negative testing samples.

embedding. This model was trained for 25 epochs, and the learning rate was 0.01. This model achieved 91.02% accuracy, 81.39% sensitivity, and 95.62% specificity in this dataset. It correctly classified 796 out of 978 depressive and 1963 out of 2053 non-depressive samples.

Fig. 14 illustrates the comparison between the proposed model and existing architecture. The proposed model outperformed all the existing architectures in terms of sensitivity and accuracy.

5. Explainability in the feature level

To explain the reasons behind classifying the samples as depressive or non-depressive, similarities between training and testing samples have been utilised in this paper. Fig. 15 illustrates the word cloud generated from the training and testing samples. Fig. 16 provides the English meaning of some of the most used Bangla words mentioned in Word Cloud. In the training depressive samples, words that mean "good", "human", "speak", "depression", life", "love", "suicide", "death" etc. are found multiple times after stemming and stop words removal. On the other hand, words that mean "human", "Bangladesh", "thanks", "money", "Allah", "fear", "country" etc. words have been found in a big chunk of the non-depressive training samples.

True positive samples mean the actual depressive samples, which were predicted as depressive. From Fig. 15, it is seen that there is a similarity between the most used words in training depressive samples and true positive samples. Words used in true positive samples are quite different from the most frequent words of non-depressive training samples. Fig. 17 shows an example of true positive sample. In this example, it is seen that words used in these samples are more frequent in the depressive samples rather than in non-depressive samples, which helped it to be classified as depressive.

In the true negative samples, words that mean "human", "Bangladesh", "thanks", and "money" are more frequently found, which were mostly observed in non-depressive training samples. The most frequent words used in true negative samples are not present in the word cloud of the training depressive samples, which indicates the absence of these words in a significant number. There is also a similarity in the word clouds of non-depressive samples and true negative samples. Fig. 18 shows an example of true negative sample. Words used in this sample are found in both depressive and non-depressive examples in almost similar percentages. However, the number of training samples of non-depressive samples was larger in the dataset. Therefore, the model found it more similar to the non-depressive sample, and predicted it as non-depressive, which was a correct prediction.

False positive samples denote the actual non-depressive samples, which were predicted as depressive. From the word clouds of false positive samples and training depressive samples, it can be noticed that there are some similarities in the usage of words. It is not seen in between the word clouds of false positive samples and training non-depressive samples. From the word clouds of the false negative samples and the training depressive samples, it can be seen that there is very less similarity between them. Rather, word clouds of false negative and training non-depressive samples are quite similar. Bangla is a very diverse language. Similar words can be used in multiple meanings. DL models need a large training set to train and discover a perfect pattern. In this case, there were only 15,031 samples which were insufficient.

Bangla	English	Bangla	English
মানুষ	human	ভালো	good
বাংলাদেশ	Bangladesh	কথা	speak
ধন্যবাদ	thanks	ডিপ্রেশন	depression
টাকা	money	জীবন	life
আল্লাহ	Allah/ God	ভালোবাসা	love
ভয়	fear	আত্মহত্যা	suicide
দেশ	country	মৃত্যু	death

Fig. 16. Some of the most commonly used Bangla words in Wordcloud and their corresponding English meaning.

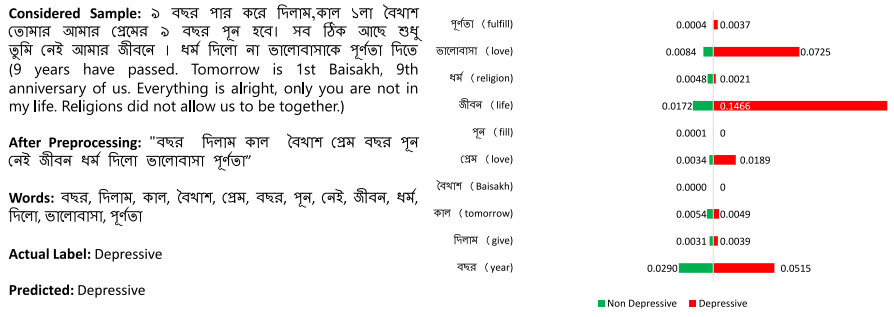

Fig. 17. Example of a true positive sample. Here, words used in these samples are found in a higher percentage in the depressed training samples.

