IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

TEmoX: Classification of Textual Emotion using Ensemble of Transformers

AVISHEK DAS[1] *Graduate Member, IEEE*, **MOHAMMED MOSHIUL HOQUE**[1] *SENIOR MEMBER, IEEE*, **OMAR SHARIF**[1] *GRADUATE MEMBER, IEEE*, **M. ALI AKBER DEWAN**[2] *MEMBER, IEEE* **AND NAZMUL SIDDIQUE**[3], *Senior Member, IEEE*

[1]Dept. of Computer Science and Engineering, Chittagong University of Engineering & Technology, Chittagong, Bangladesh (avishek.das.ayan@gmail.com; moshiul_240@cuet.ac.bd; omar.sharif@cuet.ac.bd)

[2]School of Computing and Information Systems, Faculty of Science and Technology, Athabasca University, Athabasca, AB T9S 3A3, Canada (adewan@athabascau.ca)

[3]School of Computing, Engineering and Intelligent Systems, Ulster University, UK (nh.siddique@ulster.ac.uk)

Corresponding author: Mohammed Moshiul Hoque (moshiul_240@cuet.ac.bd).

ABSTRACT Textual emotion classification (TxtEC) refers to the classification of emotion expressed by individuals in textual form. The widespread use of the Internet and numerous Web 2.0 applications has emerged in an expeditious growth of textual interactions. However, determining emotion from texts is challenging due to their unorganized, unstructured, and disordered forms. While research in textual emotion classification has made considerable breakthroughs for high-resource languages, it is yet challenging for low-resource languages like Bengali. This work presents a transformer-based ensemble approach (called *TEmoX*) to categorize Bengali textual data into six integral emotions: joy, anger, disgust, fear, sadness, and surprise. This research investigates 38 classifier models developed using four machine learning LR, RF, MNB, SVM, three deep-learning CNN, BiLSTM, CNN+BiLSTM, five transformer-based m-BERT, XLM-R, Bangla-BERT-1, Bangla-BERT-2, and Indic-DistilBERT techniques with two ensemble strategies and three embedding techniques. The developed models are trained, tuned, and tested on the three versions of the Bengali emotion text corpus BEmoC-v1, BEmoC-v2, BEmoC-v3. The experimental outcomes reveal that the weighted ensemble of four transformer models En-22: Bangla-BERT-2, XLM-R, Indic-DistilBERT, Bangla-BERT-1 outperforms the baseline models and existing methods by providing the maximum weighted $F1$-score (80.24%) on BEmoC-v3. The dataset, models, and fractions of codes are available at https://github.com/avishek-018/TEmoX.

INDEX TERMS Natural language processing, Text classification, Textual emotion classification, Bengali emotion text corpus, Ensemble of transformers.

I. INTRODUCTION

CLassifying textual emotion entails the automated process of attributing a text to an emotion category based on predetermined connotations. In recent years, the proliferation of the Internet and the rapid evolution of social media platforms have led to a significant surge in text-based content, greatly increasing its presence in everyday interactions. Online users communicate their concerns, opinions, or feelings via tweets, posts, and comments. Thus, much emotional text content is accessible on social media or online platforms. Researchers' attention has been attracted by the increasing volume of textual emotion content, as the categorization of emotions plays a vital role across numerous applications, including education, sports, e-commerce, healthcare, and amusement. With an ever-increasing number of people on virtual platforms and the rapidly producing online information, evaluating the emotions expressed in online content becomes crucial for different stakeholders, such as customers, enterprises, and online education. Textual emotion analysis of an enterprises service or product can boost brand value, sales, and prestige [1]. Automatic TxtEC helps to enhance the quality of a product or service, revise sales plans, and forecast forthcoming trends. Furthermore, it can shape brand reputation, follow client reactions,

catch general emotions, and track conformity.

Although TxtEC has made significant advancements in well-resourced languages, its current stage of development remains rudimentary when it comes to low-resource languages like Bengali. Investigating huge amount of data to unveil the underlying sentiments or emotions (particularly in Bengali) is considered a critical research problem in low-resource languages. The textual data are voluminous and unstructured. Due to their chaotic forms, it is very arduous and time-intensive to organize, store, manipulate, and extract emotional content. The difficulty arises from several constraints, including sophisticated language structures, limited resources, and substantial verb inflections [2]. Moreover, the scarcity of text processing tools and standard corpora makes textual emotion analysis more difficult in Bengali. Taking into consideration of the current impediments of textual emotion classification in Bengali, this work introduces an intelligent technique called *TEmoX* which utilizes transformer-based learning to categorize Bengali texts into six primary emotions (e.g., joy, anger, disgust, fear, sadness, surprise). Transformer-based learning has recently demonstrated significant advancements in text classification [3]–[5]. Hence, this work motivates us to use transformer-based learning to classify textual emotions in Bengali. This research extends the previous work [3], which involved utilizing three transformer models: m-BERT, XLM-R, and Bangla-BERT-1. However, this work utilizes two more new models: Indic-DistilBERT [6] and Bangla Bert-2 [6]. In addition, this study incorporates the extended version of the dataset (BEmoC-v3 [7]) and employs a stratified sampling technique [8] to address the imbalanced nature of the corpus. By exploiting the performance of 26 ensemble models, 3 deep learning models, and 4 machine learning-based models, this work proposes the En22 (XLM-R+Bangla-BERT-1+Bangla-BERT-2+Indic-DistilBER) model to perform textual emotion classification for improved results. The distinctive contributions of this work are outlined as follows:

- Proposed a textual emotion classification technique called *TEmoX* to classify Bengali text into six categories: joy, anger, disgust, fear, sadness, and surprise. *TEmoX* uses weighted ensemble of four standard transformer models (XLM-R, Bangla-BERT-1, Bangla-BERT-2, and Indic-DistilBERT) with fine-tuned hyperparameters.
- Investigated 38 classification models, including 4 machine learning (ML), 3 deep learning (DL), and 5 transformer models with ensemble strategies to find a robust model for textual emotion classification tasks in Bengali.
- Analyzed the classification outcomes of 38 models with a detailed investigation of misclassification and error rate to find many exciting characteristics of the emotion classification task that might help future researchers.

II. RELATED WORK

Recent advancements in textual emotion classification tasks primarily concentrate on high-resourced languages owing to the obtainability of standard datasets and text-processing tools. Unfortunately, no formal data repository exists in resource-constraint languages, including Bengali, like IMDB dataset[1]. There is a substantial advancement in the textual emotion classification in English, Arabic, Chinese, French, and other high-resourced languages [9]. For example, the EmoTxt toolkit is created using ML algorithms for the English language [10]. In another research, random forest (RF), decision tree (DT), and K-nearest neighbor (KNN) are used to detect multilabel multi-target emotion text in Arabic tweets where RF provides the highest F1-score of 82.6% [11]. Ahmad et al. [12] suggested a DL model for categorizing English poetry text into 13 emotion classes. A recent study [13] actively explored seven ML techniques for classifying Tweets into happy or unhappy. The ensemble of logistic regression (LR) and stochastic gradient descent (LR-SGD) emerged with the highest accuracy of 79%. An automatic classification method was developed by Hasan et al. [14] for detecting emotion from tweets. They applied a supervised ML algorithm and obtained 90% accuracy by a decision tree in four emotion categories, such as happy-inactive, happy-active, unhappy-inactive, and unhappy-active.

Several DL approaches have been studied to classify textual emotion from short sentences. For the classification of emotions in Chinese microblogs, Lai et al. [15] presented a graph convolution network architecture, and their suggested method attained an F-score of 82.32%. Using Nested Long-Short Term Memory (LSTM), Haryadi et al. [16] successfully classified English Twitter data into seven emotion classes and yielded exceptional results, achieving the highest accuracy (99.167%). The SemEval-2019 task-3 [17] proposed a Bi-LSTM model for categorizing emotion into four classes and gained a maximum F1-score of 79.59%. Ameer et al. [18] presented a detailed analysis of classifying short text messages (i.e., SMS) using several ML, DL, and transfer learning-based techniques. Their models have been developed on a code-mixed (Urdu and English) dataset containing 12 emotion classes and achieved the maximum performance by ML model with uni-gram features. Kumar et al. [19] used a dual-channel method for multi-class textual emotion detection. They employed CNN to extract textual features and the BiLSTM layer to order text and sequence information. Their work revealed that multiple layers could give more accurate results. However, their network becomes

[1] http://www.imdb.com/

comparatively slower, and GloVe requires extra time than the BERT embeddings.

Although TxtEC in low-resource languages such as Bengali is still in its infancy, a few research activities have been embarked on utilizing ML and DL approaches. Among them, Tripto et al. [20] developed a method to detect multilabel sentiments and emotions from Romanized Bengali texts based on the YouTube comments dataset of 1006 data. Their model (LSTM) detected three-label sentiment (with 65.97% accuracy), five-label sentiment (54.24% accuracy), and emotion (59.23% accuracy). Rahib et al. [21] developed a DL-based method using CNN and LSTM to classify emotion from social media response on COVID-19 text and achieved an accuracy of 84.92%. Purba et al. [22] proposed an emotion detection system employing a Multinomial Naive Bayes classifier to identify emotions into three categories (angry, sad, and happy) with an accuracy of 68.27%. Mamun et al. [23] introduced a sentiment dataset comprising 8122 text expressions categorized into negative, positive, and neutral. They showed that the ensemble technique (LR+RF+SVM) surpassed the other approaches attaining the most increased accuracy of 82%. Rayhan et al. [24] developed an emotion dataset with six emotion classes by translating an existing English emotion dataset into Bengali. They applied CNN-BiLSTM and BiGRU on the dataset and attained the highest F1-score of 67.41% using CNN-BiLSTM. Azmin et al. [25] employed three emotive classes (happy, sad, and anger). They used a dataset developed by [26] and showed Multinomial Naïve Bayes (MNB) surpassed others with a precision of 78.6%. A corpus named *Anubhuti* [27] concentrated on Bengali short stories labeled in four classes (joy, anger, sorrow, and suspense) which obtained an accuracy of 73% by LR. Analyzing emotions expressed in Bengali blog writing, Das et al. [28] utilized conditional random field (CRF) for identifying emotional content from blogs, that achieved an accuracy of 56.45%. Rupesh et al. [29] classified six basic emotions on 1200 Bengali documents from different domains using SVM and obtained 73% accuracy. Rahman et al. [26] curated a Bengali emotion dataset focused on sociopolitical issues and employed ML techniques. Their work acquired the highest accuracy (52.98%) and F1-score (33.24%) by utilizing SVM with a non-linear RBF kernel. Parvin et al. [30] utilized the ensemble of CNN and RNN architectures on their developed emotion dataset (containing 9000 text data) and achieved an F1-score of 62.46%. Iqbal et al. [7] developed an emotion dataset called BEmoC-v3 containing 7000 textual data in Bengali. The previous version of BEmoC-v3 (i.e., BEmoC-v2) consisted of 6243 data utilized by Das et al. [3] for classifying six emotions in Bengali. They employed a pre-trained BERT variant XLM-Roberta and gained the maximum F1-score (69.73%).

