SciEn
Journal of Engineering Advancements Vol. 04(02) 2023, pp 49-56 https://doi.org/10.38032/jea.2023.02.003

# Detecting Level of Depression from Social Media Posts for the Low-resource Bengali Language

*Md. Nesarul Hoque*[1,*] *and Umme Salma*[2]

[1] Department of Computer Science and Engineering, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj-8100, Bangladesh,
[2] Department of Computer Science & Engineering, Bangladesh University, Dhaka-1207, Bangladesh

Received: May 17, 2023, Revised: June 19, 2023, Accepted: June 19, 2023, Available Online: June 28, 2023

**ABSTRACT**

Depression is a mental illness that suffers people in their thoughts and daily activities. In extreme cases, sometimes it leads to self-destruction or commit to suicide. Besides an individual, depression harms the victim's family, society, and working environment. Therefore, before physiological treatment, it is essential to identify depressed people first. As various social media platforms like Facebook overwhelm our everyday life, depressed people share their personal feelings and opinions through these platforms by sending posts or comments. We have detected many research work that experiment on those text messages in English and other highly-resourced languages. Limited works we have identified in low-resource languages like Bengali. In addition, most of these works deal with a binary classification problem. We classify the Bengali depression text into four classes: non-depressive, mild, moderate, and severe in this investigation. At first, we developed a depression dataset of 2,598 entries. Then, we apply pre-processing tasks, feature selection techniques, and three types of machine learning (ML) models: classical ML, deep-learning (DL), and transformer-based pre-trained models. The XLM-RoBERTa-based pre-trained model outperforms with 61.11% F1-score and 60.89% accuracy the existing works for the four levels of the depression-class classification problem. Our proposed machine learning-based automatic detection system can recognize the various stages of depression, from low to high. It may assist the psychologist or others in providing level-wise counseling to depressed people to return to their ordinary life.

Keywords: Depression, Machine learning, Multi-class Classification, Low-resource Language, Bengali.



## 1 Introduction

The mood, thoughts, behavior, and physical health of a person all affect the mental health illness known as depression. It can affect a person's capacity to function in their way of life and range in intensity from minor to severe. Aside from causing emotional pain, it can also have negative effects on one's physical health. It frequently carries a greater mortality rate as well as a higher chance of developing chronic diseases like obesity, diabetes, and cardiovascular disease. Depression can arise from several sources, including hereditary, abusive, environmental, and psychological ones. The World Health Organization (WHO) predicted that by 2023, there would be over 280 million depressed people of all ages in the world. Here, females are more likely than males to struggle with depression.

Despite having varying effects in various nations and regions, depression was the second-leading cause of disability worldwide[1]. As an illustration, Afghanistan had the highest rate of serious depression. Due to decreased productivity and higher healthcare expenditures, depression has a huge effect on the world economy[2]. Depression is a significant factor in the nearly 800k suicide deaths that occur each year globally and for every suicide, there are over 20 attempts[3].

In Bangladesh, around 7 million people were thought to have a serious depressive illness [1]. Another study conducted among school-going students in this country reported a prevalence of depression and anxiety [2].

The prevalence rate for anxiety and depression from moderate to severe levels was 26.5% and 18.1%, respectively. Due to missed workdays and decreased production, depression has a substantial influence on a country's economy and productivity [3]. In 2016, the economic cost of depression was estimated by research in the Asian Journal of Psychiatry to be around 4.4% of GDP (gross domestic product). Depression is a significant risk factor for suicide [4]. Suicide rates were 7.3 (95% Confidence Interval, CI 5.6-9.5) per 100,000 per year, with the greatest rate seen in the age group of 60 years and above. In comparison to urban populations, the rate of suicide was found to be 17 times higher (95% CI 5.36-54.64) in rural areas.

In every depression-related suicide, we do not just lose a person. It severely hurts the victim's family and society from a financial point of view. It also interferes with the daily activities of the victim's surrounding people. Moreover, depressed individuals create an unstable work environment, which hinders the progress of an organization. It needs to counsel the depressed people according to their level of depression such that they come back to their regular life. Therefore, it is essential to identify depressed people with the severity level of depression.

[1] https://www.bbc.com/news/health-24818048
[2] https://www.who.int/redirect-pages/mega-menu/emergencies
[3] https://www.who.int/health-topics/suicide

*Corresponding Author Email Address: mnhshisir@gmail.com Published by: SciEn Publishing Group

Different social networking sites, such as Facebook, Twitter, and others, have become firmly integrated with human life in recent years because of the widespread usage of the internet. Depressed people often use these platforms to express their personal feelings and opinions. Therefore, we focus on Bengali text data gathered from popular social networking sites like Facebook. As a high-inflectional and low-resource language, detecting a Bengali depressive text is a more challenging job. We have identified several pieces of research on depressive texts in the Bengali language. Most of these studies focused on binary classification problems (depression or non-depression) [5]-[10]. Moreover, the authors developed datasets from various emotional points of view like anger, joy, fear, surprise, disgust, sadness, and others for multi-class classification problems [11],[12]. Our following research contributions address the above issues:

Developing a depressive dataset labeling with four classes: non-depressive, mild, moderate, and severe, by verifying a psychology expert.

Experimenting with pre-processing, feature selection, and various machine learning (ML) models, and show the comparative analysis.

Building a better classification system with a transformer-based model for identifying four levels of depression.