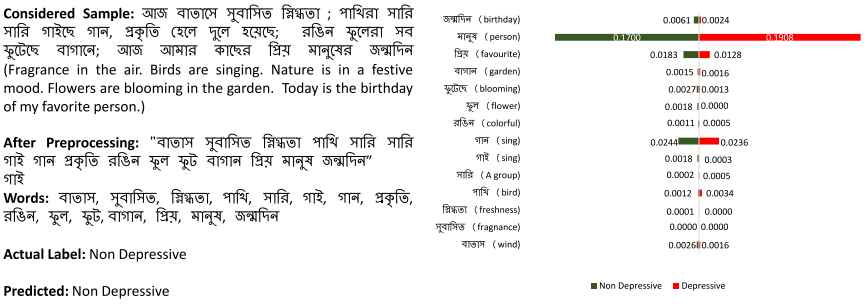

Fig. 18. Example of a true negative sample. Here, words used in these samples are found in a higher percentage in the non-depressed training samples.

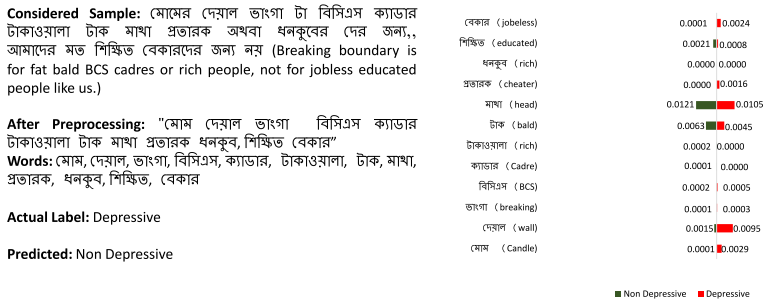

Fig. 19. Example of a false negative sample. Here, words used in these samples are found in almost equal percentages in the non-depressed and depressed training samples.

So, to get very high accuracy and discover more confusing patterns, a very large training set is needed. In the example of false negative (Fig. 19), some of the words used in this example were mostly found in non-depressive samples, where few have a higher share in depressive samples. So, the model gets confused and denotes it as non-depressive, where it was actually a depressive sample. The opposite scenario can be noticed in the example of the false positive sample (Fig. 20). Here, the example is actually labelled as non-depressive. But, words used in this sample are found in the depressive sample in higher numbers than in non-depressive samples. That is why the proposed model could not classify it as the non-depressive sample.

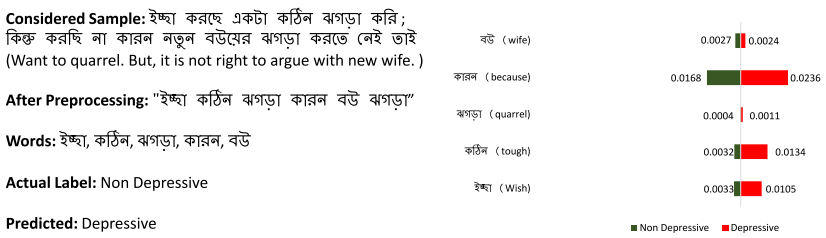

Fig. 20. Example of a false positive sample. Here, some words used in these samples are found in a higher percentage in the depressed training samples.

