*Computer Systems Science & Engineering*
DOI: 10.32604/csse.2023.027841
***Article***

Tech Science Press

# Deep-BERT: Transfer Learning for Classifying Multilingual Offensive Texts on Social Media

**Md. Anwar Hussen Wadud[1], M. F. Mridha[1], Jungpil Shin[2,*], Kamruddin Nur[3] and Aloke Kumar Saha[4]**

[1]Department of Computer Science and Engineering, Bangladesh University of Business and Technology, Dhaka, Bangladesh
[2]School of Computer Science and Engineering, University of Aizu, Aizuwakamatsu, Japan
[3]Department of Computer Science, American International University-Bangladesh, Dhaka, Bangladesh
[4]Department of Computer Science and Engineering, University of Asia Pacific, Dhaka, Bangladesh
*Corresponding Author: Jungpil Shin. Email: jpshin@u-aizu.ac.jp
Received: 26 January 2022; Accepted: 11 March 2022

**Abstract:** Offensive messages on social media, have recently been frequently used to harass and criticize people. In recent studies, many promising algorithms have been developed to identify offensive texts. Most algorithms analyze text in a unidirectional manner, where a bidirectional method can maximize performance results and capture semantic and contextual information in sentences. In addition, there are many separate models for identifying offensive texts based on monolingual and multilingual, but there are a few models that can detect both monolingual and multilingual-based offensive texts. In this study, a detection system has been developed for both monolingual and multilingual offensive texts by combining deep convolutional neural network and bidirectional encoder representations from transformers (Deep-BERT) to identify offensive posts on social media that are used to harass others. This paper explores a variety of ways to deal with multilingualism, including collaborative multilingual and translation-based approaches. Then, the Deep-BERT is tested on the Bengali and English datasets, including the different bidirectional encoder representations from transformers (BERT) pre-trained word-embedding techniques, and found that the proposed Deep-BERT's efficacy outperformed all existing offensive text classification algorithms reaching an accuracy of 91.83%. The proposed model is a state-of-the-art model that can classify both monolingual-based and multilingual-based offensive texts.

**Keywords:** Offensive text classification; deep convolutional neural network (DCNN); bidirectional encoder representations from transformers (BERT); natural language processing (NLP)

## 1 Introduction

Offensive texts are intended to annoy someone intentionally. Offensive content motivates individuals to participate in evil acts against rules and regulations, offends religious sentiments, and urges people to violence for no valid cause. On social media, offensive materials can be transmitted through a variety of

methods, including text, photos, voice, video, and graphical representations. On the other hand, the text format is the most widely utilized type of social networking. Furthermore, a hashtag-embedded message may be simply turned into links, and similar hashtags can be used to linkage the content to another group. As a result, text transmitted more quickly than any other method and aids in the quick retrieval of data. This study emphasizes on word-based social media content and classifies whether any text is offensive or not.

The problem of classifying a group of texts written in different languages (e.g., Bengali, English, Hindi, etc.) and consisting of a set of predefined classes across different languages is known as multilingual text classification (MTC). This differs from cross-language text classification [1], which requires a document written in one language to be classified using a classification system learned in another language. The problem of multilingual offensive text classification is often described as the work of a supervised classifier, in which a machine is trained with manually labeled sentences to multilingual offensive or non-offensive terms. The International Workshop on Semantic Evaluation (SemEval) has received proposals from over 100 groups for shared assignments [2]. Several attempts have been made to classify offensive languages, but only in monolingual contexts, focusing on English [2], Bengali [3], and other languages. Mridha et al. [3] proposed a hybrid model for the Bengali offensive text classification. They proposed long short-term memory (LSTM)-BERT and Adaboost-BERT models separately and then combined these two models to create the final L-Boost model. In their research, they analyzed offensive text in both Bengali and Banglish texts and compared the results with other existing models. This study was inspired by Mridha et al. [3] to identify offensive text on multilingual-based datasets. In addition, numerous languages predominate as the medium of communication, resulting in the diversification of multilingual text classification, which can be effective in a variety of situations such as identifying offensive text and preventing intrusion.

Multilingual text classification (MTC) differs from cross-language text categorization. In MTC, text written in one language must be classified into a category system learned in another language. To deal with this MTC problem, there are several approaches: 1) developing monolingual classifiers for each language [4] and 2) formatting all texts with a translation phase and then developing a classification system [5]. 3) Developing a single classifier but training with multilingual texts. El-Alami et al. [6] argued that although BERT can extract complex features from raw text automatically, it has never received attention in multilingual text classification.

The objective of this research is to propose a BERT-based deep convolutional neural network called Deep-BERT by integrating several simultaneous frames of a fixed-layer convolutional neural network (CNN) with BERT, which can classify both monolingual and multilingual offensive text. BERT is used as a sentence encoder because it can properly obtain the contextual reflection of a statement. With its remarkable ability to detect long-distance relationships in semantics and phrases, our suggested technique improves the accuracy of identifying offensive content. To develop the proposed model, a classification layer is appended on top of the encoder output, the output sequence is multiplied by the embedding matrix, and the SoftMax function is used to determine the likelihood of each vector. The proposed model consists of three concurrent frames of a 1D-convolutional neural network with BERT, each with a maximum pooling layer, that have different kernel values and filters. The text was analyzed using different CNNs by changing the kernel size (different n-grams), filters, and numerous hidden layers. Deep-BERT comprises six convolutional and max-pooling layers, two densely connected layers, and one BERT embedding layer. Different filters were applied to each layer to extract data from the training dataset. This composition of BERT with a one-dimensional deep CNN (1d-CNN) can process both organized and unstructured texts. It efficiently solves ambiguity, which is the most difficult aspect of interpreting natural language. The effectiveness of the proposed model is verified.

The overall contributions of this paper are summarized as follows:

i) Preprocessing of the dataset to identify offensive texts in both monolingual and multilingual formats.
ii) The proposed Deep-BERT model is based on a deep CNN (DCNN) and BERT to detect offensive texts.
iii) The use of various pre-trained embeddings from transformers, such as BERT, multilingual BERT (mBERT), and BanglaBERT, to classify offensive texts across multilingualism and compare the proposed model with existing models.

The remainder of this paper is organized as follows: the literature review is described in Section 2, the methodology is described in Section 3, the pre-processing techniques are addressed in Section 3-A, and the results are presented in Section 4. Finally, Section 5 describes the limitations and future work, and Section 6 concludes the study.