Table 1 summarizes the findings of a few recent studies on Bengali TxtEC in terms of the number of classes, corpus size, models used, performance, and critical weaknesses.

Most previous works in Bengali TxtEC methods used limited datasets to develop ML and DL approaches. In contrast to past studies, this research proposes an ensemble of transformer-based learning that can detect six emotions, outperforming previous methods of TxtEC in Bengali. The use of transformer models made Bengali text classification tasks more robust [32], [33].

III. BEMOC-V3: BENGALI EMOTION CORPUS

The development of an intelligent method for TxtEC in resource-constrained languages presents a significant challenge due to the lack of benchmark corpora. Thus, developing a reliable corpus is the prerequisite for any intelligent text classification model based on ML or DL techniques. The previous research [7] discussed various aspects of the development of the dataset (BEmoC-v3). This work focuses on the various analysis of the dataset. The Bengali Emotion Corpus ('BEmoC-v3'), is freely available at https://github.com/avishek-018/TEmoX.

A. BEMOC-V3 DEVELOPMENT

BEmoC-v3 comprises four sub-modules: data crawling, preprocessing, annotation, and verification. Fig. 1 illustrates the overall process of BEmoC-v3 development.

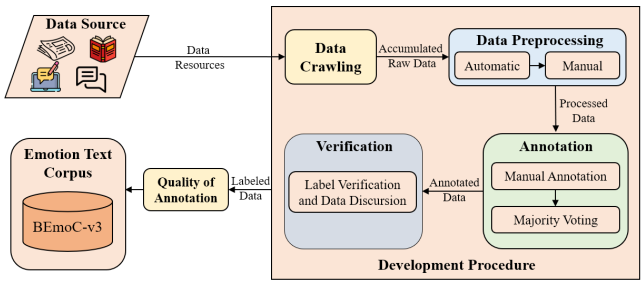

Figure 1. Development processes of BEmoC-v3.

Five human crawlers have manually accumulated Bengali text data from various online and offline sources. The primary sources include social media comments or posts (Facebook, YouTube), blog postings, textual conversations, narratives, storybooks, and news portals. A total of 7125 text documents are collected initially. Raw collected data requires several steps of preprocessing before labeling. Few pre-processing are done automatically, and the rest are performed manually:

- Automatic: Removed punctuation, digits, non-Bengali words, emoticons, and duplicate data. A module named 'BanglaProcess'[2] has been developed for automatic text pre-processing.

[2]https://pypi.org/project/BanglaProcess/

Table 1. A brief summary of previous works on Bengali textual emotion classification

Author(s)	Approach	Corpus Size	Emotion Classes	Critical Weaknesses
Tripto et al. [20]	Word2Vec + LSTM	2890	4	Dataset contained YouTube comments only
Rayhan et al. [24]	CNN + BiLSTM	7214	6	Dataset is translated from English thus inconsistent sentence formation
Azmin et al. [25]	Tf-Idf + Bigram + POS tagger + MNB	4200	3	Unable to handle morphological features
Pal et al. [27]	Tf-Idf + LR	32124	4	Only considered Bengali short stories
Das et al. [28]	Heuristic features + CRF	1300	6	Can not handle out-of-vocabulary words
Das et al. [31]	Lexical word level keyword spotting	1100	6	Suffers from the identification of morphologically changed keywords
Ruposh et al. [29]	BOW + SVM	1200	6	Failed in finding semantic relationships
Rahman et al. [26]	Tf-Idf + SVM	5640	6	Only considered Facebook comments
Das et al. [3]	XLM-Roberta	6243	6	Can not handle imbalanced data
Parvin et al. [30]	CNN + BiLSTM	9000	6	Can not dealt with the imbalanced data
Rahib et al. [21]	LSTM	10581	3	Limited to Covid-19 data only
Purba et al. [22]	CNN	995	3	Only three classes are considered

Table 2. Few samples of rejected sentences and modified labels after verification

Rejected samples	**Primary label**	**Expert label/Action**	**Cause**
সে প্রতিদিন অফিস থেকে সন্ধ্যায় বাসায় ফিরে। (He comes from office daily at the evening.)	Surprise	Discarded	Neutral emotion
তার চোখের পানি তে ছিল সুখেরই ঝলকানি। (There was a twinkle of happiness in her tears.)	Surprise	Discarded	Implicit emotion
সাহায্য না করে হাসতাছে এত বেহায়া মানুষ গুলো। (Without helping them, these brazen people are laughing.)	Anger	Disgust	Semantic of sentence
ভাই তোমাকে পুরস্কার দেওয়া উচিত। কিভাবে কাটিং গুলো মিলান অসাধারণ! (Brother, you should be rewarded. How great the cuts are!)	Joy	Surprise	Intensity of emotion word

- Manual: The text underwent a process of spelling correction and exclusion of texts containing less than three words to ensure an unwavering emotional adherence.

Following successful pre-processing, the corpus comprised 7000 texts that were subsequently forwarded to human annotators for manual labeling. The initial annotation task is assigned to five postgraduate students working in the Bengali language processing field with computer science and engineering backgrounds. The majority voting [7] process is employed to decide the primary label of the text.

B. VERIFICATION

The initial labeling of texts was examined by an expert with several years of experience conducting Bangla Language Processing (BLP) research. If any initial annotation was done incorrectly, the expert updated the labeling. Through conversations and extensive deliberations with the annotators, the NLP professional finalised the labels, ensuring a reduced likelihood of bias during the annotation process [34]. Table 2 illustrates some discarded data samples and their causes.

C. QUALITY OF ANNOTATION

We used the Cohens Kappa scores to determine inter-annotator congruence to assure the quality of the labeling. The quality of the corpus is reflected by inter-coder reliability (93.1%) and Cohens Kappa (0.91), which showed a perfect agreement among annotators [3]. The Jaccard index between the classes has been calculated for quantitative analysis. Table 3 shows the similarity values where the 200 most frequent words are utilized from each category. Two emotion class pairs (joy-surprise and anger-disgust) showed the highest similarity index of 0.51 and 0.55, respectively. These results reveal that more than half of the frequently used terms are familiar in these two groups. Nevertheless, the pair (joy-fear) obtained the lowest similarity, indicating that the frequent words in this pair are more distinctive than those in other categories. Thus, it is of concern that the similarity can significantly impact the classification task.

D. STATISTICS OF BEMOC-V3

Following the pre-processing and annotation procedure, the BEmoC-v3 comprised 7000 text documents. To

Table 3. Interclass *Jaccard* similarity index. Anger (CL1), Disgust (CL2), Fear (CL3), Joy (CL4), Sadness (CL5), Surprise (CL6)

	CL1	CL2	CL3	CL4	CL5	CL6
CL1	1.00	**0.55**	0.40	0.44	0.47	0.46
CL2	-	1.00	0.42	0.45	0.49	0.46
CL3	-	-	1.00	**0.38**	0.44	0.49
CL4	-	-	-	1.00	0.48	**0.51**
CL5	-	-	-	-	1.00	0.50

evaluate the models, the data is partitioned into three sets: training (5751 texts), validation (624 texts), and test sets (625 texts). As the data are imbalanced, it is preferable to distribute the corpus into training, validation, and test sets in such a fashion that the proportions of data in each class remain the same as they were in the original corpus [35]. Therefore, we performed a stratified sampling technique [8] while splitting the corpus. Table 4 shows statistics of data distribution in each category.

Table 4. Summary of the BEmoC-v3

Class	Training	Validation	Test
Anger	900	76	76
Disgust	1045	155	156
Fear	788	87	87
Joy	1295	114	115
Sadness	1089	119	119
Surprise	634	73	72
Total	5751	624	625

Fig. 2 shows the distribution of the number of texts versus the length of texts. According to the analysis, most of the data in this graph had a length between 15 to 35 words. Curiously, most of the texts in the *disgust* and *sadness* categories are between 20 to 30 words long. This depicts disgust that contents take more words to be expressed. The *Joy* and *Sadness* classes appear to have nearly identical numbers of textual data in the length distribution.

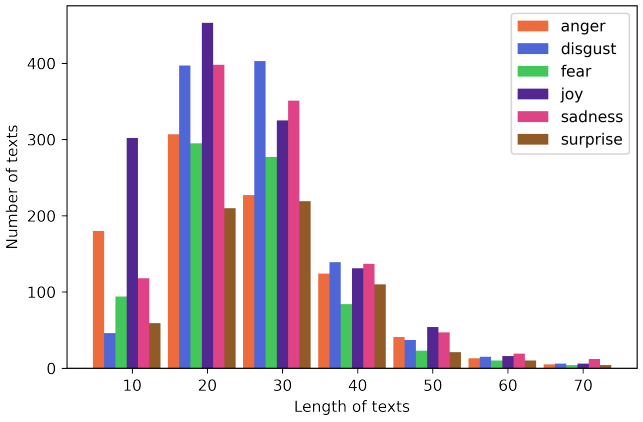

Figure 2. Corpus distribution concerning the number of texts vs length.