We have structured the rest of the paper as follows. In section 2, we have discussed the approaches with merits and demerits of the analogous depressive text detection systems. Section 3 describes our working process, including dataset creation, pre-processing tasks, feature selection, and explanation of ML models. Then, we specify the experimental configuration and discuss the implementation process in section 4. In section 5, we have displayed implementation results and elaborated on comparative performance analysis. Finally, section 6 states a concluding remark and a plan for future research.

## 2 Related Work

Finding depressive people is one of the primary concerns in society. Researchers are working in this regard by identifying the depression-related text. Unlike English and other high-resource languages, we have detected several small works for identifying Bengali depressive text. We illustrate this research works below:

Khan et al. [5] experimented on small dataset labeling with happy or sad. In the pre-processing phase, the authors replaced contractions with complete forms, removed less significant characters and stop-words, and applied stemming and tokenization. They did not specify feature selection methods to extract appropriate features. They obtained the highest accuracy (98.00%) using the Long Short-Term Memory (LSTM) model for identifying depression-related text. However, the authors presented very fewer discussions about the error analysis of the detection system.

Mohammed et al. [6] worked on a binary classification problem to detect depressive text. At first, the authors discarded numeric values, punctuation, and stop-words and performed stemming. Then they balanced the dataset using a random under-sampling technique. After that, they utilized an Extra Tree (ET) classifier to extract the features and applied the Principal Component Analysis (PCA) method to reduce feature dimension. They got the best accuracy of 92.80% and the F1-score of 93.61% by exploiting the eXtreme Gradient Boost (XGB) algorithm. The authors are concerned with extracting the contextual meaning of the words in a sentence in their detection system.

Mumu et al. [7] analyzed depressive data to classify it into two classes: depressive and non-depressive. After collecting the dataset, they removed emoticons, punctuation, URLs, and stop-words and applied stemming. Finally, they achieved the highest accuracy of 81.49% by utilizing the Logistic Regression (LR) model with TF-IDF (term frequency-inverse document frequency) vectorization. The authors considerably less discuss the anomaly of their detection system.

Uddin et al. [8] successfully identified a depressive text (depressive or non-depressive) with 86.3% accuracy by exploiting the LSTM model with parameter tuning. In the pre-processing stage, they filtered all characters except the alphanumeric characters. The authors showed less attention to feature selection tasks.

Hossen et al. [11] examined depressive text from various points of view, such as emotional aspects like joy, anger, fear, sadness, and others, dimensions of depression including non-depression, mild, moderate, and severe, just identifying the depressive text with depressive or non- depressive, and so on. They cleaned emoticons, words, stop-words, and non-Bengali characters from the dataset. Then they applied tokenization and normalization operations. The authors obtained the best result of 46.64% F1- score and 48.00% accuracy through the LR model with TF-IDF scoring in four classes of depression. The detection system struggled more (25.00% F1-score) identifying the moderate class depressive text. On the contrary, the authors obtained the highest performance of 80.00% accuracy by exploiting LSTM with word-embedding technique in the binary classification of depressive text. Therefore, we still have sufficient scope to enhance the detection performance of the multi-class classification problem.

Tasnim et al. [9] investigated the depressive dataset, labeling it into two classes: depressive and non-depressive. At first, the authors eliminated special characters, punctuation, and stop words. After that, they used tokenization and stemming operations to the entire dataset. Next, they utilized feature extraction techniques, such as count vectorizer, TF-IDF vectorizer, and word embedding, to select feature vectors. Finally, they applied various ML classifiers, where Decision Tree (DT) outperformed the other models with 97.00% accuracy. The authors experiment with the emoji character set. They observed that emojis with text data show better output than only text data. Although the authors successfully identified the depressive text, they did not deal with the intensity level of the depression. Moreover, the authors did not present a comparative analysis of three feature selection techniques in this investigation.

## 3 Materials and Methods

Identifying depressed people by detecting depressive text is a challenging task. To do this job, at first, we accumulate depression-related text. Afterward, we apply pre-processing tasks for filtering noisy content. Then, we utilize feature selection techniques to extract pertinent features. Lastly, we manipulate ML models to classify four types of Bengali depressive text. Fig. 1 delineates the overall workflow of the multi-class classification system.

### 3.1 Dataset Description

Nowadays, people use social media to treat mental satisfaction via helpful sharing as a quick method of

communication. In this composition, we have standardized the posts and the comments, which are appropriate to depression. Then, we start the data annotation process after finishing the data collection process. We deal with four types of depression data: mild, moderate, severe, and non-depression, in this research.

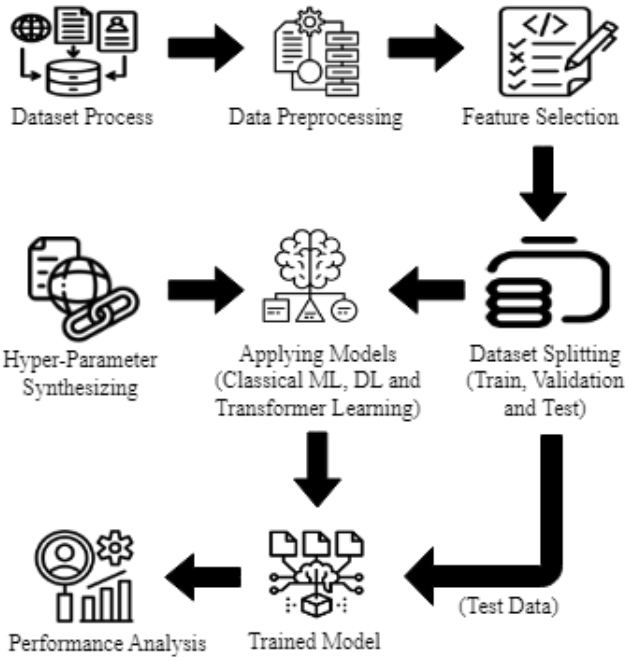


Fig. 1 Overall working process

*Mild:* It is an initial level of depression characterized by persistent feelings of sadness, hopelessness, and low mood, lasting for at least two weeks. It is also known as dysthymia or persistent depressive disorder. Individuals with mild depression may experience a loss of interest in activities they once enjoyed, have trouble sleeping or sleeping too much, feel fatigued or have low energy, experience changes in appetite or weight, and have difficulty concentrating or making decisions.

