Multilabel Aggressive Comments Detection from Social Media using Deep Learning Techniques

Mahathir Mohammad Bishal, Shawly Ahsan, and Mohammed Moshiul Hoque*
Department of Computer Science & Engineering
Chittagong University of Engineering and Technology, Chattogram-4349, Bangladesh
Email: {u1604083, u1704057}@student.cuet.ac.bd, moshiul_240@cuet.ac.bd

***Abstract*—In recent years, researchers have attempted to identify and classify unwanted textual information (i.e., aggressive, abusive, offensive, hateful, and toxic) in online media due to its adverse effects on society. Several initiatives have been implemented to reduce the consumption and propagation of such information. However, most research has primarily focused on English and other high-resourced languages, leaving low-resourced languages (e.g., Bengali) largely unaddressed. This paper proposes an intelligent technique using deep learning (DL) to classify aggressive Bengali text into five categories: ReAG, PoAG, VeAG, GeAG, and RaC. Due to the scarcity of benchmark corpora, this work constructed a large Bengali corpus comprising 4,002 comments, totaling 65,436 words, to perform the multilabel aggressive comment identification task. Five ML models, including Linear Regression (LR), Linear Support Vector Classifier (LSVC), Multinomial Naive Bayes (MNB), Decision Tree (DT), and Random Forest (RF), along with three DL models, namely CNN, LSTM, and BiLSTM, were applied to the developed corpus using various feature extraction techniques to address the downstream task. The comparative analysis revealed that the CNN with GloVe embedding (CNN+GloVe) outperformed the other methods (0.88 weighted f_1 score) on the test dataset.**

***Index Terms*—Natural language processing, Text processing, Multilabel text classification, Deep learning, Aggressive comments detection**

I. Introduction

With the expansion of user-generated content on social media, it has been observed that aggressive or undesired content has increased exponentially in recent years. Aggression on social media aims at a specific person or community to harm their identity and reduce their status and recognition [1]. Political philosophy, religious convictions, sexual orientation, gender, and nationality are some elements that influence this. Aggressive texts aim to hurt an individual or a group of individuals by inciting violence or other forms of damage. Social sites like Facebook, YouTube, and Twitter contain vast amounts of data about people's opinions and personal lives worldwide. Sometimes, one person's ideology conflicts with that of another. Along with a rise in user-generated content on social media, the level of aggression is rising [2]. Due to increased internet usage, social media and networking have grown tremendously over time. When a discussion or conversation begins, disagreements are inevitable due to differences in viewpoint. However, these disagreements often escalate into heated confrontations on social media, during which one side may resort to aggressive language. This aggressiveness could be political, religious, gendered, threatening, or racial.

Identifying aggressive texts in low-resourced languages, including Bengali, is in the introductory stage. The unavailability of standard corpora, the scarcity of language processing tools, and the complicated morphological constructs of the language are significant barriers to developing automatic aggressive text detection in Bengali. Few studies focused on detecting multiple aggression classes in Bengali [3], [4], but past studies still need to address multilabel aggression detection. A comment or text sometimes contains more than one class. In multi-label text classification, a document can be linked to several labels from a set of predefined categories, reflecting real-world scenarios where documents may simultaneously belong to more than one category. For instance, the sentence "কালার ঘরের কালা তুই ধর্মের কি বুঝিস, এইসব ভণ্ডামি ছাইড়া দিনের রাস্তায় আয়। গাধা কোথাকার।" *(You black people. What do you know about religion? Stop this hypocrisy. Come to sense, you donkey.)* expresses racial, religious, and verbal aggression at the same time. Therefore, identifying multilabel aggressive text is a critical research issue in Bengali language processing. This research aims to develop a multilabel classifier to detect and classify aggressive social media comments using DL techniques. The significant contributions of this work are illustrated in the following:

- Developed a corpus (called M-BATC) containing 4002 Bengali text data with five labels: *PoAG (Politically Aggressive)*, *ReAG (Religiously Aggressive)*, *VeAG (Verbally Aggressive)*, *GeAG (Gender Aggressive)*, and *Racism.*
- Developed a DL model with tuned hyperparameters with Glove embedding (CNN+GloVe) to classify textual aggression with multiple labels in Bengali.
- Investigated various ML and DL models to measure the task performance with a detailed error analysis for finding a suitable model for multilabel aggressive text classification in Bengali.

II. Related Work

In recent years, detecting hate speech, undesired text, or abusive content on social media and online blogs has

979-8-3315-5525-2/25/$31.00 © 2025 IEEE

attracted researchers because of its harmful societal consequences [5]. Medisety et al. [6] proposed a majority voting-based ensemble method to categorize social media posts into overtly aggressive, covertly aggressive, and non-aggressive categories. Using an ensemble of CNN, LSTM, and BiLSTM models on two datasets, they achieved the highest F1-score of 0.604 for Facebook and 0.504 for other social media posts. Similarly, Ibrohim et al. [7] created a hate speech dataset of 5,561 Indonesian tweets, grouped into weak, moderate, and strong levels. They applied multilabel classification with various ML techniques and label transformation methods, achieving 77.36% accuracy. Mathur et al. [8] developed a dataset of 3,600 code-switched Hindi-English tweets, divided into abuse, hate speech, and non-offensive classes. Their study utilized CNN with transfer learning strategy, resulting in 71% f_1 score.