Fig. 3 represents the most frequent word distribution using Wordcloud. The words in the center are the most common, while those on the periphery are less common.

Figure 3. Wordcloud representation of the high-frequency emotion words in BEmoC-v3 for each emotion: (a) Anger, (b) Disgust, (c) Fear, (d) Joy, (e) Sadness, and (f) Surprise.

IV. METHODOLOGY

This work exploits several ML, DL, and transformer-based learning models with ensemble techniques for performing textual emotion classification in Bengali. Fig. 4 depicts a high-level overview of textual emotion classification.

The TF-IDF and Bag of Words feature extraction techniques are used for ML-based models (LR, RF, MNB, SVM), whereas Word2Vec, FastText, and pre-trained GloVe embeddings are used for DL-based models (CNN, BiLSTM, CNN+BiLSTM). Furthermore, we used transformer-based models (i.e., m-Bert, XLM-R, distil-BERT and two variants of Bangla-Bert: Bangla-BERT-1 and Bangla-BERT-2. This research also investigates the effect of transformer-based ensembling models for textual emotion classification. The same dataset (i,e. BEmoC-v3) will be used to train and tune all models.

A. FEATURE EXTRACTION

Several feature extraction techniques were utilized, including TF-IDF, Word2Vec, and FastText. These techniques transform the text data into a numerical representation of matrix or high dimensional vectors. These feature extractors are shown in Fig. 4.

1) Traditional Feature Extractor

Term Frequency-Inverse Document Frequency (TF-IDF): The TF-IDF [36] determines the significance of a word in text content. We extracted a

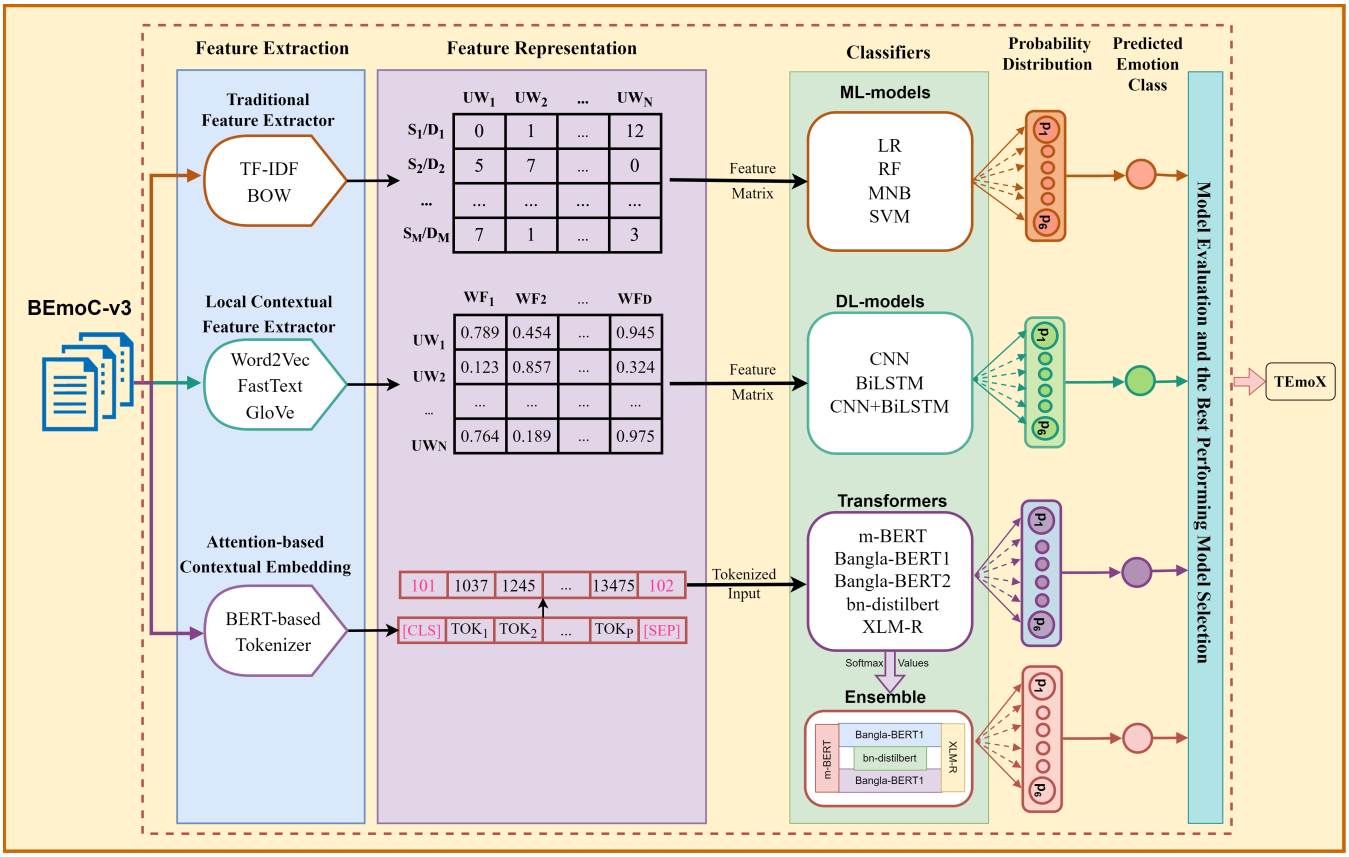

Figure 4. Abstract process of textual emotion classification in Bengali. Here, UW, S, D, and WF denote Unique Words, Sentences, Documents, and Word Features respectively. Moreover, M, N, and D are total sentences/documents, total unique words, and embedding dimensions, respectively. In BERT Tokenizer, TOK denotes Token.

combination of uni-gram and bi-gram features from the most frequent 20000 words of BEmoC-v3.

Bag of Words (BOW): The BOW [37] describes the frequency of words in a dataset. Unlike TF-IDF it does not provide information for more or less important words according to other documents in the dataset. We used the same parameters used in TF-IDF to train the BOW model.

2) Local Contextual Feature Extractor

Word2Vec: The Word2Vec is a popular and widely used word embedding technique for detecting the semantic similarities between words in a datasets context [38]. The Word2Vec algorithms have two variants: skip-gram and continuous BOW. According to [39], skip-gram works effectively with a tiny training data set and accurately depicts even uncommon words or phrases. In this work, the Word2Vec is trained using skip-gram with a window size of 7, embedding dimension of 100, and minimum word count of 4.

FastText: The Word2Vec algorithm can not handle out-of-vocabulary words; thus, any word not present in the test set can not be vectorized with a corresponding embedding value. The FastText algorithm is used to tackle this problem [40]. By leveraging sub-word information, this technique employs character n-grams to establish semantic relationships between words within a given context [41]. In this approach, when a word is absent in the training vocabulary, it can be synthesized using its constituent n-grams. Like Word2Vec, the FastText algorithm is available in both Skip-Gram and Continuous-BOW variations. We trained the FastText algorithm using skip-gram with a window size of 5, a character n-gram of size 5, and an embedding dimension of 100.

GloVe: It is a word vector technique for learning embeddings using word co-occurrences [42]. The GloVe does not rely solely on words local context information (like Word2Vec) to yield embeddings but instead utilizes global statistics on word co-occurrence. We used the pre-trained word vectors by [43] containing 39 M tokens, a vocab size of 0.18 M, and an embedding dimension of 100.

3) Attention-based Feature Extractor

Bert-based tokenizer: Bert-based multilingual tokenizers leverage the power of BERT's contextual embeddings to encode words and sentences in different languages, capturing their semantic meaning and context. XLM-R, m-BERT are such kinds of multilingual tokenizers. Besides utilizing these multilingual tokenizers we also used Bangla-BERT-1, Bangla-BERT-2, Bn-Distilbert which are pretrined on the Bangla language only.

B. FEATURE REPRESENTATION

By utilizing the traditional feature extraction algorithms that are frequency-based algorithms, we get a feature matrix. From Fig. 4 we can see, UW, S and D denote Unique Words, Sentences and Documents. Moreover, M and N are total sentences/documents and total unique words, respectively. embedding dimensions, respectively. In BERT Tokenizer, TOK denotes Token. The [CLS] token is a special token added at the beginning of each input sequence in BERT, representing the entire sequence for sentence-level tasks. The [SEP] token is used to separate segments or sentences when working with pairs of sequences, indicating their boundaries.

C. CLASSIFIERS

1) ML-based Approach

This work explored the four most widely used ML models to build an emotion detection system, including SVM, LR, MNB, and RF, where the TF-IDF and BoW are used as text vectorizers. For the LR, we choose the '*lbfgs* solver with the '*l1* penalty and set the maximum iteration to 400 for the solver to converge. The C value is kept at 1 for both LR and SVM. The SVM utilizes the rbf kernel with l2 penalizer. The RF is implemented with 100 estimator trees, and we keep the lowest number of instances required to divide an internal node at 2. For MNB, we set the Laplace smoothing parameter (alpha) to 1, enabling it to learn prior class probabilities. Table 5 shows a brief synopsis of the parameters employed for ML models.