*Moderate:* It is the medium level of depression. People suffering from this depression may have negative thoughts about themselves, their lives, or the future and may experience feelings of worthlessness or guilt.

*Severe:* It is the extreme level of depression, where people may direct to suicidal thoughts or engage in self-harming behaviors.

*Non-depression:* The text that does not fall into the above three categories is known as non-depressive text.

We have divided the data collection and annotation process into three phases. In the first phase, we have assigned three annotators who collect data according to the definition (mentioned above) of the four classes of depression. The annotators compile an aggregate of 3,000 data points from various Facebook pages and personal profiles with their endeavored diligence. In the second phase, we distribute these data points by changing between the three annotators. After the second time annotation, we filter the mismatched-label data and get 2,598 data points. In the final stage, we examined these data points with a psychologist to correct mislabeling data. We provide an example of each label of depression in Table 1 and present some statistical information about the dataset, including the number of entries (*quantity*), percentage of each class (*percent*), minimum (*minWords*), maximum (*maxWords*), and average (*avgWords*) number of words in a text, in Table 2.

### 3.2 Data Pre-processing

In the pre-processing stage, we have removed various unwanted and noisy elements. At first, 118 duplicated instances are discarded. Then, HTML tags, URLs, punctuation marks, special characters, and digits are removed from the dataset. After that, we eliminated all other languages except Bengali. Lastly, we have performed stop-words removal and stemming [13] operations to the entire dataset. Throughout the empirical experiment, we have observed that stop-word removal and stemming operations degrade the overall performance of the classification system. In addition, retaining stop-words and not performing stemming leads to more variety of features [14],[15]. For that reason, these two pre-processing tasks are not considered in the final experiment.

Table 1 Samples of the depression dataset.

| **Text** | **English Translation** | **Label** |
|---|---|---|
| *কোন জিনিসই অতিরিক্ত হওয়া ভাল নয় দুটি জিনিস ছাড়া। এক: জ্ঞান দুই: ভদ্রতা।* | Nothing is decent in excess except two things. One: Knowledge, Two: Gentleness. | Non-depressive |
| *আটকে রাখার মানুষ অনেক! আগলে রাখার মানুষ কই?* | There are a lot of people to hold back! Where are the people to keep up? | Mild |
| *নিজের ঘরেই যে মূল্যহীন, পুরো পৃথিবীর কাছে তার মূল্য থাকলেও নিজেকে নিঃস্ব বলেই মনে হয়।* | He, who is worthless in his own house, though he has importance to the whole world, feels himself destitute. | Moderate |
| *ইদানিং আমি আমার নিজের রাগকে কন্ট্রোল করতে পারিনা যখন রাগ উঠে তখন আমি আমার ৪ বছরের ছেলেটাকে অনেক মারি, এমন মারা মারি যে মনে হয় সৎ মা ও তাকে এভাবে মারত না, পরে নিজে কান্না করি নিজেরই খারাপ লাগে। ছোট বাচ্চা টাকে গায়ে হাত তুলতে দ্বিধা বোধ করি না, নিজের আত্মবিশ্বাসটা খুবই কম কাউকে সহজে বিশ্বাস করতে পারিনা।* | I cannot control my anger when I get angry. I beat my 4-year-old son a lot; I feel like any stepmother would not beat him like that, and then I cry and feel regret for myself. Sometimes I don't hesitate to beat a small child; my self-confidence is very low, and I can't trust anyone easily. | Severe |

Table 2 Statistical information of the dataset.

| **Class** | ***quantity*** | ***percent*** | ***minWords*** | ***maxwords*** | ***avgwords*** |
|---|---|---|---|---|---|
| Non-depressive | 949 | 36.56% | 4 | 115 | 19 |
| Mild | 775 | 29.85% | 4 | 172 | 23 |
| Moderate | 608 | 23.42% | 4 | 261 | 34 |
| Severe | 264 | 10.17% | 4 | 245 | 40 |

### 3.3 Feature Selection

At this stage, we have extracted features from the pre-processed data. Here, we do this task separately for the three types of ML models, classical ML, DL, and transformer-based pre-trained models. For classical ML models, we have utilized character, word, and combinations of character-word N-gram techniques with TF-IDF scoring. Through the experimental observation, we have fixed the value of N as 3 to 5 for the character N-gram and 1 to 2 for the word N-gram. In the case of

DL models, we have observed through empirical analysis that the fastText word embedding approach gives better output than the word2vec and GloVe methods. The main reason is that the fastText employs the character N-gram technique to get sub-word level features, which handle the unknown word vocabulary of the dataset [16],[17]. Here, we have used the *max_len* (the maximum number of tokens in each text) is 132 and the *vector_size* (the size of the feature vector) is 100 to maintain the same input dimension. In the case of transformer-based models, they use their own embedding techniques, where the BERT uses the WordPiece[4] technique and the XLM-R utilizes the Sentence Piece Model (SPM) [18] method to extract token-based sub-word level features. The SPM combines two sub-word segmentation methods: the uni-gram language model [19] and byte-pair-encoding (BPE) [20].