6. Conclusion

Depression is one of the most common mental health problems that can appear at any age. Suicide caused by depression is the fourth leading cause of death for young adults. Social media has become the platform for sharing the thoughts of people. People share their daily events, emotions and feelings through it. As depression is directly related to emotion and feelings, there is always a chance to find meaningful relationships between the textual contents of social media and depression. Due to the huge increase in social media usage, it is nearly impossible to manually assess and analyse the social media contents and act accordingly. So, a fully automated system is necessary to analyse ever-growing social media content to detect and prevent depression and depression-related incidents. To solve this problem, an attention-based BiLSTM-CNN model has been proposed to detect depressive social media text in this paper, providing a remarkable performance. This model is mainly trained and tested in Bangla, one of the world's most popular languages. A large Bangla depressive text detection dataset was also created in this work which will surely assist in future research. Several related DL and ML models were also created to compare the performance of the proposed model. The proposed model was also found as better performing than an ensemble approach and existing architectures. The proposed model also proved its efficiency in classifying depressive texts in English by acquiring over 96% accuracy. Finally, a feature-level explanation was provided to clarify the reasons behind the prediction. In the future, the researchers should create a more diverse dataset to deal with different forms of texts commonly seen on social media. They should also consider the reliability factor and concentrate on building models for low-resource languages like Bangla.

CRediT authorship contribution statement

Tapotosh Ghosh: Original idea, Methodology, Results interpreted, Writing – original draft, Writing – review & editing. **Md. Hasan Al Banna:** Methodology, Results interpreted, Writing – original draft, Writing – review & editing. **Md. Jaber Al Nahian:** Methodology, Writing – original draft, Writing – review & editing. **Mohammed Nasir Uddin:** Results interpreted, Writing – original draft, Writing – review & editing. **M. Shamim Kaiser:** Original idea, Results interpreted, Writing – original draft, Writing – review & editing. **Mufti Mahmud:** Original idea, Results interpreted, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This research received funding from the ICT division of the Government of the People's Republic of Bangladesh for the 2020–21 financial year (tracking no: 20FS13595). M Mahmud has been supported by the QR fund of Nottingham Trent University.

References

Addis, M. E., & Jacobson, N. S. (1996). Reasons for depression and the process and outcome of cognitive–behavioral psychotherapies. *Journal of Consulting and Clinical Psychology*, *64*(1417).

Ahmad, H., Asghar, D. M., Alotaibi, F., & Hameed, I. (2020). Applying deep learning technique for depression classification in social media text. *Journal of Medical Imaging and Health Informatics*, *10*(6), 2446–2451. http://dx.doi.org/10.1166/jmihi.2020.3169.

Al Nahian, M. J., Ghosh, T., Uddin, M. N., Islam, M. M., Mahmud, M., & Kaiser, M. S. (2020). Towards artificial intelligence driven emotion aware fall monitoring framework suitable for elderly people with neurological disorder. In *International conference on brain informatics* (pp. 275–286). Springer, http://dx.doi.org/10.1007/978-3-030-59277-6_25.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Berryman, C., Ferguson, C. J., & Negy, C. (2018). Social media use and mental health among young adults. *Psychiatric Quarterly*, *89*, 307–314.

Billah, M., & Hassan, E. (2019). Depression detection from Bangla facebook status using machine learning approach. *International Journal of Computer Applications*, *975*, 8887.

BNLP (2021a). Bengali natural language processing(BNLP) – BNLP latest documentation. https://bnlp.readthedocs.io/en/latest/#word-embedding. (Accessed on 02 July 2021).

BNLP (2021b). Index of /bnwiki/latest/. https://dumps.wikimedia.org/bnwiki/latest/. (Accessed on 02 July 2021).

Chiu, C. Y., Lane, H. Y., Koh, J. L., & Chen, A. L. (2021). Multimodal depression detection on instagram considering time interval of posts. *Journal of Intelligent Information Systems*, *56*, 25–47.

Cong, Q., Feng, Z., Li, F., Xiang, Y., Rao, G., & Tao, C. (2018). Xa-bilstm: A deep learning approach for depression detection in imbalanced data. In *2018 IEEE international conference on bioinformatics and biomedicine* (pp. 1624–1627). IEEE.