Table 5. Parameters used for ML models

Classifier	Parameters
LR	optimizer = lbfgs, max_iter = 400, penalty = l1, C = 1
SVM	kernel = rbf, random_state = 0, gamma = scale, tol = 0.001
RF	criterion = gini, n_estimators = 100, min_samples_split = 2
MNB	alpha = 1.0, fit_prior = true, class_prior = none,

2) DL-based Approach

Various DL models (CNN, BiLSTM, CNN+ BiLSTM) are applied to BEmoC-v3, and their performances are investigated. All DL models employ Word2Vec and fastText as feature embedding. The DL algorithms performance depends heavily on the hyperparameters, which are tuned carefully to get an optimized network [44]. In general, this task is carried out by humans, which likely leads to suboptimal results. With enough computing resources, one can apply a grid search that executes all possible combinations of hyperparameters. However, as the hyperparameters and parameter space increase, the computation becomes intractable. The past study reveals that a model developed with randomly selected hyperparameters values could show better performance in lower computational than exhaustive grid search [45]. We have empirically determined the hyperparameter values of the embedding models and the classifiers based on our developed corpus. The models utilized the ADAM optimizer with a learning rate of 0.001 and were trained for 35 epochs per batch (with 16 samples). Keras callbacks were used to monitor the training process and save the model with the maximum validation accuracy in each epoch. The loss function chosen was sparse_categorical_crossentropy.

CNN: For analyzing the performance of CNN [46], we passed BEmoC-v3 to our scratch CNN model. For all of the convolutional and dense layers, we employed rectified linear units to introduce non-linearity, while the softmax activation was used for the output layers. Only one convolution block had a 1D convolution layer containing 64 filters with a size of 7. The training weights of Word2Vec and FastText embeddings are passed to the embedding layer, which generates a sequence matrix. This matrix is then processed by the following layer, global max pooling, to extract the maximum value from each filter. This process produces a single-dimensional vector of the same length as the number of filters used. Finally, an output layer with six nodes computes the probability distribution for each of the six emotion categories.

BiLSTM: Bidirectional Long-Short Term Memory (BiLSTM) is a kind of recurrent neural network (RNN) that can store information in both directions [47]. Basic RNN only looks at recent information while iterating over data and fails where long-term dependency is needed. We may need to look further back to get the semantic meaning of a text in the emotion detection task. BiLSTM overcomes this problem and works tremendously well for long-term dependency problems. The BiLSTM network contains an Embedding layer initialized with the Word2Vec or FastText embedding weights. The model includes a BiLSTM layer with 32 hidden units and a fully-connected dense layer with 16 neurons and ReLU activation. The output layer utilizes a softmax activation function to produce probability distributions for six emotion classes.

CNN+BiLSTM: A hybrid architecture combining CNN and BiLSTM has been explored to leverage the

Table 6. Hyperparameters for DNN methods

Hyperparameters	Hyperparameter space	CNN	BiLSTM	CNN + BiLSTM
Filter size	3,5,7,9	7	-	3
Nature of pooling	max, average	max	-	max
Embedding dimension	30, 35, 50, 70, 90, 100, 150, 200, 250, 300	100	100	100
Number of units	16, 32, 64, 128, 256	64	32	64,64,32
Dense layer units	16, 32, 64, 128, 256	64	16	-
Batch size	16, 32, 64, 128, 256	16	16	16
Activation function	'relu', 'softplus', 'tanh', 'sigmoid'	'relu'	'relu'	'relu'
Optimizer	'RMSprop', 'Adam', 'SGD', 'Adamax'	'Adam'	'Adam'	'Adam'
Learning rate	0.5, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001	0.001	0.001	0.001

advantages of both designs. Starting with an embedding layer initialized as in the previous procedure, a 1D convolutional layer with 64 filters (size 3) is added on top. Obeyed by this, a max-pooling layer downsampled the CNN features and transmitted them to two BiLSTM layers. The first layer comprises 64 LSTM units, while the second contains 32 LSTM units. Finally, the BiLSTM layer outputs are fed into a softmax-activated output layer that gives the probability distribution of six emotion classes.

Table 6 outlines the optimized hyperparameters of various DL models. The hyperparameter values are taken from the ranges mentioned in the 'Hyperparameter Space' field.

3) Transformer-based Approach

Transformer-based models, such as BERT (Bidirectional Encoder Representations from Transformers), can capture contextualized word representation from unlabeled texts [48]. It makes use of the encoder representation technique of the transformer architecture first introduced in [49]. There have been many pre-trained transformer models available in the Huggingface[3] transformers library to be used in text processing tasks. Recently pre-trained transformers variants are being employed in different domains of Bengali text processing tasks, including sentiment analysis [50] and document categorization [51], [52]. They outperformed the ML and DL models with higher accuracy.

This work implemented five transformer-based models: m-BERT, XLM-R, Indic-Transformers Bengali DistilBERT, and two variants of Bangla-BERT. All models are fine-tuned on the emotion corpora by employing Ktrain [53]. We embed specific start and end sequence tokens (SOS and EOS) at the beginning and end of the transformer model for fine-tuning. When required, we applied padding at the end of the sequence or remove any additional tokens that exceed the predetermined sequence length. The padded tokens are excluded during training to ensure they do not affect the training process. These tokens then go through multiple self-attention layers before being input into the transformer models.

m-BERT: We used the bert-base-multilingual-cased' model on BEmoC-v3 and fine-tuned it by modifying the batch size, learning rate, and epochs. The training process of m-BERT [54] involved using the most popular 104 languages with the most extensive Wikipedia data, including Bengali. The pre-trained m-BERT contains about 110M parameters.

XLM-R: XLM-R [55] trained with a multilingual masked language model. Providing various unique training procedures enhance the performance of BERT. These include (1) training the model for a more extended period with more data, (2) training with larger batch sizes and more extensive sequences, (3) dynamically constructing the masking pattern, and so on. The XLM-R model significantly surpasses other multilingual BERT models, especially in low-resource languages. The 'xlm-Roberta-base technique is implemented on BEmoC-v3 using a batch size of 12.

Bangla BERT: This work uses two variants of Bangla Bert that are dedicatedly pre-trained in the Bengali language only. The first one is 'sagorsarker/Bangla-bert-base'(hereafter called Bangla-BERT-1) [56] that is trained on Bengali corpus from OSCAR[4] and Bengali Wikipedia Dump Dataset[5]. Another one is 'csebuetnlp/banglabert'(hereafter called Bangla-BERT-2) [6]. Both pre-trained models are based on mask language modeling described in the original BERT paper [48].

Indic-DistilBERT: We implemented 'indic-transformers-bn-distilbert' on BEmoC-v3 and fine-tuned it to acquire adequate performance. The Indic Distilbert [57] is pre-trained on three main Indian languages (Hindi, Bengali, and Telugu), on which the amount of Bengali data is around 6 GB. We fine-tuned all the models on BEmoC-v3 using the Ktrain' auto fit' technique. All models are trained for 20 epochs using a learning rate of $2e^{-5}$ with a batch size of 12. Model weights are saved at checkpoints, and the most acceptable model is chosen

[3] https://huggingface.co/transformers/

[4] https://oscar-corpus.com/

[5] https://dumps.wikimedia.org/bnwiki/latest/

according to its performance on the validation set. The maximum sequence length for the texts settled at 50 words.

4) Ensemble-based Approach:

After individually deploying the pre-trained transformer models, we approached the transformers' ensemble. In recent years, the ensemble of transformer models proved to be more efficient than the individual ones [58]–[60]. We performed the ensembling to consider all possible combinations of classifier models using the weighted average and average ensemble techniques. The weighted average ensembles have specific effects on the ensembled outcome since the primary results of the base models can influence the ensemble outcomes. As a result, the best-performing model takes precedence over the others. On the other hand, the average ensemble takes the softmax probabilities of all the participating models and averages them. The output class in this averaging is the one with the highest probability. Prior base classifier results are not taken into account in this strategy [61], [62].

The framework of the ensemble of transformer-based Bengali TxtEC (i.e., TEmoX) is depicted in Fig. 5. The dataset is first sent to the BERT tokenizer, and the tokens are passed to the embedding layer (E) of each model. After passing the intermediate representation layers, a contextual representation (T) is achieved. Finally, a softmax probability distribution over the emotion classes is obtained. The probabilities are passed to a combination generator to generate the ensemble sets. Eq. 1 is used to determine the total number of ensemble sets generated from the combination generator.

$$C(m,r) = \frac{m!}{r!(m-r)!} \tag{1}$$

Here, $C(m,r)$ returns the number of total combinations, m represents the number of transformer models and r is the number of choosing models for the ensemble. For our task $m = 5$ and $r = 2, 3, 4, 5$, as we will be generating combinations of 2, 3, 4, and 5 models respectively. The formula can be rewritten as follows:

$$\begin{aligned} C(m_5, r_{2,3,4,5}) &= \frac{5!}{2!(5-2)!} + \frac{5!}{3!(5-3)!} \\ &\quad + \frac{5!}{4!(5-4)!} + \frac{5!}{5!(5-5)!} \\ &= 10 + 10 + 5 + 1 \\ &= 26 \end{aligned}$$

The 26 different combinations of the transformer models are named from EN-1 to EN-26. The probabilities from each variety are now passed to the 'Average Ensemble Algorithm' to get the output class.

Let us assume that we have 'd' test instances and the number of transformer models is 'm'. Each model classifies an instance d_i into one of the pre-defined categories from n_{class}. Thus for each instance d_i, a model m_j gives a softmax probability distribution vector(prb[]) of size n_{class}. Thus, the output becomes:

$$\begin{gathered} prb_{11},\ prb_{21},\ prb_{31},\ prb_{41},\ .....,\ prb_{d1} \\ prb_{12},\ prb_{22},\ prb_{32},\ prb_{42},\ .....,\ prb_{d2} \\ \vdots \\ prb_{1m},\ prb_{21},\ prb_{3m},\ prb_{4m},\ .....,\ prb_{dm} \end{gathered}$$

In the *average ensemble* technique the average of each softmax class value provided by 'm' models is calculated for each instance. Finally, the maximum of the probabilities is used to compute the output class using Eq. 2.

$$O = argmax(\frac{\forall_{i\in(1,d)} \sum_{j=1}^{m} prb_{ij}[]}{n_{class}}) \tag{2}$$

Here, the *argmax* function returns the class index of the maximum of the probabilities. The 'Average Ensemble' algorithm is briefly described in Algorithm 1.