### 3.4 Model Classifiers

In this experiment, we have chosen nine machine learning models, MNB, SVM, RF, LR, LSTM, BiLSTM, CNN-BiLSTM, BERT, and XLM-R models, which show better performance in several Bengali text classification tasks. The authors of [10],[11],[21], and [22] proposed MNB, LR, SVM, and RF, respectively among the classical ML models for depression or sentiment-related text classification tasks. On the contrary, Khan et al. [5], Tasnim et al. [9], and Mumu et al. [7] achieved better outputs through the LSTM, BiLSTM, and CNN-LSTM-based DL models. However, in recent times, the BERT and XLM-R-based pre-trained models present outstanding performance in the text classification task for low-resource languages [23],[24]. We have explained each model as follows:

Multinomial Naive Bayes (MNB): MNB utilizes the basic principle of the Bayes theorem [25]. Every feature and class variable is computed in the training process through this theorem with prior and conditional probability formulas. Then, the test dataset is subjected to these prior and conditional probabilities. MNB then determines the likelihood for each class of each data point based on the characteristics of the test data and chooses a class with a greater probability. The probability, P(C|F), is measured through the following Bayes theorem:

$$P(C|F) = \frac{P(F|C) * P(C)}{P(F)}$$

where $C$ represents a class variable, and $F$ denotes the distinctive feature.

*Support Vector Machine (SVM):* By maximizing the distance across the margins of the two types of support vectors (like positive and negative), SVM creates a hyper-plane space. This marginal distance leads to the production of a more broadly applicable model. SVM also addresses errors by applying a regularization technique to incorrectly categorized data points that are based around the soft margin hyper-plane [26].

*Random Forest (RF):* The RF comes by combining several decision trees that are computed individually [27]. Here, the "Gini Impurity" formula is used to obtain the information gained for each tree as follows:

$$Gini = 1 - \sum_{i=1}^{N} (P_i)^2$$

where $N$ is the total number of possible classes, and $P_i$ is the probability of $i^{th}$ class.

*Logistic Regression (LR):* The operational procedure of this model consists of the sigmoid and cost functions, calculated as Eqs. 1 and 2, respectively.

$$h_\theta(x) = \frac{1}{1 + e^{-\beta^T x}} \quad (1)$$

$$C(\theta) = \frac{1}{j} \sum_{i=1}^{j} c(h_\theta(x_i), y_i) \quad (2)$$

$$c(h_\theta(x), y) = \begin{cases} -\log(1 - h_\theta(x)), & if\ y = 0 \\ -\log(h_\theta(x)), & if\ y = 1 \end{cases}$$

where, $\beta^T$ denotes the transpose of regression coefficient matrix ($\beta$), $j$ indicates the total training observations, $h_\theta(x_i)$ represents the hypothesis function of the $i^{th}$ training sample, and $y_i$ gives the true output of the $i^{th}$ training observation.

*Long Short-Term Memory (LSTM):* This paradigm can detain long-term dependencies in sequential data such as time series, text, and speech [28]. They manage the flow of information using a memory cell and gates, enabling them to keep or get rid of information as needed. The input gate, forget gate and output gate are all connected to the memory cell in different ways. The input gate selects the information that needs to be stored in the memory cell. The memory cell's information is selected for discard by the forget gate. As a last step, the output gate unit selects the data that will be sent to the subsequent LSTM cell state.

*Bidirectional LSTM (BiLSTM):* A Bidirectional LSTM [29] is a sequential data processing model that has two LSTM layers, one of which processes the input sequence forward and the other of which processes it backward. The final forecast is created by merging the results from the two directions, with each layer maintaining its own hidden states. Additionally, BiLSTMs effectively expand the network's pool of knowledge, giving the algorithm better context.

*Convolutional Neural Network with BiLSTM (CNN-BiLSTM):* A Convolutional Neural Network with Bidirectional Long Short-Term Memory (CNN-BiLSTM) is a hybrid model that combines the unique features of both convolutional neural networks (CNNs) and bidirectional LSTM (BiLSTM) networks. The CNN-BiLSTM architecture consists of three main features: The CNN consists of one or more convolutional layers, followed by pooling layers and activation functions. Convolutional layers apply filters to the input data, rotate the spatial dimensions, and create feature maps that illustrate various facets of the input. The network may gather context data for the past and the future thanks to its bidirectional processing. The final sequence representation is created by concatenating the outputs of the forward and backward LSTMs. After the CNN-BiLSTM levels, fully connected layers apply non-linear transformation changes to all the neurons from the preceding layers and connect them to the output layer, where they produce the final predictions or classifications [30].

*Bidirectional Encoder Representations from Transformers (BERT):* BERT [31] can be fine-tuned for a variety of downstream natural language processing (NLP) tasks, including question-answering, named entity recognition, and sentiment

[4] https://huggingface.co/course/chapter6/6?fw=pt

analysis. It is pre-trained on substantial volumes of unlabeled text data. During the pre-trained, BERT uses the Next Sentence Prediction (NSP) and Mask Language Model (MLM) unsupervised tasks to learn the context of the specified languages. BERT takes into consideration both the left and right contexts of each word in a phrase simultaneously, unlike traditional models that analyze text sequentially (left-to-right or right-to-left).