Coyne, S. M., Rogers, A. A., Zurcher, J. D., Stockdale, L., & Booth, M. (2020). Does time spent using social media impact mental health?: An eight year longitudinal study. *Computers in Human Behavior*, *104*, Article 106160.

datareportal (2021). Global social media stats - datareportal – Global digital insights. URL: https://datareportal.com/social-media-users.

Dean, B. (2021). How many people use social media in 2021? (65+ statistics). https://backlinko.com/social-media-users. (Accessed on 29 June 2021).

Deshpande, M., & Rao, V. (2017). Depression detection using emotion artificial intelligence. In *2017 international conference on intelligent sustainable systems* (pp. 858–862). IEEE.

Escobar-Viera, C. G., Whitfield, D. L., Wessel, C. B., Shensa, A., Sidani, J. E., Brown, A. L., et al. (2018). For better or for worse? a systematic review of the evidence on social media use and depression among lesbian, gay, and bisexual minorities. *JMIR Mental Health*, *5*, Article e10496.

Ghannay, S., Favre, B., Esteve, Y., & Camelin, N. (2016). Word embedding evaluation and combination. In *Proceedings of the tenth international conference on language resources and evaluation* (pp. 300–305).

Ghosh, T., Al Banna, M. H., Al Nahian, M. J., Taher, K. A., Kaiser, M. S., & Mahmud, M. (2021). A hybrid deep learning model to predict the impact of COVID-19 on mental health form social media big data. Preprints.

Go, A., Bhayani, R., & Huang, L. (2009). *Twitter sentiment classification using distant supervision: CS224N project report, Stanford, 1*, 2009.

Hannah Ritchie, M. R. (2017). Mental health - Our world in data. https://ourworldindata.org/mental-health#depression. (Accessed on 29 June 2021).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Islam, M. R., Kamal, A. R. M., Sultana, N., Islam, R., Moni, M. A., et al. (2018). Detecting depression using k-nearest neighbors (KNN) classification technique. In *2018 international conference on computer, communication, chemical, material and electronic engineering* (pp. 1–4). IEEE.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2016). Fasttext.zip: Compressing text classification models. arXiv preprint arXiv:1612.03651.

Kaium, A. (2021). bnltk.pypi. https://pypi.org/project/bnltk/. (Accessed on 03 July 2021).

Kanter, J. W., Busch, A. M., Weeks, C. E., & Landes, S. J. (2008). The nature of clinical depression: Symptoms, syndromes, and behavior analysis. *The Behavior Analyst*.

Lai, S., Liu, K., He, S., & Zhao, J. (2016). How to generate a good word embedding. *IEEE Intelligent Systems*, *31*, 5–14.

Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. *Advances in Neural Information Processing Systems*, *27*, 2177–2185.

Liao, M., Shi, B., Bai, X., Wang, X., & Liu, W. (2017). Textboxes: A fast text detector with a single deep neural network. In *Thirty-first AAAI conference on artificial intelligence*.

Martinsen, E. W. (2008). Physical activity in the prevention and treatment of anxiety and depression. *Nordic Journal of Psychiatry*, *62*, 25–29.

Mumu, T. F., Munni, I. J., & Das, A. K. (2021). Depressed people detection from Bangla social media status using LSTM and CNN approach. *Journal of Engineering Advancements*, *2*, 41–47.

O'Reilly, M., Dogra, N., Hughes, J., Reilly, P., George, R., & Whiteman, N. (2019). Potential of social media in promoting mental health in adolescents. *Health Promotion International*, *34*, 981–991.

O'Reilly, M., Dogra, N., Whiteman, N., Hughes, J., Eruyar, S., & Reilly, P. (2018). Is social media bad for mental health and wellbeing? exploring the perspectives of adolescents. *Clinical Child Psychology and Psychiatry*, *23*, 601–613.

P, I. (2021). Depression detection by using tweeter post. https://github.com/eddieir/Depression_detection_using_Twitter_post/blob/master/depressive_tweets_processed.csv. (Accessed on 06 July 2021).