Algorithm 1: Average Ensemble Algorithm

```
m[] ← models
d[] ← test instances
prb[] ← softmax probabilities
sum[] ← summed probabilities
avg[] ← average probabilities
n_class ← total number of classes
for ( i ∈ (1, d) ) {
    for ( j ∈ (1, m) ) {
        sum_i = sum_i + prb_ij[];
        j = j + 1;
    }
    avg_i = sum_i / n_class;
    i = i + 1;
}
O = argmax(avg) //Output class indices
```

The *weighted average ensemble* technique utilizes an extra weight with the softmax probabilities of the models. Given the prior weighted f1-scores of 'm' models, i.e., $wf_1, wf_2, ...wf_m$, the algorithm uses Eq. 3 to compute the outputs.

$$O = argmax(\frac{\forall_{i\in(1,d)} \sum_{j=1}^{m} prb_{ij}[] * wf_j}{\sum_{j=1}^{m} wf_j}) \tag{3}$$

Here, wf_j denotes the weighted F1 score of each model.

The 'Weighted Average Ensemble' algorithm is briefly described in Algorithm 2.

V. EXPERIMENTS

The entire experiment is conducted in a multicore processor furnished with NVIDIA Geforce GTX 960M GPU with 4GB graphics memory, 8GB physical memory (RAM), and a 2.3GHz intel core i5 processor. Scikit-learn (0.24.2) package is used to develop the ML models. The Ktrain (0.26.3) package is employed to train the

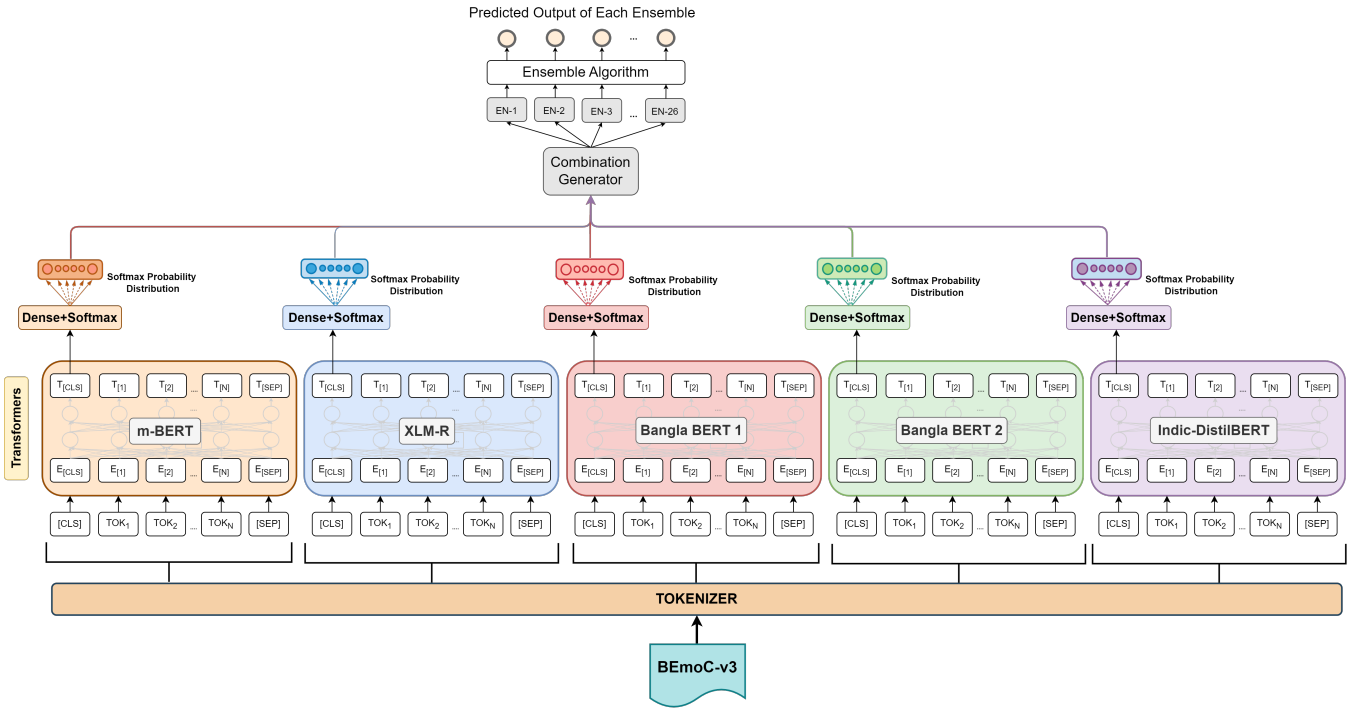

Figure 5. Framework for ensemble of transformer-based Bengali TxtEC. E_i denotes the input embedding for TOK_i, T_i represents the contextual representation for each TOK_i. EN-1 to EN-26 are the 26 different ensemble combinations made from the 5 transformer models.

Algorithm 2: Weighted Average Ensemble Algorithm

$m[\,] \leftarrow models$
$d[\,] \leftarrow test\ instances$
$prb[\,] \leftarrow softmax\ probabilities$
$sum[\,] \leftarrow summed\ probabilities$
$wf[\,] \leftarrow weighted\ f1\ scores$
$n_{class} \leftarrow total\ number\ of\ classes$
for ($i \in (1, d)$) {
  for ($j \in (1, m)$) {
    $sum_i = sum_i + prb_{ij}[\,]^* \ wf_j;$
    $j = j + 1;$
  }
  $i = i + 1;$
}
n_sum=0;
for ($j \in (1, l)$) {
  n_sum = n_sum + $wf_j;$
  $j = j + 1;$
}
$P = sum/n\_sum;$
$//Normalized\ Probabilities$;
$O = argmax(avg)$ $//Output\ class\ indices$

transformer models. Statistical measures such as precision (Pr), accuracy (Acc), recall (Re), and F1-score are utilized to assess and compare the models performance. Keras (2.4.3) with Tensorflow (2.5.0) backend is utilized as the DL framework. Python version is kept at v3.6 for all the experiments.

A. RESULTS

The evaluation results of individual models on the test set are presented in Table 7, with the excellency of the models determined based on the weighted F1-score. The results of the ensemble-based approaches based on various combination of the transformer models are reported in Table 8.

1) ML-based Approach

Results revealed that LR with TF-IDF features earned the most elevated $F1$-score (68.06%) among the ML approaches, surpassing SVM (64.81%), RF (60.57%) and MNB (57.53%). On the other hand, ML models with BOW features performed slightly less $F1$-score than TF-IDF. In addition, LR with TF-IDF outperformed other ML models concerning the highest Pr (68.52%), Re (67.84%), and Acc (67.84%) measures.

2) DL-based Approach

In relation to all of the evaluation parameters in DL models, CNN+BiLSTM with GloVe outperformed the remaining DL-based models by obtaining the highest $F1$-score of 63.39%, which is approximately 4.67% lower than the best ML approach (i.e., LR + TF-IDF).

Table 7. Performance of ML, DL and Transformer based models on the test set

Method	Classifier	Pr(%)	Re(%)	F1(%)	Acc(%)
ML models	**LR + TF-IDF**	**68.52**	**67.84**	**68.06**	**67.84**
	SVM + TF-IDF	69.66	64.32	64.81	64.32
	RF + TF-IDF	65.33	60.64	60.57	60.64
	MNB + TF-IDF	69.25	59.68	57.53	59.68
	LR + BOW	66.18	65.76	65.91	65.76
	SVM + BOW	61.19	58.56	59.14	58.56
	RF + BOW	65.67	59.84	60.51	59.84
	MNB + BOW	66.83	64.32	64.12	64.32
DL Models	CNN (Word2Vec)	61.72	58.56	59.00	58.56
	CNN (FastText)	59.31	54.24	55.86	54.24
	CNN (GloVe)	59.41	59.66	58.39	59.66
	BiLSTM (Word2Vec)	61.97	61.12	61.28	61.12
	BiLSTM (FastText)	62.03	60.16	60.55	60.16
	BiLSTM (GloVe)	63.17	63.20	63.34	63.20
	CNN + BiLSTM (Word2Vec)	63.78	62.24	62.23	62.24
	CNN + BiLSTM (FastText)	60.11	59.36	59.43	59.36
	CNN + BiLSTM (GloVe)	**63.52**	**63.76**	**63.39**	**63.76**
Transformers	m-BERT	63.45	61.76	62.25	61.76
	XLM-R	72.68	72.64	72.54	72.64
	Bangla-BERT-1	68.50	68.16	68.24	68.16
	Bangla-BERT-2	75.02	74.72	74.77	74.72
	Indic-DistilBERT	**77.11**	**77.12**	**77.11**	**77.12**

3) Transformer-based Approach

There is a considerable proliferation in all scores regarding the transformer-based models. The m-BERT acquired the lowest $F1$-score among all transformer-based models, only 62.25%, which is even lower than the best-performing ML model. The Bangla-BERT-1, on the other hand, has a nearly 5.99% higher $F1$-score (e.g., 68.24 %) than m-BERT. Compared to Bangla-BERT-1 and m-BERT, the XLM-R model significantly improved and obtained a $F1$-score of 72.54%. The Bangla-BERT-2 model was solely developed for the Bengali language with a larger corpus size than Bangla-BERT-1. As expected, Bangla-BERT-2 has outperformed the Bangla-BERT-1 and the aforementioned transformer-based models with an $F1$-score of 74.77%. Among all models, the Indic-DistilBERT obtained the highest $F1$-score of 77.11%. This model beats the m-BERT, XLM-R, Bangla-BERT-1, and Bangla-BERT-2 by 14.68%, 4.57%, 8.87%, and 2.34% $F1$-score, respectively. The best-performing model card is available at the huggingface model hub[6].