*Cross-lingual Language Model with Robustly Optimized BERT (XLM-RoBERT):* XLM-RoBERTa [24] is a sophisticated language model with numerous layers and features. Each encoder layer consists of sub-layers, including multi-head self-attention and position-wise feed-forward networks, which allow the model to capture the contextual relationships between words in the input sequence. A series of tokens encoding the text is commonly used as the input to XLM-RoBERTa. Using SPM [18], these tokens are first transformed into numerical embeddings. The embeddings accurately depict each token's semantic meaning within the context of the phrase. It utilizes cross-lingual pre-training to develop its ability to comprehend and produce text in several languages. The model is trained on a sizable corpus of monolingual data from many languages during pre-training. This enables it to acquire representations independent of a language and to identify the linguistic traits that are common to all languages. After pre-training, the model is more trained on certain downstream tasks, such as the classification of texts, named entity identification, or machine translation. The model's knowledge is adjusted during this step of fine-tuning, which also aids in enhancing the model's performance on the targeted language-related tasks.

## 4 Experimental Set-up and Implementation

The deep learning and the transformer-based models need a highly configured processing environment. For this purpose, we have used the Google Colab cloud environment, which provides the Jupyter Notebook Python programming language editor and felicitates NVIDIA Graphics Processing Units (GPU) with many built-in Python modules and packages [32]. In this case, the GPU is a Tesla T4 with 15GB of RAM. The experimental configuration of this work is discussed from three aspects, for classical ML, DL, and transformer-based models. Regarding classical ML models, at first, the dataset is split into two parts: train and test with a ratio of 90% and 10%, respectively. After that, sklearn python packages are used with default hyper-parameter values to execute each model. During the execution, a 10-fold cross-validation approach is incorporated to get a more reliable classification system [33]. In the case of DL models, we have split the dataset into train, validation, and test with a ratio of 70%, 20%, and 10%, respectively. We have utilized tensorflow Python packages to implement DL models. We use two LSTM layers, one BiLSTM layer, and one CNN and one BiLSTM layer for implementing LSMT, BiLSTM, and CNN-BiLSTM models, respectively. In every model, we add two hidden (dense) layers and one output layer. In addition, a dropout function is utilized between every two layers to handle the overfitting problem [34]. Throughout the empirical experiment, the overall parameter values of DL models are settled which are figured out in Table 3. On the other hand, we utilize ktrain python packages to implement transformer-based models. We use the same dataset splitting ratio as DL models. We have tried various values of *max_len*, *learning_rate*, and *batch_size*. We obtained the best-configured values for the *learning_rate* of 4e-05 and the *batch_size* of 12 for the bert-base-multilingual-uncased and the xlm-roberta-base models. However, *max_len* is set as 120 for the BERT-based model and 124 for the XLM-R-based model. In this work, we run each model up to 10 epochs.

Table 3 Hyper-parameter values during the implementation of deep learning models.

| **Parameter** | **Data Type** | **Description** | **Value** |
|---|---|---|---|
| LSTM units | Integer | The amount of LSTM output units. | 140 |
| BiLSTM units | Integer | The amount of BiLSTM output units (for the BiLSTM model) in each direction. | 140 |
| Filters | Integer | The number of convolution filters. | 192 |
| Kernel size | Integer | The length of the convolution window. | 3 |
| Pooling types | String | Two types: max pooling and average pooling, are used to reduce the dimension of the feature map. | max pooling’ |
| BiLSTM units | Integer | The amount of BiLSTM output units (for the CNN-BiLSTM model) in each direction. | 128 |
| Hidden units | Integer | The number of neurons of each dense layer. | 128 |
| Activation function | String | The function defines the output value of each neuron. | ’relu’ |
| Kernel initializer | String | A procedure to assign a set of small random values of a neural network at the very early stage. | ’glorot_uniform’ |
| Dropout rate | Float | Fraction of the number of neurons to drop. | 0.2 |
| Learning rate (Adam) | Float | It controls how fast a loss function moves toward a point where the curves meet. | 0.0001 |
| Batch size | Integer | The number of samples participating in each iteration during training of the model. | 12 |
| Epochs | Integer | The number of times all training data is utilized to train the model. | 30 |

## 5 Result and Discussion

For measuring the performance of each ML model, we have taken four evaluation metrics: precision, recall, F1-score, and accuracy. We use the weighted average value for each metric. We have illustrated this section from four aspects: results of classical ML models, results of DL models, results of transformer-based models, and overall result analysis. First, we discuss the comparative analysis of classical ML models. Subsequently, we compare the outputs of the DL models. We then analyze the results of the transformer-based pre-trained models. Lastly, we present the overall analysis of the ML models.

### 5.1 Result of Classical ML Models

The outputs of four classical ML models: SVM, RF, LR, and MNB are articulated in Table 4. Here, the SVM with character N-grams features obtains the highest performance in precision, recall, F1-score, and accuracy of 54.83%, 54.45%, 52.13%, and 54.45%, respectively. Character N-grams give better accuracy than the word N-grams and the combined N-grams to the other three models: 52.47% for MNB, 51.86% for RF, and 52.83% for LR. The main reason behind this

performance is that character N-grams technique produces many sub-words level features, which play an important role for classifying Bengali depressive text.