## 2 Related Works

In this section, first, monolingual-based existing offensive text classification works will be discussed in English and Bengali. The existing functions of multilingual-based offensive text classification are then discussed. Multilingual text classification for offensive content is a new field of study in natural language processing. In this study, existing techniques and methods for detecting multilingual offensive text were explored.

Yadav et al. [7] proposed a text-filter mechanism and built an abusive keyword database. The authors described numerous text-classification strategies in their study. Nobata et al. [8] suggested a machine-learning approach for identifying abusive language from online comments. Using the Aho-Corasick string pattern-matching algorithm in their dictionary to identify keywords, Yadav and Manwatkar [9] suggested a text filtration and categorization strategy. In their work [9], forbidden terms were not made publicly available and semantically comparable words were not considered. Chu et al. [10] used LSTM and CNN deep learning algorithms as well as a neural language processing tool to recognize and categorize offensive remarks. They evaluated three methods: a recurrent neural network (RNN) with word embedding and LSTM cell, a CNN with word embedding, and a CNN with character embedding. For abuse categorization, Wulczyn et al. [11] explored the use of logistic regression and multilayer perceptrons. The researchers then analyzed their findings using a human baseline. Shah et al. [12] proposed a deep-learning-based fusion model for analyzing patient feedback. In their model, they considered both textual and photographic data to analyze patients' feelings. They stated that the performance of the deep learning-based model was better than that of traditional machine learning models [13].

Ishmam et al. [14] introduced the gated RNN (GRNN) approach to identify dirty Bengali texts and achieved a 70.10 percent success rate in 5 K datasets with six classifications. To train their model and achieve precision, 900 documents were used for each class. Eshan and Hasan [15] investigated a variety of machine classifiers, including multinomial naive Bayes (MNB), random forest, and support vector machine (SVM) with linear, polynomial, and sigmoid radial basis function kernels (RBF). CountVectorizer and term frequency–inverse document frequency (TF-IDF) was used to compare and learn each classifier based on n-gram features. The support vector machine with a linear kernel outperformed the TF-IDF vectorizer based on trigram features, achieving 85% accuracy. Karim et al. [16] proposed DeepHateExplainer architecture, focusing on different BERT models. They used XML, a robust optimized BERT pretraining approach (RoBERTa) [17], multilingual BERT, and Bangla-BERT to build their models. They worked on various Bengali abusive text classes, all of which were written in Bengali or Bengali with English terms. For Bengali text categorization, Rahman et al. [18] employed BERT and efficiently learned an encoder that classifies token replacement accurately (ELECTRA). In Bengali text,

they outlined the systematic technique of the BERT model for sentiment analysis. Recently, several NLP researchers have used BERT-based transformer models for text categorization [3], [19], [20]. A few researchers [3], [19], [20] used fine-tuning of BERT transformers, whereas Cohan et al. [19] exploited pre-trained and fine-tuned datasets in sequential text classification and achieved better accuracy than other models. Hussain et al. [21] reviewed comments on different social media platforms and proposed an algorithm for identifying abusive Bengali statements with string unigram characteristics surpassing other criteria. Granizo et al. [22] used computer vision and NLP on Twitter and linked websites to detect offensive information. Illegal texts were classified based on certain aspects including human trafficking, abductions, and disappearances.

Ranasinghe and Zampieri [23] developed an offensive message detection model for different languages. To recognize Spanish, Hindi, and Bengali texts, they employed cross-lingual embedding and a transformer-based approach (XLM-R). Lee et al. [24] introduced a semantic similarity index-based multilingual language classification system for English and Chinese. Mittal and Dhyani [25] examined the classification of multilingual texts using n-gram algorithms. They analyzed MTC in three languages: English, Italian, and Spanish. They started by guessing the language of a paper and then classified it using a naive Bayes classifier. Rani et al. [26] used several machine learning techniques, such as the decision tree (DT), K-nearest neighbors (KNN), SVM, genetic algorithms, and self-organizing maps to handle the issues of MTC in English and Hindi texts. They used a variety of feature extraction strategies to improve experimental performance. The multilingual BERT (mBERT) [27] and cross-lingual model (XLM) [28] are two multilingual masked language models that have pretrained giant transformer models in different languages. These methods have been tested using cross-lingual comprehension tests [28,29].

Despite a significant amount of research on cross-lingual text categorization, research on multilingual text classification (MTC) is rare, and only a few studies have utilized traditional approaches, such as KNN and SVM. Furthermore, studies in the area of offensive language identification have hitherto been conducted solely using a monolingual paradigm. Delvin et al. [27] proposed BERT, which is designed to pre-train deep bidirectional representations from unlabeled text, is promising, and can be used effectively for multilingual texts.

## 3 Methodology

Fig. 1 presents the step-by-step workflow of the proposed offensive text detection process using social media datasets and classification algorithms using the BERT model. The proposed offensive text detection system has several modules, including preprocessing, tokenization, text presentation, and classification.

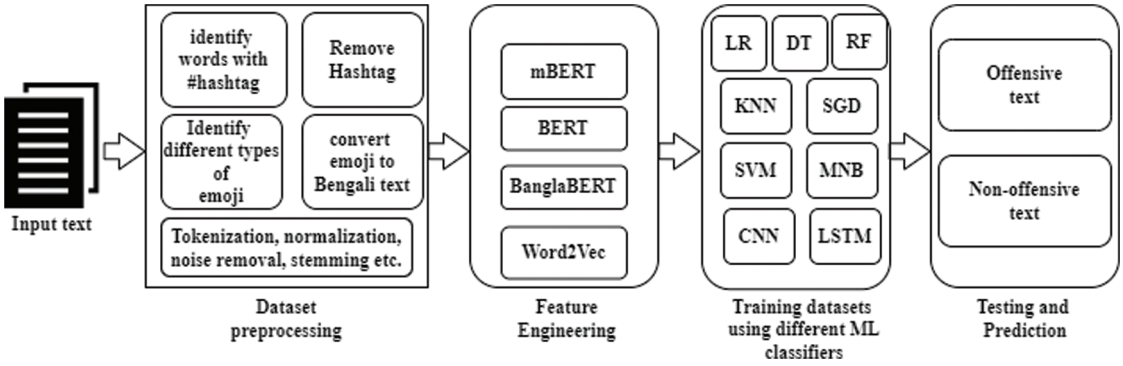



**Figure 1:** The workflow of the proposed offensive text detection process

### *3.1 Dataset Preprocessing*

Text pre-processing is an important aspect of automatic text classification. The most common text pre-processing methods include punctuation, commas, and case transitions. The preprocessing phase consisted of several stages to extract only the necessary details. In addition to common text preprocessing, more steps are required to preprocess multilingual texts found on social media. These additional pre-processing steps are described in the following paragraph.