4) Ensemble-based Approaches

After analyzing the individual model's (i.e., base models) performance, we analyzed the ensemble of pre-trained transformers. Table 8 illustrates the evaluation scores of various ensemble sets on the test data concerning the average and weighted ensembles.

Results of ensembling indicate that the ensemble set En-22 utilizing the weighted-average ensemble approach demonstrated the best performance in terms of the highest precision (80.45%), recall (80.16%), and F1-score (80.24%). Thus, the result confirms that among a total of 38 models, the weighted-average ensemble model (Bangla-BERT-2 + XLM-R + Indic-DistilBERT + Bangla-BERT-1) is the best-performing model to classify textual emotion in Bengali. Therefore, it is to be called TEmoX.

[6]https://huggingface.co/avishek-018/bn-emotion-temox

B. ERROR ANALYSIS

Table 8 demonstrated that ensemble set *En-22* (Bangla-BERT-2 + XLM-R + Indic-DistilBERT + Bangla-BERT-1) is the best-performing ensemble model for classifying textual emotion in Bengali, as evidenced by its high performance. A detailed error analysis is conducted to gain additional insights regarding the performance of the proposed method.

1) Quantitative Analysis

Fig. 6 depicts the class-wise fraction of predicted labels relating to the confusion matrix.

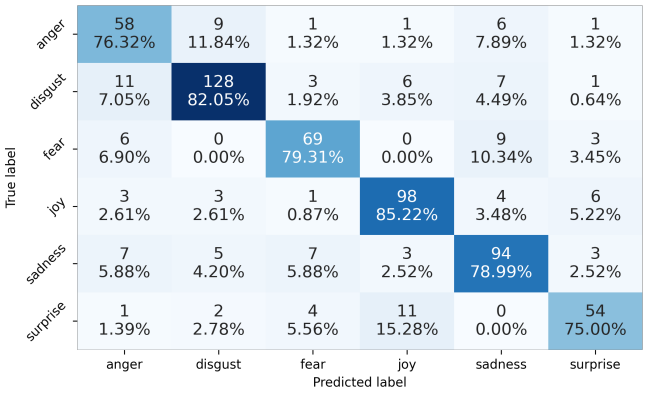

Figure 6. Confusion matrix of the proposed ensemble model (En-22).

Table 8. Performance of Ensemble-based models. Here the combination of 2, 3, 4, and 5 models are shown separately with their average/weighted Pr, Re, and F1-score

Ensemble Sets	Models	Weighted Ensemble			Average Ensemble		
		Pr(%)	Re(%)	F1(%)	Pr(%)	Re(%)	F1(%)
EN-1	Bangla-BERT-2, XLM-R	74.19	73.92	73.93	74.82	74.4	74.46
EN-2	Bangla-BERT-2, Indic-DistilBERT	77.14	76.96	76.99	77.74	77.44	77.49
EN-3	Bangla-BERT-2, mBERT	73.55	73.28	73.3	72.76	72.16	72.26
EN-4	Bangla-BERT-2, Bangla-BERT-1	74.21	73.76	73.87	75.17	74.88	74.97
EN-5	XLM-R, Indic-DistilBERT	76.52	76.48	76.5	75.79	75.68	75.71
EN-6	XLM-R, mBERT	72.32	72.32	72.29	69.85	69.44	69.56
EN-7	XLM-R, Bangla-BERT-1	75.06	74.88	74.92	74.02	73.92	73.95
EN-8	Indic-DistilBERT, mBERT	75.82	75.52	75.61	72.86	72.32	72.43
EN-9	Indic-DistilBERT, Bangla-BERT-1	75.90	75.68	75.74	74.87	74.72	74.76
EN-10	mBERT, Bangla-BERT-1	69.13	68.64	68.8	68.31	67.84	67.91
EN-11	Bangla-BERT-2, XLM-R, Indic-DistilBERT	79.15	79.04	79.07	78.70	78.56	78.60
EN-12	Bangla-BERT-2, XLM-R, mBERT	76.08	75.68	75.79	75.75	75.36	75.47
EN-13	Bangla-BERT-2, XLM-R, Bangla-BERT-1	78.12	77.92	77.98	77.82	77.60	77.67
EN-14	Bangla-BERT-2, Indic-DistilBERT, mBERT	78.26	77.92	77.99	77.88	77.60	77.66
EN-15	Bangla-BERT-2, Indic-DistilBERT, Bangla-BERT-1	78.59	78.40	78.47	79.21	79.04	79.09
EN-16	Bangla-BERT-2, mBERT, Bangla-BERT-1	74.47	74.08	74.15	74.36	73.76	73.85
EN-17	XLM-R, Indic-DistilBERT, mBERT	76.75	76.48	76.56	75.37	75.20	75.23
EN-18	XLM-R, Indic-DistilBERT, Bangla-BERT-1	76.95	76.8	76.84	76.89	76.80	76.83
EN-19	XLM-R, mBERT, Bangla-BERT-1	74.58	74.40	74.42	74.18	73.92	73.97
EN-20	Indic-DistilBERT, mBERT, Bangla-BERT-1	75.14	74.88	74.94	74.81	74.56	74.61
EN-21	Bangla-BERT-2, XLM-R, Indic-DistilBERT, mBERT	78.53	78.24	78.32	77.83	77.60	77.64
EN-22	**Bangla-BERT-2, XLM-R, Indic-DistilBERT, Bangla-BERT-1**	**80.45**	**80.16**	**80.24**	**79.54**	**79.20**	**79.29**
EN-23	Bangla-BERT-2, XLM-R, mBERT, Bangla-BERT-1	77.23	76.80	76.91	76.58	76.16	76.24
EN-24	Bangla-BERT-2, Indic-DistilBERT, mBERT, Bangla-BERT-1	78.20	77.76	77.9	76.59	76.16	76.27
EN-25	XLM-R, Indic-DistilBERT, mBERT, Bangla-BERT-1	76.98	76.80	76.87	76.56	76.32	76.40
EN-26	Bangla-BERT-2, XLM-R, Indic-DistilBERT, mBERT, Bangla-BERT-1	78.54	78.24	78.33	78.20	77.92	77.99

Table 9. Error rate of various approaches on the test set

Method	Classifier	Error Rate(%)
ML models	LR + TF-IDF	32.16
	RF + TF-IDF	35.68
	MNB + TF-IDF	39.36
	SVM + TF-IDF	40.32
	LR + BOW	34.24
	RF + BOW	41.44
	MNB + BOW	40.16
	SVM + BOW	35.68
DL models	CNN (Word2Vec)	41.44
	CNN (FastText)	45.76
	CNN (GloVe)	40.34
	BiLSTM (Word2Vec)	38.88
	BiLSTM (FastText)	39.84
	BiLSTM (GloVe)	36.80
	CNN + BiLSTM (Word2Vec)	37.76
	CNN + BiLSTM (FastText)	40.64
	CNN + BiLSTM (GloVe)	36.24
Transformers	m-BERT	38.24
	XLM-R	27.36
	Bangla-BERT-1	31.84
	Bangla-BERT-2	25.28
	Indic-DistilBERT	22.88
Ensemble of-Transformers	**Average Ensemble (Proposed)**	**19.84**

The confusion matrix revealed that a few data instances are not classified correctly. Among the 76 *anger* instances, 9 were predicted to be *disgust*. The same scenario can be noticed in the *disgust* class, where 11 instances are classified as *anger*. 9 out of 87 data instances in the *fear* class were incorrectly classified as *sadness* whereas 7 out of 119 data points in the *sadness* class were incorrectly classified as *fear*. Furthermore, out of 72 data points in the *surprise* class, 11 are predicted to be *joy*. The misclassification ratio is the highest in this class. The reasons for these misclassifications can be explained by the Jaccard similarity of the corpus (Table 3). Overlapping of the most frequent words can hamper the classification task. Also, an *anger* emotion can sometimes be expressed as *disgust* and thus their sentence pattern can have similarities too. This phenomenon remains true for the other two class pairs too (*joy-surprise* and *sadness-fear*). The error analysis reveals that the *disgust* class achieved the highest rate of correct classification (82.05%), while the *surprise* class achieved the lowest rate of correct classification (61.84%). Table 9 presents the error rate for various approaches where the proposed ensemble of transformers achieved the lowest error rate of 19.84%.

Table 10. Data samples demonstrating the differing nature of transformer models. Here MB, XR, BB1, BB2 and IDB refers to m-BERT, XLM-R, Bangla-BERT-1, Bangla-BERT-2 and Indic-DistilBERT. **A** denotes the actual label and the wrong predictions marked in bold

Sample	A	MB	XR	BB1	BB2	IDB	Proposed
গুটিয়ে লাল করে দেয়া দরকার। আহাম্মকদের এই হিপোক্রেট মোল্লাদের জন্য বাকি মুসলমানরা বিপদে আছে। (They should be beaten hard. The rest of the Muslims are in danger for these idiot hippocratic mullahs.)	Anger	Anger	Anger	**Sadness**	Anger	Anger	Anger
দূর থেকে মাকড়শা দেখলে, আমার ত্বকে সুড়সুড়ি অনুভব করি এবং নিজের অজান্তেই ভীত হয়ে উঠি। (When I see a spider from a distance, I feel a tingling sensation on my skin and I get scared unknowingly.)	Fear	**Sadness**	**Surprise**	Fear	Fear	Fear	Fear
বেহায়া গুলো মিডিয়ার মুখ দেখাবে কিভাবে? তারা তো পাকিস্তানে ঘোড়া ডিম পেড়ে এসেছে। (How will those shameless face the media? They built mare's nest in Pakistan)	Anger	**Disgust**	Anger	**Disgust**	Anger	**Disgust**	Anger
বিশ্বাস করতে পারি এমন কারো সাথে ভয় পাওয়ার ব্যাপার নিয়ে কথা বলি। (I talked to someone I trusted about me being scared.)	Fear	Fear	Fear	Fear	**Surprise**	Fear	Fear
আজীবন সাজাপ্রাপ্ত দণ্ডপ্রাপ্ত আসামীর মত বড় একা আমি বড় একা (I am as big as a convict sentenced to life)	Sadness	**Joy**	Sadness	Sadness	Sadness	Sadness	Sadness

2) Qualitative Analysis

Table 10 reports the predicted labels of some instances by the applied transformer models as well as their actual labels. It noticed that, for each sample, one model is predicting the correct label while the other is not. The ensemble technique (En-22) can help deal with these contradictory characteristics. It measures the softmax probability distribution average for each model and thus gives predictions based on maximum weighted-average probability. However, it is challenging to classify texts with overlapping words in different classes and it increases the misclassification rate of some models. Contextual analysis of such texts could bring about the development of better classification models.