Table 4 Performance evaluation of the classical ML models

| Approach | Precision (%) | Recall (%) | F1-score (%) | Accuracy (%) |
|---|---|---|---|---|
| Character N-grams + SVM | **54.83** | **54.45** | **52.13** | **54.45** |
| Word N-grams + SVM | 50.95 | 50.82 | 47.16 | 50.82 |
| Combined N-grams + SVM | 53.24 | 53.32 | 51.65 | 53.32 |
| Character N-grams + MNB | 53.12 | 52.47 | 50.66 | 52.47 |
| Word N-grams + MNB | 48.83 | 48.23 | 47.69 | 48.23 |
| Combined N-grams + MNB | 52.89 | 51.70 | 51.17 | 51.70 |
| Character N-grams + RF | 51.22 | 51.86 | 47.54 | 51.86 |
| Word N-grams + RF | 48.69 | 46.98 | 40.76 | 46.98 |
| Combined N-grams + RF | 50.68 | 50.25 | 45.94 | 50.25 |
| Character N-grams + LR | 53.20 | 52.83 | 49.30 | 52.83 |
| Word N-grams +LR | 49.06 | 49.36 | 43.46 | 49.36 |
| Combined N-grams + LR | 52.62 | 52.79 | 49.40 | 52.79 |

## 5.2 Result of DL Models

We notify the implementation results of three DL models: LSTM, BiLSTM, and CNN-BiLSTM in Table 5. The BiLSTM model obtains the top score in the four-evaluation metrics: 58.96% of precision, 58.87% of recall, 56.50% of F1-score, and 58.87% of accuracy. This DL model gets contextual understanding from a sentence with forward and backward directions, which are the principal reasons for the better performance, compared to the other two DL models. However, the CNN-BiLSTM model shows a similar output to BiLSTM with a slight decrease in value. Since the LSTM model works only forward direction, the F1-score and accuracy give a lower value of 52.75% and 56.05%, respectively.

Table 5 Performance evaluation of the DL models

| Approach | Precision (%) | Recall (%) | F1-score (%) | Accuracy (%) |
|---|---|---|---|---|
| LSTM | 50.77 | 56.05 | 52.75 | 56.05 |
| BiLSTM | **58.96** | **58.87** | **56.50** | **58.87** |
| CNN-BiLSTM | 58.52 | 58.06 | 55.86 | 58.06 |

Table 6 Performance evaluation of the transformer-based models

| Approach | Precision (%) | Recall (%) | F1-score (%) | Accuracy (%) |
|---|---|---|---|---|
| bert-base-multilingual-uncased | 59.45 | 56.85 | 56.87 | 56.85 |
| xlm-roberta-base | **61.63** | **60.89** | **61.11** | **60.89** |
| LR + TF-IDF [15] | | | 46.64 | 48.00 |

## 5.3 Result of Transformer-based Models

We figure out the performance of two transformer-based models: BERT and XLM-R, in Table 6. The XLM-R-based approach achieves better results compared to the BERT-based method and the earlier work [11] in terms of precision (61.63%), recall (60.89%), F1-score (61.11%), and accuracy (60.89%). The XLM-R is pre-trained over a large dataset (two terabytes) from one hundred languages, including low-resource languages like Bengali. Furthermore, this pre-trained model utilizes the SPM technique with a larger vocabulary (250k) to get sub-word level feature vectors [24]. For these reasons, the XLM-R-based approach shows much better performance for detecting depressive text.

## 5.4 Overall Result Analysis

By observing Table 4-Table 6 we see that the XLM-R-based approach outperforms the other ML models in this research. This model achieves the best score in all four-evaluation metrics such as precision, recall, F1-score, and accuracy. For that reason, we only focus on this model in the subsequent analysis.

Fig. 2 displays the training and validation accuracy curve for the XLM-R-based approach. The training curve gradually increases from 0.35 to 0.90 until the tenth epoch. On the contrary, the validation curve shows ups and downs with an upward trend, and the final accuracy value reaches 0.56 at the tenth epoch.

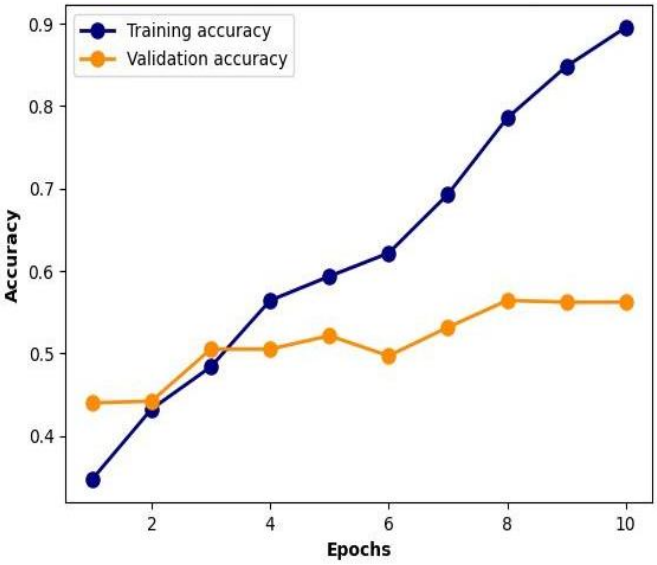


Fig. 2 Training and validation accuracy curve

We also visualize the training and validation loss in Fig. 3. The training loss moderately reduces from the first (loss value is 1.32) to the tenth (loss value is 0.31) epoch. However, the validation loss presents a different scenario. In the third epoch, we count the lowest loss as 1.07. After that, the curve slowly rises until the last epoch, and the final loss value reaches 1.47.

We now observe the class-wise performance of the depressive text dataset (Table 7).