#### *3.1.1 Emoji and Emoticon Conversion*

People use various emoticons, emojis, and text symbols in social media messages to express their feelings and thoughts. Emojis, emoticons, and text symbols are often nonverbal, brief expressions of feelings that are not expressed as sentences. Notably, emojis, emoticons, and text symbols can also turn out to be offensive; thus, emojis and emoticons are significant criteria for detecting offensive comments.

Tab. 1 shows some emojis used for offensive text on social media. For example, #sexy_baby 🍑 is an offensive text that can be identified based on the body part (butt) that resembles an emoji symbol [10]. Emojis have different meanings depending on culture, language, and region [10,30]. In Bengali, several emojis have distinct meanings compared to English and vice versa. For example, in English, the "Pointed Finger" emoji normally indicate "pointed right" while in Bengali, it means "fingering." Tab. One represents the generic meaning of several emoji icons and their contextual meanings in the Bengali culture. Using Unicode [31], emojis and emoticons were transformed into readable forms. All emojis and emotions were scraped and specified on the Unicode website using the Beautifulsoup4 [32] Python library module. Next, emojis and emoticons are substituted with corresponding senses in English text if they exist. For Bengali, all emoji texts were translated from English to Bengali using the Python Translate library package [32], and it was manually identified which emojis are objectionable in Bengali or not.

**Table 1:** Emojis frequently used for suggestive sexual messages in english and bengali language [30]

| Emoji for English Language | | | Emoji for Bengali Language | | |
|---|---|---|---|---|---|
| Emoji | Generic Meaning | Contextual Meaning | Emoji | Generic Meaning | Contextual Meaning |
| 🍆 | Eggplant | Penis | 👉 | Pointed right | অঙ্গুলিসঁচালন (Fingering) |
| 💦 | Sweat droplets | Ejaculation | 🔥 | Fire | যৌন আবেদনপূর্ণ (Hot & Sexy) |
| 😘 | Blowing a kiss face | Kissing face | 🍒 | Cherries | স্তন (Female Breast) |
| 👅 | Tongue | Oral sex | 👅 | Tongue | ওরাল সেক্স (Oral sex) |
| 🍑 | Peach | Buttock | 🍌 | Banana | পুরুষ যৌনাঙ্গ (Penis) |

#### *3.1.2 Hashtag Segmentation*

Hashtags have become very common in social media posts, such as Facebook, Twitter, and Instagram. A user usually uses hashtags to categorize or organize posts such that a quick link can be used to find all related posts. When other users click on the hashtag, all of the same tagged posts appear. Analyzing all the hashtag messages will also be quite successful in identifying offensive content.

#### *3.1.3 Miscellaneous Text Processing*

The miscellaneous text preprocessing used in our proposed system includes number-to-word conversion, punctuation removal, white space removal, accent mark exclusion, stop word exclusion, etc.

There is no such list of stop words in Bengali, as there is in English. The list of stop words was completely changeable based on a particular task. For example, few stop words are - "বশে, শুধু, কত, করার, করে, করি, দওেয়ার, হচ্ছে, উপরে, সহতি, দছিে, দুটো, যান, দখো, স্পষ্ট, উপর, দুটি, দলিনে, দয়িছে" For offensive text analysis, which words will be stop words or not are detected manually and removed punctuation and unwanted words from the text using the Regex and Python libraries.

**Table 2:** Sample texts in the combined dataset [16, 33]

| Example | Transliteration (Bengali) | Transliteration (English) | Data Label |
|---|---|---|---|
| Do you get the feeling he is kissing behind🍑 so he can humiliate him later? | আপনি কি অনুভব করছেন যে তিনি (নিতম্বের) পিছনে চুম্বন করছেন যাতে তিনি পরে তাকে অপমান করতে পারেন | Do you get the feeling he is kissing behind buttock so he can humiliate him later? | Offensive |
| শালা একটা ঘুষখোর, এখন ঘুষের বিরুদ্ধে বড় বড় ভাষণ দিচ্ছে। তাকে জুতাপেটা করা দরকার। | সালা আস্ত শালা একটা ঘুষখোর এখন ঘুষের বিরুদ্ধে বড় বড় ভাষণ দিচ্ছে তাকে জুতাপেটা করা দরকার | Shala is a bribe-taker, now giving big speeches against bribery. He needs to be beaten by the shoe. | Offensive |
| জনগণের ভোটে নেতা হয়েছে, জনগণের সেবা কর | জনগণের ভোটে নেতা হয়েছে, জনগণের সেবা কর | You have become the leader of the people's vote; you should serve the people | Non-offensive |

*3.1.4 Translation*

The goal of the translation process is to translate multilingual comments into a single world language. In the first corpus, the Google Translator application programming interface (API) was used to convert English comments into Bengali. An expert then manually examined all translated comments for more accuracy. Bengali and translated comments were then combined. Next, BanglaBERT [34] Tokenizer was used to process Bengali texts. The same API was used to translate Bengali comments into English for the second corpus and verified by humans. The English and translated comments were combined. Subsequently, all comments were sent via the BERT Tokenizer. Tab. 2 presents some examples of sample datasets, their actual meanings, and the classification of each sentence.

### *3.2 Feature Engineering*

Feature engineering of textual data is also known as vectorization, where words within at text document are encoded as binary numbers of numeric or floating-point vectors. In this study, Word2Vec [35], TF-IDF [36], and BERT [37] feature extraction methods were used on the textual datasets.

*3.2.1 Word2Vec*

The Word2Vec algorithm uses a two-layer neural network model to learn word associations from larger datasets. After training the model, Word2Vec can recognize a word's context in a document, its semantic and structural similarities, and its relationships with other words. For each distinct word, Word2Vec generates a feature vector in the text corpus. The Python scikit-learn library was used to implement the feature vectors.

*3.2.2 TF-IDF Vectorizer*

When working with a large text dataset, some terms are frequently observed, but these terms do not have sufficient information. During text processing, these data often outperform other significant data frequencies, such as calculations. This problem can be solved using the TF-IDF feature extraction method. This can be defined as follows:

$$tf\_idf(t,\ d)\ := tf(t,\ d) \times idf(t) \tag{1}$$

where, $tf(t, d)$ is the term frequency, and $idf(t)$ the inverse document frequency.

### 3.2.3 BERT

In the bidirectional encoder representations from transformers (BERT) model, a multilevel bidirectional transformer encoder was used to produce the BERT transformer. Deep-learning model-based transformers are utilized as encoders and decoders for translation.