Some words are commonly used in a variety of contexts across multiple classes. A high degree of class imbalance in the used corpus might be a probable cause for inaccurate predictions. The high value of the Jaccard index (Table 3), on the other hand, provides several intriguing aspects. For example, hateful words can convey both emotions: anger and disgust. The same scenario can be observed in the case of *joy* and *surprise* classes as a surprising incident might result in a positive outcome. Apart from these, emotion categorization is very subjective and relies on the individual's perspective, also people can think about a statement in numerous different forms depending on their preferences.

C. COMPARISON WITH EXISTING TECHNIQUES

The analysis of results demonstrated that the ensemble method, En-22, emerged as the most effective model for categorizing textual emotions in Bengali. We further evaluated the effectiveness of the proposed model by comparing its performance to that of existing techniques. Some past techniques [3], [20], [25]–[29] were implemented and evaluated on the BEmoC-v3. Table 11 shows the results of the comparisons. We can see the best performing model (i.e. TEmoX) outperformed the previous models and achieved the highest f1-score of 80.24%. Moreover, to exhibit the generalizability of the proposed technique, we experimented with its performance on another Bengali emotion dataset [26] (Dataset 2). This dataset consists of 6314 Facebook comments annotated with six emotion classes.

Table 11. Summary of the performance comparison

Approaches	F1(%)	
	BEmoC-v3	Dataset 2 [26]
TF-IDF + MNB [25]	57.53	40.42
Word2Vec + LSTM [20]	61.28	32.02
Heuristic features + CRF [28]	56.45	34.56
BOW + SVM [29]	59.14	52.55
TF-IDF + LR [27]	68.06	50.56
SVM + TF-IDF [26]	64.81	47.61
XLM-R [3]	72.54	52.32
En-22 (**Proposed**)	80.24	56.45

The comparative analysis exhibits (in Table 11) that the suggested technique is more robust than the existing techniques for classifying textual emotion in Bengali. Although Dataset 2 showed a relatively poor performance than BEmoC-v3, it performed better in the proposed method than in the past techniques. Some possible reasons might impact the unsatisfactory performance on Dataset 2. For further investigations, we have investigated the Jaccard Index of Dataset 2 and the confusion matrix of the proposed model. Table 12 presents the Jaccard Index, which showcases the overlapping of the most frequent words among the classes. The analysis considers the top 200 most frequent words from each emotion class to determine the degree of overlap. Thus,

it is evident that most class pairs have a similarity above 50%, which causes a high chance of misclassification.

Table 12. Interclass *Jaccard* similarity index. Anger (CL1), Disgust (CL2), Fear (CL3), Joy (CL4), Sadness (CL5), Surprise (CL6)

	CL1	CL2	CL3	CL4	CL5	CL6
CL1	1.00	0.47	0.51	0.40	0.56	0.47
CL2	-	1.00	0.52	0.43	0.54	0.52
CL3	-	-	1.00	0.47	0.54	0.50
CL4	-	-	-	1.00	0.45	0.41
CL5	-	-	-	-	1.00	0.50

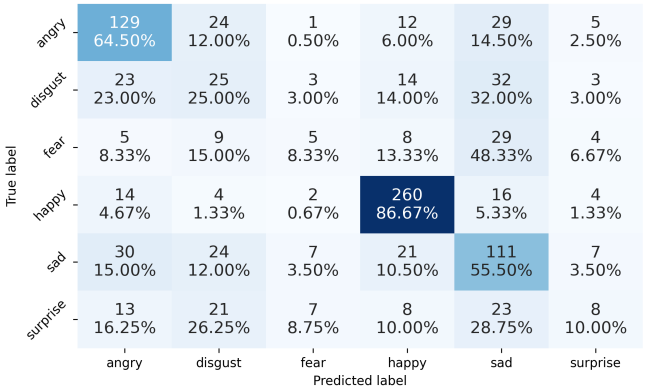

Figure 7. Confusion matrix of the proposed ensemble transformer models on Dataset-2.

Table 13. Results of the proposed method on different versions of BEmoC

Methods	F1(%)		
En-22 (**Proposed**)	BEmoC-v1 [63]	BEmoC-v2 [3]	BEmoC-v3 [7]
	69.45	76.95	80.24

In the confusion matrix in Fig. 7, it can be noticed that the number of misclassification is higher than the correct prediction in the *disgust, fear, and surprise* classes. The overlapping of the most frequent words among classes might be the reason for such performance. Moreover, there is no apparent justification for the construction of the dataset. Therefore, the quality indices of Dataset 2 might also influence the performance.

D. PERFORMANCE OF THE PROPOSED MODEL ON BEMOC DATSETS

The previous work [63] utilized a Bengali emotion dataset called *BEmoC-v1* containing 5200 texts, whereas Das et al. [3] used an extended version of BEmoC-v1 (renamed as *BEmoC-v2* having 6243 data. Later the dataset is extended again, containing 7000 texts, and we called it *BEmoC-v3*. To investigate the performance, we evaluated the proposed model on the three versions of BEmoC (v1-v3). The datasets are partitioned into the train, test, and validation sets, where the test set is kept identical so that it can assess the influence of increasing train data relatively. Table 13 illustrates the performance of the proposed model on the different versions of BEmoC. The analysis demonstrates that BEmoC-v1 (F1-score = 69.45%) performed relatively lower than BEmoC-v2 (F1-score = 76.96%) as it has fewer data. On the other hand, it clearly shows that BEmoC-v3 performed better by achieving the highest F1-score (80.24%) than BEmoC-v2 and BEmoC-v1 due to its more significant amount of text data.

VI. CONCLUSION

This research developed 38 classifier models for identifying textual emotions in Bengali and investigated their outcomes to determine six emotion classes (anger, fear, disgust, sadness, joy, and surprise). The performance of the proposed model (En-22: an ensemble of XLM-R, Bangla-BERT-1, Bangla-BERT-2, and Indic-DistilBER) is evaluated on a Bengali corpus (BEmoC-v3) and compared with seven existing techniques. The evaluation results clearly indicate that the proposed En-22 (TEmoX) approach outperformed both the base models and previous Bengali emotion classification approaches in terms of classifying textual emotion. The En-22 approach achieved the highest performance with an impressive $F1$-score of 80.24% in Bengali emotion classification. This performance indicated an improvement of 12.18%, 16.85%, and 3.13% to the best-performed ML, DNN, and transformer-based models. In the future, this work plans to identify more emotion categories (such as love, hate, and stress) with more diverse data in the developed corpus. Future research also aims to analyze the proposed models suitability to detect emotion conveyed by emoticons, code-mix or switching data, and mixed-emotion text. It will also be attractive to examine the effect of the developed model in classifying multilabel textual emotion in Bengali.

References

[1] K. Garg and D. Lobiyal, "Hindi emotionnet: A scalable emotion lexicon for sentiment classification of hindi text," *ACM Tran. on Asian & Low-Resource Language Information Processing*, vol. 19, no. 4, pp. 1–35, 2020.

[2] T. Parvin, O. Sharif, and M. M. Hoque, "Multi-class textual emotion categorization using ensemble of convolutional and recurrent neural network," *SN Com. Sci.*, vol. 3, no. 62, 2022.

[3] A. Das, O. Sharif, M. M. Hoque, and I. H. Sarker, "Emotion classification in a resource constrained language using transformer-based approach," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*. Online: Association for Computational Linguistics, Jun. 2021, pp. 150–158. [Online]. Available: https://aclanthology.org/2021.naacl-srw.19

[4] A. Wadhawan and A. Aggarwal, "Towards emotion recognition in Hindi-English code-mixed data: A transformer based approach," in *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Online: Association for Computational Linguistics, Apr. 2021, pp. 195–202. [Online]. Available: https://aclanthology.org/2021.wassa-1.21

[5] J.-B. Delbrouck, N. Tits, M. Brousmiche, and S. Dupont, "A transformer-based joint-encoding for emotion recognition and sentiment analysis," in *Second Grand-Challenge and Workshop on Multimodal Language (Challenge-HML)*. Seattle, USA: Association for Computational Linguistics, Jul. 2020, pp. 1–7. [Online]. Available: https://aclanthology.org/2020.challengehml-1.1

[6] A. Bhattacharjee, T. Hasan, K. Samin, M. S. Islam, M. S. Rahman, A. Iqbal, and R. Shahriyar, "Banglabert: Combating embedding barrier in multilingual models for low-resource language understanding," *CoRR*, vol. abs/2101.00204, 2021. [Online]. Available: https://arxiv.org/abs/2101.00204

[7] M. Iqbal, A. Das, O. Sharif, M. M. Hoque, and I. H. Sarker, "Bemoc: A corpus for identifying emotion in bengali texts," *SN Computer Science*, vol. 3, no. 2, pp. 1–17, 2022.