Table 7 Class-wise performance measure of the XLM- R-based approach

| Class | Precision (%) | Recall (%) | F1-score (%) |
|---|---|---|---|
| Non-depressive | 77.38 | 73.03 | 75.14 |
| Mild | 49.43 | 57.33 | 53.09 |
| Moderate | 55.77 | 50.00 | 52.73 |
| Severe | 56.00 | 53.85 | 54.90 |

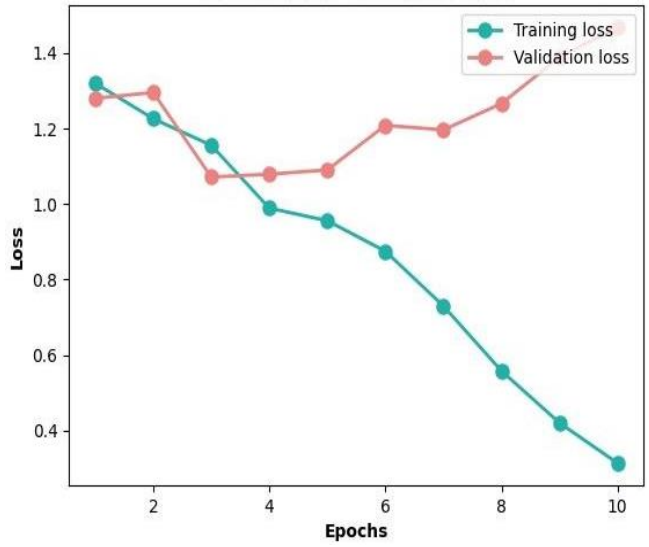


Fig. 3 Training and validation loss curve

Since the Non- depressive class comprises the most data (36.56%), it performs much better in the precision of 77.38%, recall of 73.03%, and F1-score of 75.14% than the other three classes. The Mild, Moderate, and Severe classes give nearly similar F1-scores of 53.09%, 52.73%, and 54.90%, respectively.

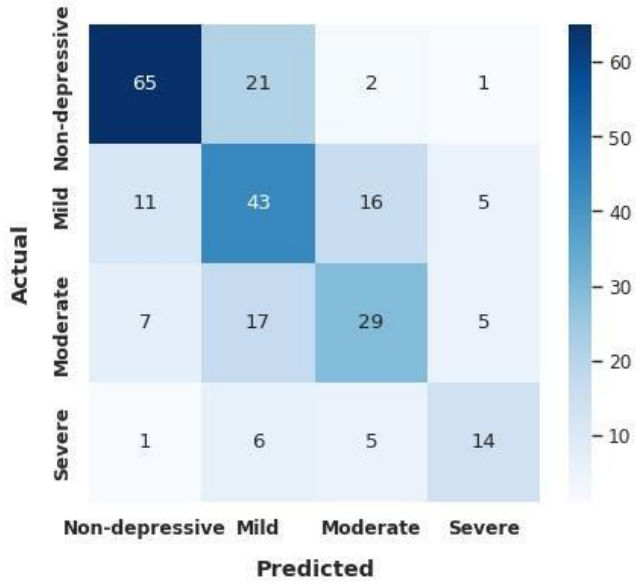


Fig. 4 Confusion matrix of the XLM-R-based detection system

The highest precision and recall value among these three classes is 56.00% for the severe class and 57.33% for the Mild class. Now, we visualize biases and misclassification issues of every class in Fig. 4. The Non- depressive is mostly biased by the Mild class, and vice-versa. The Mild and Moderate categories are also highly influenced by each other. And the Severe class overlaps with the Mild and Moderate groups. In a nutshell, the three classes: Mild, Moderate, and Severe, have more inclinations to overlap with one another. Thus, we get lower performance scores for these three classes (Table 7).

## 6 Conclusion

Our proposed XLM-R-based approach outperforms the existing works for the multi-class classification of the Bengali depressive text. We have tried various feature selection techniques and ML models in this research. Finally, we observe that the XLM-R-based approach successfully detects a depressive text with a 61.11% F1-score and 60.89% accuracy. However, there is still enough space to enhance the detection system. Since we experiment with a small dataset, we will enlarge our dataset in the subsequent research. In addition, we will handle the overlapping issue of the three depression classes: Mild, Moderate, and Severe. Lastly, we will concentrate on pre-processing and feature selection levels and apply other cutting-edge ML models to improve the detection performance of the depressive text.

## Acknowledgment

The authors are thankful to Mrs. Farhana Yeasmin Satu, a Psychologist in Bangladesh, who provides expert opinions in the entire process of the dataset development. Mrs. Satu completed her Bachelor of Science and Master of Science degrees from the University of Chittagong, Bangladesh. She achieved a certificate in Mental Health, Depression, Anxiety, and Autism from the United States of America (USA).

## References

[1] Arusha, A.R. and Biswas, R.K., 2020. Prevalence of stress, anxiety and depression due to examination in Bangladeshi youths: A pilot study. Children and youth services review, 116, p.105254.

[2] Islam, M.S., Rahman, M.E., Moonajilin, M.S. and van Os, J., 2021. Prevalence of depression, anxiety and associated factors among school going adolescents in Bangladesh: Findings from a cross-sectional study. Plos one, 16(4), p.e0247898..

[3] Ogbo, F.A., Mathsyaraja, S., Koti, R.K., Perz, J. and Page, A., 2018. The burden of depressive disorders in South Asia, 1990–2016: findings from the global burden of disease study. BMC psychiatry, 18(1), pp.1-11.

[4] Mashreky, S.R., Rahman, F. and Rahman, A., 2013. Suicide kills more than 10,000 people every year in Bangladesh. Archives of Suicide Research, 17(4), pp.387-396.

[5] Rafidul Hasan Khan, M., Afroz, U.S., Masum, A.K.M., Abujar, S. and Hossain, S.A., 2021. A deep learning approach to detect depression from Bengali text. In Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2020, Volume 2 (pp. 777-785). Springer Singapore.