Fig. 2 depicts a typical BERT transformer, where $E_1$, $E_2$, …, $E_n$ represent the inputs of the BERT model. A set of tokens, special symbols, sentences, and other data can be used as inputs. After reaching the input level, there were several multilayer transformers. These bidirectional transformers were used to encode the input text and produce similar output vectors.

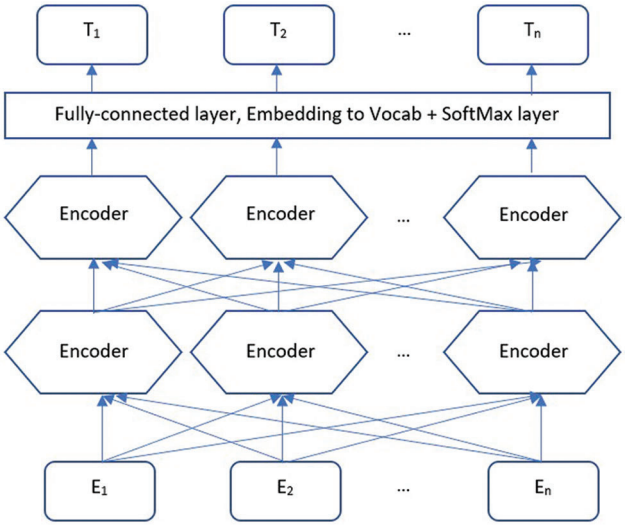



**Figure 2:** Typical pre-trained bi-directional BERT model [3]

BERT, developed at Google, was designed to pre-train deep bidirectional representations with support for token embedding, segment embedding, and several position-embedding features. Two types of position embedding exist: absolute position embedding (AE) and relative position embedding (RE). In AE, each point is mapped to the representation space elements, and in RE, the pointing gap between words in a phrase is transferred to the embedding space (i.e., $x - y$ for $x, y \in N$, where $N$ is the set of all word positions).

The weighted value ($W_v$), weighted query ($W_q$), and weighted key ($W_k$) in the attention head are used by the BERT transformer to compute the attention. Let $x \in N$ and $y \in N$ be two locations, where $WV_x$ is being the word vector for the x positional word, Ex is the embedding for the x position, and $E_{x-y}$ is being the relative position embedding. Then, given the x positional word, the q, v, and k vectors can be computed as follows:

$$AE: \begin{bmatrix} q_x \\ k_x \\ v_x \end{bmatrix} = (WV_x + E_x) \times \begin{bmatrix} W_q \\ W_k \\ W_v \end{bmatrix} \tag{2}$$

$$RE: \begin{bmatrix} q_x \\ k_x \\ v_x \end{bmatrix} = (WV_x + E_x) \times \begin{bmatrix} W_q \\ W_k \\ W_v \end{bmatrix} + \begin{bmatrix} 0 \\ E_{x-y} \\ E_{x-y} \end{bmatrix} \tag{3}$$

The sum of all attention(a) head values is the final result where the attention weight depends on $a = qk^T$, therefore,

$$Attention(q,\ k,\ v)\ := softmax\left(qk^T / \sqrt{d_k}\right)v \tag{4}$$

The TF-IDF algorithm determines the relevance of a word in a text and calculates its score, whereas Word2Vec combines the senses of all words into a single vector. Word2Vec does not operate outside the wordbook, where BERT can solve the Word2Vec limitation, and BERT finds word vectors using attention-based positional encoding. For English features, the BERT tokenizer was utilized, whereas for multilingual feature vectors, mBERT [27] was employed. The Bangla BERT-based model [34], a pretrained language model built with BERT-based mask language modeling, was used to analyze the Bengali text in this study.

### *3.3 Fine-Tuning of BERT*

Each encoder in the BERT model contained 12 transformer blocks, 768 hidden size representations, and 12 self-focusing heads. In each input sequence, BERT can handle up to 512 tokens in the input text. Therefore, one or more segments were examined for each sequence. In the BERT model, the beginning of a text sentence is indicated by the [CLS] (stands for class) token, and the [SEP] (stands for separator) token is used to indicate the ending of a text sentence. To identify specific meanings from a sequence during text categorization, BERT evaluates the last hidden state h of a sequence, beginning with [CLS]. Some of the criteria used in fine-tuning techniques for capturing multiple syntactic and semantic layers for classifying offensive text are described below.

#### *3.3.1 Long Text Pre-Processing*

The highest sequence length supported by BERT is 512. There are two different ways to process the highest-length texts. The first approach is the cutting method, in which the BERT model selects 512 tokens either from the beginning part of the text or end of the text. Another way of cutting approach is to select 256 tokens from both the beginning and ending parts of a text to make 512 tokens. Generally, the beginning and ending parts of a text document contain key points of content. Another approach used in our proposed model is the hierarchical approach, in which longer texts are divided into subtexts of 512 tokens and each token is applied to the BERT model. Finally, all subtexts were merged to find a representation of the full text.

#### *3.3.2 The Most Effective Layer Selection*

Individual layers were used in the BERT model to capture different features during the classification of offensive text. We have observed the utility of features from different levels. The model was fine-tuned based on the test error rates of a particular layer's performance. Tab. 3 shows the test error rates of different layers, where the first-level performance is much worse than the last level.

#### *3.3.3 Handle Overfitting Problem*

Distinct learning rates are applied during fine-tuning to reduce overfitting. During the 2.5e-5 learning rate, lower layers created test error rates from 5.58 to 6.00 and when learning 2.0e-5, their error rates range from 5.95 6.25. From the experiments, we found that during BERT fine-tuning, a lower learning rate ($lr < 2.5e-5$) works well in the bottom layer.

**Table 3:** Test error rates of different BERT layers on multilingual-based dataset

| Layer name | Error rates (%) | Layer name | Error rates (%) |
|---|---|---|---|
| Layer0 | 12.52 | Layer6 | 7.20 |
| Layer1 | 12.01 | Layer7 | 6.92 |
| Layer2 | 10.43 | Layer8 | 6.46 |
| Layer3 | 9.82 | Layer9 | 5.87 |
| Layer4 | 8.86 | Layer10 | 5.65 |
| Layer5 | 7.69 | Layer11 | 5.58 |

### *3.4 Proposed Deep-BERT Model*

In this study, different types of BERT transformers were used as sentence encoders because they can accurately obtain the relevant reflection of a statement. Our model consists of three concurrent frames of a 1D-CNN with BERT that have different kernel values and filters, each with a maximum pooling layer. The texts were analyzed using different CNNs by changing the kernel size, filters, and numerous hidden layers. A graphical design of the proposed Deep-BERT model is shown in Fig. 3.