[8] V. L. Parsons, "Stratified sampling," *Wiley StatsRef: Statistics Reference Online*, pp. 1–11, 2014.

[9] N. Alswaidan and M. E. B. Menai, "A survey of state-of-the-art approaches for emotion recognition in text," *Knowledge and Information Systems*, pp. 1–51, 2020.

[10] F. Calefato, F. Lanubile, and N. Novielli, "Emotxt: a toolkit for emotion recognition from text," in *2017 seventh international conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*. IEEE, 2017, pp. 79–80.

[11] S. Alzu'bi, O. Badarneh, B. Hawashin, M. Al-Ayyoub, N. Alhindawi, and Y. Jararweh, "Multi-label emotion classification for arabic tweets," in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE, 2019, pp. 499–504.

[12] S. Ahmad, M. Z. Asghar, F. M. Alotaibi, and S. Khan, "Classification of poetry text into the emotional states using deep learning technique," *IEEE Access*, vol. 8, pp. 73 865–73 878, 2020.

[13] A. Yousaf, M. Umer, S. Sadiq, S. Ullah, S. Mirjalili, V. Rupapara, and M. Nappi, "Emotion recognition by textual tweets classification using voting classifier (lr-sgd)," *IEEE Access*, vol. 9, pp. 6286–6295, 2021.

[14] M. Hasan, E. Rundensteiner, and E. Agu, "Automatic emotion detection in text streams by analyzing twitter data," *International Journal of Data Science and Analytics*, vol. 7, no. 1, pp. 35–51, 2019.

[15] Y. Lai, L. Zhang, D. Han, R. Zhou, and G. Wang, "Fine-grained emotion classification of chinese microblogs based on graph convolution networks," *World Wide Web*, vol. 23, no. 5, pp. 2771–2787, 2020.

[16] D. Haryadi and G. P. Kusuma, "Emotion detection in text using nested long short-term memory," *IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 10, no. 6, 2019.

[17] A. Chatterjee, K. N. Narahari, M. Joshi, and P. Agrawal, "Semeval-2019 task 3: Emocontext contextual emotion detection in text," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, 2019, pp. 39–48.

[18] I. Ameer, G. Sidorov, H. Gómez-Adorno, and R. M. A. Nawab, "Multi-label emotion classification on code-mixed text: Data and methods," *IEEE Access*, vol. 10, pp. 8779–8789, 2022.

[19] P. Kumar and B. Raman, "A bert based dual-channel explainable text emotion recognition system," *Neural Networks*, vol. 150, pp. 392–407, 2022.

[20] N. I. Tripto and M. E. Ali, "Detecting multilabel sentiment and emotions from bangla youtube comments," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*. IEEE, 2018, pp. 1–6.

[21] M. Rahib, R. H. Khan, A. H. Tamim, M. Z. Tahmeed, and M. J. Hossain, "Emotion detection based on bangladeshi peoples social media response on covid-19," *SN Computer Science*, vol. 3, no. 2, pp. 1–6, 2022.

[22] S. A. Purba, S. Tasnim, M. Jabin, T. Hossen, and M. K. Hasan, "Document level emotion detection from bangla text using machine learning techniques," in *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*. IEEE, 2021, pp. 406–411.

[23] M. M. R. Mamun, O. Sharif, and M. M. Hoque, "Classification of textual sentiment using ensemble technique," *SN Computer Science*, vol. 3, no. 1, p. 49, Nov 2021. [Online]. Available: https://doi.org/10.1007/s42979-021-00922-z

[24] M. M. Rayhan, T. Al Musabe, and M. A. Islam, "Multilabel emotion detection from bangla text using bigru and cnn-bilstm," in *2020 23rd International Conference on Computer and Information Technology (ICCIT)*. IEEE, 2020, pp. 1–6.

[25] S. Azmin and K. Dhar, "Emotion detection from bangla text corpus using naïve bayes classifier," in *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*. IEEE, 2019, pp. 1–5.

[26] M. A. Rahman and M. H. Seddiqui, "Comparison of classical machine learning approaches on bangla textual emotion analysis," 2019.

[27] A. Pal and B. Karn, "Anubhuti–an annotated dataset for emotional analysis of bengali short stories," *arXiv preprint arXiv:2010.03065*, 2020.

[28] D. Das and S. Bandyopadhyay, "Word to sentence level emotion tagging for bengali blogs," in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, 2009, pp. 149–152.

[29] H. A. Ruposh and M. M. Hoque, "A computational approach of recognizing emotion from bengali texts," in *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*. IEEE, 2019, pp. 570–574.

[30] T. Parvin, O. Sharif, and M. M. Hoque, "Multi-class textual emotion categorization using ensemble of convolutional and recurrent neural network," *SN Computer Science*, vol. 3, no. 1, p. 62, Nov 2021. [Online]. Available: https://doi.org/10.1007/s42979-021-00913-0

[31] D. Das, S. Roy, and S. Bandyopadhyay, "Emotion tracking on blogs-a case study for bengali," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 2012, pp. 447–456.

[32] F. Alam, M. A. Hasan, T. Alam, A. Khan, J. Tajrin, N. Khan, and S. A. Chowdhury, "A review of bangla natural language processing tasks and the utility of transformer models," *CoRR*, vol. abs/2107.03844, 2021. [Online]. Available: https://arxiv.org/abs/2107.03844

[33] T. Alam, A. Khan, and F. Alam, "Bangla text classification using transformers," *CoRR*, vol. abs/2011.04446, 2020. [Online]. Available: https://arxiv.org/abs/2011.04446

[34] O. Sharif and M. M. Hoque, "Identification and classification of textual aggression in social media: Resource creation and evaluation," in *Combating Online Hostile Posts in Regional Languages during Emergency Situation*, T. Chakraborty, K. Shu, H. R. Bernard, H. Liu, and M. S. Akhtar, Eds. Cham: Springer International Publishing, 2021, pp. 9–20.

[35] R. J. May, H. R. Maier, and G. C. Dandy, "Data splitting for artificial neural networks using som-based stratified sampling," *Neural Networks*, vol. 23, no. 2, pp. 283–294, 2010.

[36] T. Tokunaga and I. Makoto, "Text categorization based on weighted inverse document frequency," in *Special Interest Groups and Information Process Society of Japan (SIG-IPSJ*. Citeseer, 1994.

[37] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1-4, pp. 43–52, 2010.

[38] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[39] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13. Red Hook, NY, USA: Curran Associates Inc., 2013, p. 31113119.

[40] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext. zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651*, 2016.

[41] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

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

[43] S. Sarker, "Bnlp: Natural language processing toolkit for bengali language," *arXiv preprint arXiv:2102.00405*, 2021.

[44] D. J. C. MacKay, *Hyperparameters: Optimize, or Integrate Out?* Dordrecht: Springer Netherlands, 1996, pp. 43–59. [Online]. Available: https://doi.org/10.1007/978-94-015-8729-7_2

[45] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 281–305, 2012.

[46] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[47] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

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

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017. [Online]. Available: https://arxiv.org/pdf/1706.03762.pdf

[50] M. A. Hasan, J. Tajrin, S. A. Chowdhury, and F. Alam, "Sentiment classification in bangla textual content: A comparative study," in *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, 2020, pp. 1–6.

[51] T. Alam, A. Khan, and F. Alam, "Bangla text classification using transformers," *arXiv preprint arXiv:2011.04446*, 2020.

[52] A. Kunchukuttan, D. Kakwani, S. Golla, A. Bhattacharyya, M. M. Khapra, P. Kumar *et al.*, "Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages," *arXiv preprint arXiv:2005.00085*, 2020.

[53] A. S. Maiya, "ktrain: A low-code library for augmented machine learning," *arXiv preprint arXiv:2004.10703*, 2020.

[54] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[55] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

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

[57] K. Jain, A. Deshpande, K. Shridhar, F. Laumann, and A. Dash, "Indic-transformers: An analysis of transformer language models for indian languages," *CoRR*, vol. abs/2011.02323, 2020. [Online]. Available: https://arxiv.org/abs/2011.02323

[58] V. Bhatnagar, P. Kumar, S. Moghili, and P. Bhattacharyya, "Divide and conquer: An ensemble approach for hostile post detection in hindi," *arXiv preprint arXiv:2101.07973*, 2021.

[59] S. Tawalbeh, M. Hammad, and M. AL-Smadi, "KEIS@JUST at SemEval-2020 task 12: Identifying multilingual offensive tweets using weighted ensemble and fine-tuned BERT," in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. Barcelona (online): International Committee for Computational Linguistics, Dec. 2020, pp. 2035–2044. [Online]. Available: https://aclanthology.org/2020.semeval-1.269

[60] K. A. Das, A. Baruah, F. A. Barbhuiya, and K. Dey, "KAFK at SemEval-2020 task 12: Checkpoint ensemble of transformers for hate speech classification," in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. Barcelona (online): International Committee for Computational Linguistics, Dec. 2020, pp. 2023–2029. [Online]. Available: https://aclanthology.org/2020.semeval-1.267

[61] S. Gundapu and R. Mamidi, "Transformer based automatic COVID-19 fake news detection system," *CoRR*, vol. abs/2101.00180, 2021. [Online]. Available: https://arxiv.org/abs/2101.00180

[62] S. M. S. Shifath, M. F. Khan, and M. S. Islam, "A transformer based approach for fighting COVID-19 fake news," *CoRR*, vol. abs/2101.12027, 2021. [Online]. Available: https://arxiv.org/abs/2101.12027

[63] A. Das, M. A. Iqbal, O. Sharif, and M. M. Hoque, "Bemod: Development of bengali emotion dataset for classifying expressions of emotion in texts," in *Intelligent Computing and Optimization*, P. Vasant, I. Zelinka, and G.-W. Weber, Eds. Cham: Springer International Publishing, 2021, pp. 1124–1136.