Peng, Z., Hu, Q., & Dang, J. (2019). Multi-kernel svm based depression recognition using social media data. *International Journal of Machine Learning and Cybernetics*, *10*, 43–57.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1532–1543).

Pisani, A. R., Gould, M. S., Gallo, C., Ertefaie, A., Kelberman, C., Harrington, D., et al. (2022). Individuals who text crisis text line: Key characteristics and opportunities for suicide prevention. *Suicide and Life-Threatening Behavior*, http://dx.doi.org/10.1111/sltb.12872.

Rahman, N. I. A., Ismail, S., Binti, T. N. A., Seman, T., Binti, N. F. A., Mat, S. A. B., et al. (2013). Stress among preclinical medical students of university Sultan Zainal Abidin. *Journal of Applied Pharmaceutical Science*, *3*(76).

Sarker, S. (2021). Github - sagorbrur/glove-Bengali: Bengali glove pretrained word vector. https://github.com/sagorbrur/GloVe-Bengali. (Accessed on 03 July 2021).

Sharmin, S., & Chakma, D. (2021). Attention-based convolutional neural network for Bangla sentiment analysis. *AI & Society*, *36*, 381–396.

Spencer, L. J., Degu, A., Kalkidan, H. A., & Solomon, M. A. (2018). Global, regional, and national incidence, prevalence, and years lived with disability for 354 diseases and injuries for 195 countries and territories, 1990–2017: A systematic analysis for the global burden of disease study 2017. *The Lancet*, *392*, 1789–1858.

StatCounter (2021). Social media stats worldwide | statcounter global stats. https://gs.statcounter.com/social-media-stats. (Accessed on 29 June 2021).

Suman, S. K., Shalu, H., Agrawal, L. A., Agrawal, A., & Kadiwala, J. (2020). A novel sentiment analysis engine for preliminary depression status estimation on social media. arXiv preprint arXiv:2011.14280.

Tadesse, M. M., Lin, H., Xu, B., & Yang, L. (2019). Detection of depression-related posts in reddit social media forum. *IEEE Access*, *7*, 44883–44893.

Tipton, C. M. (2014). The history of "exercise is medicine" in ancient civilizations. *Advances in Physiology Education*, *38*, 109–117.

Trotzek, M., Koitka, S., & Friedrich, C. M. (2018). Utilizing neural networks and linguistic metadata for early detection of depression indications in text sequences. *IEEE Transactions on Knowledge and Data Engineering*, *32*, 588–601.

Uddin, A. H., Bapery, D., & Arif, A. S. M. (2019a). Depression analysis from social media data in Bangla language using long short term memory (LSTM) recurrent neural network technique. In *2019 international conference on computer, communication, chemical, materials and electronic engineering* (pp. 1–4). IEEE.

Uddin, A. H., Bapery, D., & Arif, A. S. M. (2019b). Depression analysis of Bangla social media data using gated recurrent neural network. In *2019 1st international conference on advances in science, engineering and robotics technology* (pp. 1–6). IEEE..

Venkataraman, D., & Parameswaran, N. S. (2018). Extraction of facial features for depression detection among students. *International Journal of Pure Appllied Mathematics*, *118*, 455–463.

Wang, P. S., Aguilar-Gaxiola, S., Alonso, J., Angermeyer, M. C., Borges, G., Bromet, E. J., et al. (2007). Use of mental health services for anxiety, mood, and substance disorders in 17 countries in the who world mental health surveys. *The Lancet*, *370*, 841–850.

WHO (2014). *Preventing suicide: A global imperative*. World Health Organization.

WHO (2021a). Depression. URL: https://www.who.int/news-room/fact-sheets/detail/depression.

WHO (2021b). Suicide worldwide in 2019: Global health estimates.

Wu, S., & Manber, U. (1992). Fast text searching: Allowing errors. *Communications of the ACM*, *35*, 83–91.

Zogan, H., Razzak, I., Jameel, S., & Xu, G. (2021). Depressionnet: A novel summarization boosted deep framework for depression detection on social media. arXiv preprint arXiv:2105.10878.