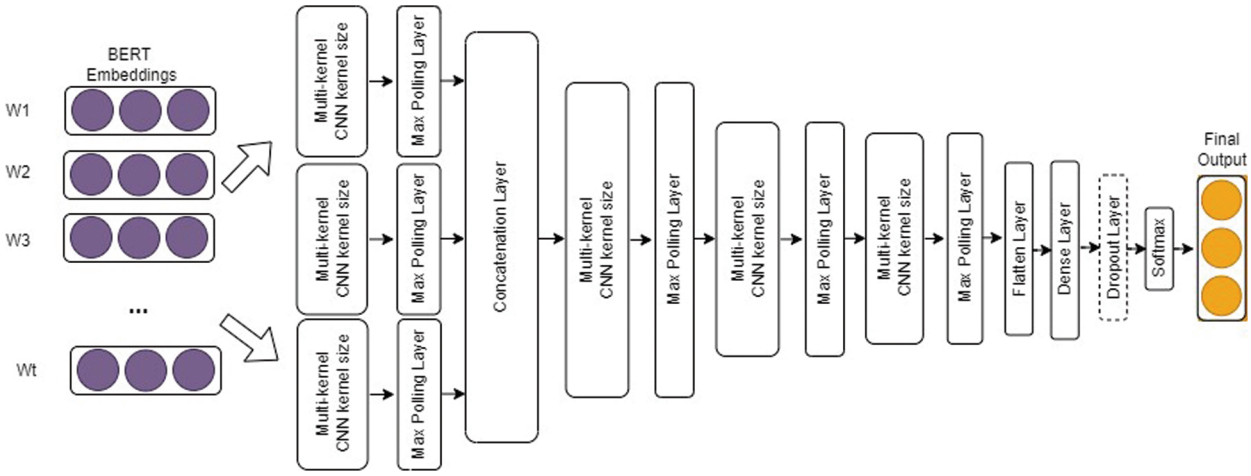



**Figure 3:** The architecture of proposed Deep-BERT model

The BERT fine-tuning layer consists of different types of pre-trained BERT versions, such as the uncased BERT base for English, banglaBERT base [34] for Bengali, and mBERT [27] for multilingual texts. We used specific BERT transformers for encoding specific languages, and each transformer had 12 levels, 768 hidden units, and 110 million parameters.

The convoluted layer is composed of a collection of filters and kernels that work together to improve the semantic relationship between words of varying lengths. We allocated three convolution layers in parallel, where each block had a 1D-CNN layer with different kernel sizes and filters. In our proposed model, we used a total of six convolution layers where three (3) layers were used in parallel after the BERT embedding layer and the other three (3) layers were used just after the concatenation layer. In this Deep-BERT model, the kernel sizes of the first three convolution layers are 3, 4, and 5, respectively, and the kernel size of the next three (3) convocation layers is 5, with 128 filters in each convolution layer.

The max-pooling layer was used immediately after each convolution layer to reduce the size of the required computing activity in the system after the convolution layer output. The purpose is to gradually shrink the presentation spatial dimension to reduce the number of parameters and calculations in the network. We assigned six (6) max-pooling layers to our proposed model, where a kernel size of five was used for each layer. There is also a flattened layer between the convolution and fully connected layers that converts the 2D matrix into a vector format for future use. The CNN outputs are processed through a dense layer with dropout, and then a Softmax layer.

## 4 Result Analysis

The main objective of this study is to test the effectiveness of the Deep-BERT model by applying different BERT models to different datasets. The tests were classified into two categories. The first section deals with monolingual-based evaluation and is divided into two parts: it examines both the BERT and BanglaBERT transformer models. The first part of the test is to translate all the Bengali comments into English, combine the translated dataset with the original English dataset, and apply an uncased BERT-based transformer with our deep-BERT offensive text classification model. In the second part, we translate all the English comments in Bengali using the same method, combine the Bengali datasets with the translated Bangla texts, and apply the BanglaBERT-based transformer with the proposed model. The next section is a multilingual-based assessment, where we combine all Bengali and English datasets into a single form and then use multilingual BERT (mBERT) to extract useful features from the combined text of our Deep-BERT model. To conduct our studies, we utilized Ubuntu 18.04 operating system on an Intel (R) Core (TM) i5-6500 CPU with 16GB of RAM. TensorFlow 2.2.1, with Python 3.6.9, was used to develop all offensive text categorization models; Scikit learn 0.22.2 and Panda 1.0.3 Data Frame was used to generate data sets for training and testing. Datasets containing all offensive and general texts were randomly shuffled throughout the training and testing processes to ensure that the training and test datasets included a combination of offensive and non-offensive content. We built the model described in Section 3-D and the Tab. 4 contains the list of the parameters and hyperparameters utilized in this model.

**Table 4:** List of parameters and hyper parameters

| Name | Value | Name | Value |
|---|---|---|---|
| seq_len | 512 | patience | 0 |
| feed_forward_dim | 3072 | verbose | 1 |
| batch_size | 16 | epochs | 10 |
| transformer_num | 12 | dropout_rate | 0.5 |
| head_num | 12 | pos_num | 512 |

### *4.1 Dataset Description*

No standardized corpus is available for evaluating multilingual offensive texts. We chose Bengali and English to construct a multilingual dataset. We used the SOLID English dataset of nine million offensive tweets created by Rosenthal et al. [34] in the SemEval2020 contest. There are three categories of text: (1) offensive language detection, (2) categorization of offensive language, and (3) offensive language target identification.

Among these nine million datasets, we used 7000 English tweets in this study, which were divided into offensive and non-offensive categories. In this paper, 6500 comments from the datasets created by Karim

et al. [16] were used to generate the Bengali dataset. The authors divided their dataset into five different groups: (1) gender abuse, (2) religious, (3) personal, (4) political, and (5) geopolitical, with each group containing offensive and non-offensive texts. We combined all groups and used only offensive and non-offensive text for the Bengali dataset. The multilingual data collection statistics are presented in Tab. 5. The dataset contains 7000 English comments and 6500 Bengali comments, which are divided into two sections. In the combined Bengali and English datasets, there were 5085 offensive comments and 8415 non-offensive comments. Among all the datasets, 20% were set aside for testing, with the remaining 80% used for training purposes.