[6] Mohammed, M.B., Abir, A.S.M., Salsabil, L., Shahriar, M. and Fahmin, A., 2021, December. Depression Analysis from Social Media Data in Bangla Language: An Ensemble Approach. In 2021 Emerging Technology in Computing, Communication and Electronics (ETCCE) (pp. 1-6). IEEE.

[7] Mumu, T.F., Munni, I.J. and Das, A.K., 2021. Depressed people detection from bangla social media status using lstm and cnn approach. Journal of Engineering Advancements, 2(01), pp.41-47.

[8] Uddin, A.H., Bapery, D. and Arif, A.S.M., 2019, July. 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 (IC4ME2) (pp. 1-4). IEEE.

[9] Tasnim, F., Habiba, S.U., Nafisa, N. and Ahmed, A., 2022. Depressive Bangla text detection from social media post using different data mining techniques. In Computational Intelligence in Machine Learning: Select Proceedings of ICCIML 2021 (pp. 237-247). Singapore: Springer Nature Singapore.

[10] Khan, M.R.H., Afroz, U.S., Masum, A.K.M., Abujar, S. and Hossain, S.A., 2020, July. Sentiment analysis from bengali depression dataset using machine learning. In 2020 11th international conference on computing, communication and networking technologies (ICCCNT) (pp. 1-5). IEEE.

[11] Hossen, I., Islam, T., Rashed, M.G. and Das, D., 2022, October. Early Suicide Prevention: Depression Level Prediction Using Machine Learning and Deep Learning Techniques for Bangladeshi Facebook Users. In Proceedings of International Conference on Fourth Industrial Revolution and Beyond 2021 (pp. 735-747). Singapore: Springer Nature Singapore.

[12] Das, A., Sharif, O., Hoque, M.M. and Sarker, I.H., 2021. Emotion classification in a resource constrained language using transformer-based approach. arXiv preprint arXiv:2104.08613.

[13] Seddiqui, M.H., Maruf, A.A.M. and Chy, A.N., 2016. Recursive suffix stripping to augment bangla stemmer. In International Conference Advanced Information and Communication Technology (ICAICT).

[14] Ahmed, M.T., Rahman, M., Nur, S., Islam, A. and Das, D., 2021, February. Deployment of machine learning and deep learning algorithms in detecting cyberbullying in bangla and romanized bangla text: A comparative study. In 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT) (pp. 1-10). IEEE.

[15] Kumar, R., Lahiri, B. and Ojha, A.K., 2021. Aggressive and offensive language identification in hindi, bangla, and english: A comparative study. SN Computer Science, 2(1), p.26.

[16] Joulin, A., Grave, E., Bojanowski, P. and Mikolov, T., 2016. Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.

[17] Grave, E., Bojanowski, P., Gupta, P., Joulin, A. and Mikolov, T., 2018. Learning word vectors for 157 languages. arXiv preprint arXiv:1802.06893.

[18] Kudo, T. and Richardson, J., 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:1808.06226.

[19] Kudo, T., 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. arXiv preprint arXiv:1804.10959.

[20] Sennrich, R., Haddow, B. and Birch, A., 2015. Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909.

[21] N. R. Bhowmik, M. Arifuzzaman, M. R. H. Mondal, and M. Islam, "Bangla text sentiment analysis using supervised machine learning with extended lexicon dictionary," Natural Language Processing Research, vol. 1, no. 3-4, pp. 34–45, 2021.

[22] Tabassum, N. and Khan, M.I., 2019, February. Design an empirical framework for sentiment analysis from Bangla text using machine learning. In 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE) (pp. 1-5). IEEE.

[23] Jahan, M.S., Haque, M., Arhab, N. and Oussalah, M., 2022, June. BanglaHateBERT: BERT for Abusive Language Detection in Bengali. In Proceedings of the Second International Workshop on Resources and Techniques for User Information in Abusive Language Analysis (pp. 8-15).

[24] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L. and Stoyanov, V., 2019. Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116.

[25] Xu, S., Li, Y. and Wang, Z., 2017. Bayesian multinomial Naïve Bayes classifier to text classification. In Advanced Multimedia and Ubiquitous Engineering: MUE/FutureTech 2017 11 (pp. 347-352). Springer Singapore.

[26] Cortes, C. and Vapnik, V., 1995. Support-vector networks. Machine learning, 20, pp.273-297.

[27] Ali, J., Khan, R., Ahmad, N. and Maqsood, I., 2012. Random forests and decision trees. International Journal of Computer Science Issues (IJCSI), 9(5), p.272.

[28] Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.

[29] Schuster, M. and Paliwal, K.K., 1997. Bidirectional recurrent neural networks. IEEE transactions on Signal Processing, 45(11), pp.2673-2681.

[30] Karim, M.R., Chakravarthi, B.R., McCrae, J.P. and Cochez, M., 2020, October. Classification benchmarks for under-resourced bengali language based on multichannel convolutional-lstm network. In 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA) (pp. 390-399). IEEE.

[31] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[32] Carneiro, T., Da Nóbrega, R.V.M., Nepomuceno, T., Bian, G.B., De Albuquerque, V.H.C. and Reboucas Filho, P.P., 2018. Performance analysis of google colaboratory as a tool for accelerating deep learning applications. IEEE Access, 6, pp.61677-61685.

[33] Wong, T.T. and Yeh, P.Y., 2019. Reliable accuracy estimates from k-fold cross validation. IEEE Transactions on Knowledge and Data Engineering, 32(8), pp.1586-1594.

[34] Baldi, P. and Sadowski, P.J., 2013. Understanding dropout. Advances in neural information processing systems, 26.