**Table 5:** Dataset description

| Attributes name | English datasets | Bengali datasets |
|---|---|---|
| Number of comments | 7000 | 6500 |
| Offensive comments | 3193 | 1892 |
| Non-offensive comments | 3807 | 4608 |
| Largest text length | 1560 | 2354 |
| Smallest text length | 5 | 11 |

### *4.2 Performance Analysis Parameters*

The performance of the proposed model was evaluated based on accuracy, precision, recall, and F1-score.

#### *4.2.1 Accuracy*

Classification accuracy is a metric used to measure how often a machine-learning system accurately classifies a data item. Accuracy measures the number of data points that are properly predicted from all data points. It is computed as the ratio of accurate forecasts to total predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

where, TP= Ture Positive; FP= False Positive; TN= True Negative; FN= False Negative.

#### *4.2.2 Precision*

Precision is the ratio of true positive predictions (offensive text) to the total number of texts (both offensive and non-offensive).

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

#### *4.2.3 Recall*

The ratio of accurately predicted true positives to all experimental data (true positives and false negatives) in the class (offensive and non-offensive texts) is the recall in this context.

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

#### 4.2.4 F1-score

The weighted mean of accuracy and recall is represented by the F1-score. As a result, this score takes into account both false positives and negatives.

$$F1_{Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

### 4.3 Monolingual-based Results

To conduct the experiments, a number of text classification algorithms and various feature extraction methods were applied to predict offensive text. In our experiment, we used multinomial naive Bayes (MNB), k-nearest neighbors (KNN), logistic regression (LR), stochastic gradient descent (SGD), support vector machine (SVM), decision tree (DT), random forest (RF), LSTM [38], and CNN [39] text classification methods using a combination of TF-IDF, Word2Vec, and BERT feature extraction models.

#### 4.3.1 Experiments on English Dataset

In Tab. 6, we present various classification results for the English dataset. The proposed Deep-BERT yielded a maximum accuracy of 93.11%, and the minimum accuracy result was 65.21% for the MNB classification with the TF-IDF feature extraction model. Additionally, the MNB and KNN models performed less accurately than the others for the TF-IDF and Word2Vec models. This discovery demonstrates the crucial significance of identifying the interconnections between various word combinations that are more achievable in BERT than in the other models.

**Table 6:** The Monolingual-based accuracy scores of different classifiers

| Accuracy scores on english dataset [33] | | Accuracy scores on bengali dataset [16] | |
|---|---|---|---|
| Classification model | Accuracy (%) | Classification model | Accuracy (%) |
| TF-IDF + MNB | 65.21 | TF-IDF + MNB | 63.12 |
| TF-IDF + KNN | 67.23 | TF-IDF + KNN | 68.22 |
| TF-IDF + RF | 66.34 | TF-IDF + RF | 69.67 |
| TF-IDF + DT | 70.87 | TF-IDF + DT | 74.85 |
| Word2Vec + MNB | 67.64 | Word2Vec + MNB | 67.76 |
| Word2Vec + DT | 69.34 | Word2Vec + DT | 69.34 |
| Word2Vec + KNN | 65.61 | Word2Vec + KNN | 68.62 |
| Word2Vec + RF | 66.45 | Word2Vec + RF | 65.55 |
| Word2Vec + LSTM | 67.12 | Word2Vec + LSTM | 78.23 |
| Word2Vec + CNN | 75.34 | Word2Vec + CNN | 75.34 |
| BERT + MNB | 85.34 | BanglaBERT + MNB | 85.37 |
| BERT + LR | 83.87 | BanglaBERT + LR | 86.46 |
| BERT + KNN | 85.88 | BanglaBERT + KNN | 86.84 |
| BERT + SGD | 84.58 | BanglaBERT + SGD | 85.34 |
| BERT + SVM | 87.34 | BanglaBERT + SVM | 88.78 |
| BERT + DT | 87.51 | BanglaBERT + DT | 87.65 |

(Continued)

**Table 6 (continued)**

| Accuracy scores on english dataset [33] | | Accuracy scores on bengali dataset [16] | |
|---|---|---|---|
| Classification model | Accuracy (%) | Classification model | Accuracy (%) |
| BERT + RF | 86.67 | BanglaBERT + RF | 87.67 |
| BERT + LSTM | 88.22 | BanglaBERT + LSTM | 89.46 |
| BERT + CNN | 92.34 | BanglaBERT + CNN | 90.23 |
| RoBERTa | 89.80 | DeepHateExplainer | 87.67 |
| Deep-BERT | 93.11 | Deep-BERT | 92.45 |

#### *4.3.2 Experiments on Bengali Dataset*

In Tab. 6, we present various classification results of the Bengali dataset, including the proposed Deep-BERT, which utilizes the BanglaBERT [34] feature extraction approach. Among the three feature extraction approaches, BanglaBERT outperformed the other methods. The decision tree (DT) classification algorithm with TF-IDF feature extraction showed an accuracy score of 74.85%, which is the highest score among all core machine learning classification algorithms during TF-IDF and Word2Vec fracture extraction. We compared our model with the existing DeepHateExplainer [16] model, where the proposed model performed better than the others and achieved an accuracy of 92.45%.

### *4.4 Multilingual-based Results*

In this subsection, we apply various machine learning classifiers to multilingual-based datasets. We used a multilingual BERT (mBERT) transformer-based feature extraction model to determine the bidirectional and semantic relationships between the texts. Tab. 7 shows the experimental results of different classification models, where our proposed Deep-BERT classification model achieves the highest results in both accuracy (91.83%) and f1-score (92.93%) compared to all other models. mBERT with the CNN model shows an accuracy score of 90.79% %because the CNN model has a default size of the max-pooling layer with convolution, fixed filters, and hidden layers. However, in the Deep-BERT model, we used different filters and hidden layers in our proposed model, which increased the accuracy more than the CNN model alone.

**Table 7:** Classification results on multilingual-based dataset

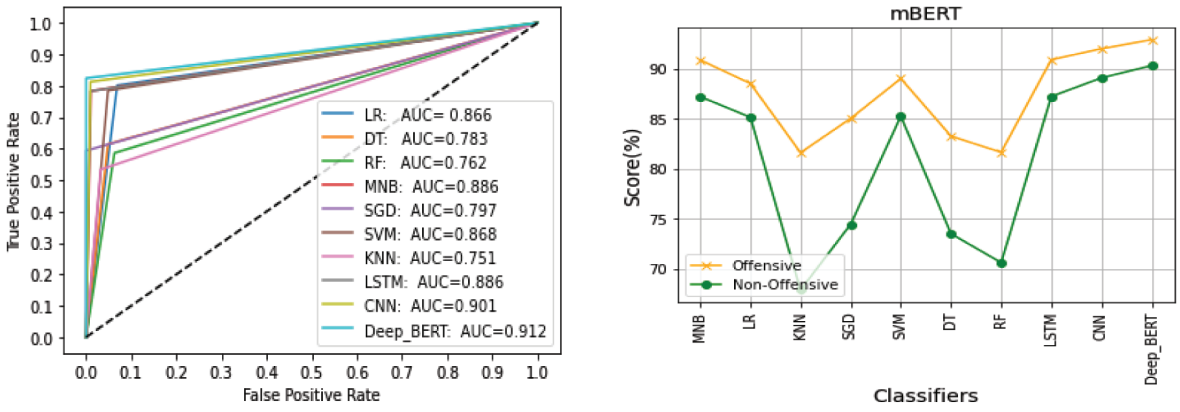
| Classification model | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|
| TF-IDF + MNB | 82.76 | 79.4 | 91.33 | 84.95 |
| TF-IDF + KNN | 78.91 | 75.26 | 90 | 81.97 |
| TF-IDF + DT | 77.08 | 74.14 | 87.49 | 80.26 |
| TF-IDF + RF | 79.68 | 75.86 | 90.72 | 82.62 |
| Word2Vec + MNB | 73.90 | 68.14 | 95.36 | 79.48 |
| Word2Vec + DT | 64.09 | 62.38 | 80.60 | 74.70 |
| Word2Vec + KNN | 68.03 | 59.62 | 83.04 | 76.83 |
| Word2Vec + RF | 77.12 | 70.54 | 97.63 | 81.90 |
| Word2Vec + LSTM | 71.16 | 64.76 | 93.45 | 78.61 |

(Continued)

**Table 7 (continued)**

| Classification model | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|
| Word2Vec + CNN | 77.49 | 70.20 | 90.41 | 82.49 |
| mBERT + MNB | 89.37 | 84.08 | 98.93 | 90.90 |
| mBERT + LR | 87.08 | 84.40 | 93.13 | 88.55 |
| mBERT + KNN | 76.65 | 70.59 | 96.79 | 81.64 |
| mBERT + SGD | 81.15 | 74.00 | 98.90 | 85.06 |
| mBERT + SVM | 87.44 | 83.66 | 95.16 | 89.04 |
| mBERT + DT | 79.51 | 74.05 | 95.16 | 83.29 |
| mBERT + RF | 77.44 | 72.37 | 93.74 | 81.68 |
| mBERT + LSTM | 89.40 | 84.05 | 99.03 | 90.93 |
| mBERT + CNN | 90.79 | 85.95 | 99.03 | 92.03 |
| Deep-BERT | 91.83 | 86.79 | 99.50 | 92.93 |

Receiver operating characteristic (ROC) arcs are frequently used to illustrate the relationship between sensitivity and specificity for a proposed cut-off of an experiment or experiments. The ROC curve study on multilingual-based datasets using the mBERT feature-extraction approach is shown in Fig. 4. The proposed model achieved a maximum area under the ROC curve (AUC) value of 91.2%, whereas the CNN classifiers showed an AUC value of 90.1%, which was the highest value among all the baseline classification algorithms, and the lowest AUC score was 75.1% for the KNN classification algorithm.




(a) ROC Curve

(b) F1-Score

**Figure 4:** Receiver operating characteristic (ROC) curve and F1-Score of different classifiers on multilingual dataset

The F1-score is a harmonic measure of a model's accuracy and recall, which is used to evaluate it. The accuracy evaluation measure is used to accurately identify numbers; however, it does not account for false positives or negatives and produces inaccurate results when data are not uniformly distributed. The F1-score considers all the constraints of the accuracy measure, making it more powerful than the machine-learning accuracy evaluation metric. Fig. 4 shows the F1-score of the offensive and non-offensive classes, where

the KNN classifier f1 score is very low for the non-offensive class, and the F1-score of the RF classifier is the lowest for the offensive class.

## 5 Limitation and Future Work

At present, this research considers only two languages Bengali and English to check the performance of the proposed architecture. This research can further be easily extended for other widely used languages such as Spanish, Chinese, Hindi, etc. which is the future work of this research. Furthermore, the proposed architecture can be tested using with larger datasets in the future.

## 6 Conclusion

In this study, we proposed a transfer-learning-based DCNN model called Deep-BERT, which is the first deep learning-based BERT model for multilingual offensive text classification. The novelty of this model is that it can classify both monolingual and multilingual-based offensive text. In our proposed model, we added a classification layer on top of the encoder output, multiplied the output sequence by the embedding matrix, and finally used the SoftMax function to determine the likelihood of each vector. We used three concurrent frames of a 1D-convolutional neural network with BERT with different kernel values and filters, each with a maximum-pooling layer. The text was analyzed using various CNNs by changing the kernel size, filters, and numerous hidden layers. In this study, the authors evaluated translation-based and multilingual-based tests on different datasets and compared the results with existing models. The proposed model outperformed all other methods in both translation -and multilingual-based offensive text analyses.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest regarding this study.

## References

[1] N. Bel, C. H. Koster and M. Villegas, "Cross-lingual text categorization," in *Int. Conf. on Theory and Practice of Digital Libraries*, Springer, Berlin, Heidelberg, pp. 126–139, 2003.

[2] M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov *et al.,* "Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020)," arXiv Preprint arXiv:2006.07235, pp. 1425–1447, 2020.

[3] M. F. Mridha, M. A. H. Wadud, M. A. Hamid, M. M. Monowar, M. Abdullah-Al-Wadud *et al.,* "L-Boost: Identifying offensive texts from social media post in bengali," *IEEE Access*, vol. 9, pp. 164681–164699, 2021.

[4] M. R. Amini, C. Goutte and N. Usunier, "Combining coregularization and consensus-based self-training for multilingual text categorization," in *Proc. of the 33rd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Geneva, Switzerland, pp. 475–482, 2010.

[5] M. A. Bentaallah and M. Malki, "The use of wordnets for multilingual text categorization: A comparative study." in *Int. Conf. on Web and Information Technologies*, Sidi Bel Abbes, Algeria, pp. 121–128, 2012.

[6] F. E. Alami, S. O. E. Alaoui and N. E. Nahnahi, "A multilingual offensive language detection method based on transfer learning from transformer fine-tuning model," *Journal of King Saud University - Computer and Information Sciences*, pp. 1–9, 2021.

[7] S. H. Yadav and B. L. Parne, "A survey on different text categorization techniques for text filtration," in *2015 IEEE 9th Int. Conf. on Intelligent Systems and Control (ISCO)*, Coimbatore, India, pp. 1–5, 2015.

[8] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad and Y. Chang, "Abusive language detection in online user content," in *Proc. of the 25th Int. Conf. on World Wide Web*, Montreal, Canada, pp. 145–153, 2016.

[9] S. H. Yadav and P. M. Manwatkar, "An approach for offensive text detection and prevention in social networks," in *2015 Int. Conf. on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, pp. 1–4, 2015.

[10] T. Chu, K. Jue and M. Wang, "Comment abuse classification with deep learning," Von https://web.stanford.edu/class/cs224n/reports/2762092.pdf.abgerufen, 2016.

[11] E. Wulczyn, N. Thain and L. Dixon, "Ex machina: Personal attacks seen at scale," in *Proc. of the 26th Int. Conf. on World Wide Web*, Perth, Australia, pp. 1391–1399, 2017.

[12] A. Shah, X. Yan, S. Shah and G. Mamirkulova, "Mining patient opinion to evaluate the service quality in healthcare: A deep-learning approach," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 2925–2942, 2020.

[13] A. Shah, X. Yan, S. Khan, W. Khurrum and Q. Khan, "A Multi-modal approach to predict the strength of doctor-patient relationships," *Multimedia Tools and Applications*, vol. 80, pp. 23207–23240, 2021.

[14] M. Ishmam and S. Sharmin, "Hateful speech detection in public facebook pages for the bengali language," in *2019 18th IEEE Int. Conf. on Machine Learning and Applications (ICMLA)*, Boca Raton, FL, USA, pp. 555–560, 2019.

[15] S. C. Eshan and M. S. Hasan, "An application of machine learning to detect abusive bengali text," in *2017 20th Int. Conf. of Computer and Information Technology (ICCIT)*, Dhaka, Bangladesh, pp. 1–6, 2017.

[16] M. R. Karim, S. K. Dey, T. Islam, S. Sarker, M. H. Menon *et al.,* "Deephateexplainer: Explainable hate speech detection in under-resourced bengali language," in *2021 IEEE 8th Int. Conf. on Data Science and Advanced Analytics (DSAA)*, Porto, Portugal, pp. 1–10, 2021.

[17] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi *et al.,* "Roberta: A robustly optimized bert pretraining approach," arXiv preprint arXiv:1907.11692, 2019.

[18] M. M. Rahman, M. A. Pramanik, R. Sadik, M. Roy and P. Chakraborty, "Bangla documents classification using transformer based deep learning models," in *2020 2nd Int. Conf. on Sustainable Technologies for Industry 4.0 (STI)*, Dhaka, Bangladesh, pp. 1–5, 2020.

[19] Y. Song, J. Wang, Z. Liang, Z. Liu and T. Jiang, "Utilizing bert intermediate layers for aspect-based sentiment analysis and natural language inference," arXiv preprint arXiv:2002.04815, 2020.

[20] A. Cohan, I. Beltagy, D. King, B. Dalvi and D. S. Weld, "Pretrained language models for sequential sentence classification," arXiv preprint arXiv:1909.04054, 2019.

[21] M. G. Hussain, T. A. Mahmud and W. Akthar, "An approach to detect abusive bangla text," in *2018 Int. Conf. on Innovation in Engineering and Technology (ICIET)*, Dhaka, Bangladesh, pp. 1–5, 2018.

[22] S. L. Granizo, Á. L. V. Caraguay, L. I. B. López and M. Hernández-Álvarez, "Detection of possible illicit messages using natural language processing and computer vision on twitter and linked websites," *IEEE Access*, vol. 8, pp. 44534–44546, 2020.

[23] T. Ranasinghe and M. Zampieri, "Multilingual offensive language identification with cross-lingual embeddings," arXiv preprint arXiv:2010.05324, 2020.

[24] C. H. Lee, H. C. Yang and S. M. Ma, "A novel multilingual text categorization system using latent semantic indexing," in *First Int. Conf. on Innovative Computing, Information and Control-Volume I (ICICIC'06)*, Beijing, China, vol. 2. pp. 503–506, 2006.

[25] S. Mittal and P. Dhyani, "Multilingual text classification," *International Journal of Engineering Research & Technology*, vol. 4, no. 3, pp. 99–101, 2015.

[26] K. Rani and Satvika, "Text categorization on multiple languages based on classification technique," *International Journal of Computer Science and Information Technologies*, vol. 7, no. 3, pp. 1578–1581, 2016.

[27] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*., Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186, Jun. 2019.

[28] G. Lample and A. Conneau, "Cross-lingual language model pretraining," arXiv preprint arXiv:1901.07291, 2019.

[29] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek *et al.,* "Unsupervised cross-lingual representation learning at scale," arXiv preprint arXiv:1911.02116, 2019.

[30] S. Thomson, E. Kluftinger and J. Wentland, "Are you fluent in sexual emoji?: Exploring the use of emoji in romantic and sexual contexts," *The Canadian Journal of Human Sexuality*, vol. 27, no. 3, pp. 226–234, 2018.

[31] The Unicode Consortium, "The world standard for text and emoji, 2021," Accessed on: December 08, 2021. [Online]. Available: https://home.unicode.org.

[32] Python Software Foundation, "PyPl the python package index, 2021," Accessed on: December 08, 2021. [Online]. Available: https://pypi.org/.

[33] S. Rosenthal, P. Atanasova, G. Karadzhov, M. Zampieri and P. Nakov, "A Large-scale semi-supervised dataset for offensive language identification," arXiv preprint arXiv:2004.14454, 2020.

[34] S. Sarker, "BanglaBERT: Bengali mask language model for bengali language understading," 2020. [Online]. Available: https://github.com/sagorbrur/bangla-bert.

[35] M. A. H. Wadud and M. R. H. Rakib, "Text coherence analysis based on misspelling oblivious word embeddings and deep neural network," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 1, pp. 194–203, 2021.

[36] G. Jie and C. Li-chao, "Research of improved if-idf weighting algorithm," in *the 2nd Int. Conf. on Information Science and Engineering*, Hangzhou, China, pp. 2304–2307, 2010.

[37] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," arXiv preprint arXiv:1908.10084, 2019.

[38] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[39] W. Yin, K. Kann, M. Yu and H. Schütze, "Comparative study of CNN and RNN for natural language processing," arXiv Preprint arXiv:1702.01923, pp. 1–7, 2017.