Despite the considerable strides made in English and other languages with high resources, progress in language processing tasks for Bengali remains limited. Only a few studies have explored the detection of aggressive or undesired text in this language. For example, Das et al. [9] developed a dataset of 7,425 Bengali comments divided into seven classes. Their model achieved 77% accuracy using LSTM and Gated Recurrent Unit (GRU). A recent study [10] proposed a transformer-based method (XLM-R) to classify violent Bengali literature into overt, covert, and non-aggressive categories. They used the Bengali dataset [11], which contains 4,000 texts, and their method achieved the best $F1$-score of 0.842. A transformer-based model is proposed for detecting cyberbullying in Bengali, outperforming ML and DL baselines with an accuracy of 80.2% [12]. Ghosal et al. [13] devised an unsupervised framework for detecting hate content in Hindi and Bengali, achieving a maximum F1-score of 0.74 for Hindi and 0.88 for Bengali datasets. Islam et al. [14] employed a CLSTM model to detect cyberbullying using a multi-label dataset, which obtained 0.883 macro f_1 score.

In summary, most past studies have focused on classifying toxic comments or abusive language using ML techniques and assigning texts to binary or multiple classes. A key distinction of the present work is its focus on developing a DL-based technique capable of handling numerous labels within aggressive text.

III. M-BATC: Multilabel Bengali Aggressive Text Corpus

Due to the unavailability of a standard aggressive text dataset in Bengali, this work developed an M-BATC (Multilabel Bengali Aggressive Text Corpus) following the guidelines articulated by Sharif et al. [4]. This corpus was initially annotated by three undergraduate computer science students and further validated by an NLP expert, an academic with several years of experience in NLP.

A. Data Collection and Preprocessing

Aggressive text or comments were collected from two significant sources: Facebook (63.9%) and YouTube (35.1%), with the remaining 1% of data collected from other online sources. Three students with an undergraduate engineering background were assigned to accumulate data. They manually collected 4,002 text comments over six months (March 2023 to August 2023). Raw data may contain inconsistencies, duplication, noise, and unnecessary information. Preprocessing is necessary to reduce discrepancies and get precise analytical results. By applying automated data preprocessing, we removed punctuation, numbers, emojis, and unnecessary characters from the text, as they do not carry any meaningful information.

B. Task Description and Data Annotation

This work aims to classify a text into one of the five distinctive, aggressive classes. Thus, we first define the scope of five distinct textual aggression classes based on the previous literature [4].

- **Religiously Aggressive (ReAG):** Incite violence by insulting a community's faith, religious organization, or religious belief (Catholic, Hindu, Jew, or Islamic, for example).
- **Politically Aggressive (PoAG):** Denigrate political ideologies, urge political party supporters, or incite people to violence against law enforcement and the state
- **Verbally Aggressive (VeAG):** Seek to do wrong or hurt others, denigrate social standing by the use of curse words, filthy words, provocative and other threatening language
- **Gender Aggressive (GeAG):** Make an aggressive allusion about sexual orientation, sexuality, bodily parts, or other vulgar content to an individual or group
- **Racism (RaC):** Attack or insult people and promote aggression based on race.

The data is accumulated from several sources, including Facebook (2557), YouTube (1404), and others (41). Each data point is labeled manually and followed by a majority label in a class to assign the suitable label (Algorithm 1). Matrix I represents the labels assigned by annotators to the data samples, where the element d_{ab} in matrix I indicates the label given by annotator b to data sample a. Matrix L denotes the final label selected for each data sample, with L_a signifying the final label assigned to data sample a. The label with the most votes from the annotators is chosen as the initial label. Three undergraduate students with a computer engineering background completed the initial labeling or annotation assignments.

C. Label Verification

The annotators were instructed to annotate the texts without bias toward any particular demographic region,

Algorithm 1: Compute Final Label

```
$C \leftarrow$ Class Number;
$D \leftarrow$ Text Data;
$I \leftarrow$ Initial Labels;
$L \leftarrow$ Final Labels;
for $d_a \in D$ do
    $label\_count[C] \leftarrow 0$;
    for $d_{ab} \in I$ do
        label_count[d$_{ab}$]++;
    end
    $L_a = get\_index(max(label\_count))$;
end
```

culture, sensitive subject, or religion. Each labeling performed by the annotators was validated by a specialist who has been an academician working on NLP for several years. If the expert's label matches the initial label, the label is correct; otherwise, the expert changed the labels on the data since the annotators' initial labeling had errors. Using Cohen's kappa score [15], we investigate how much the annotators agree on class assignments. The average Kappa value for our dataset is 0.87, which corresponds to almost perfect agreement according to the Kappa scale.

D. Dataset Statistics

As the dataset is a multi-label dataset, a single text instance can have multiple labels (for example, one sentence may be labeled as both 'PoAG', 'VeAG', and 'GeAG'). Although the total number of multilabel unique texts in the M-BATC is 4,002, the dataset contains a total of 8,487 texts due to overlapping labels. Among 8447 texts, 1894 are labeled as ReAG, 1595 as PoAG, 3453 as VeAG, 1321 as GeAG, and 224 as RaC. Table I illustrates the dataset statistics in each class. The text's length varies in each class; most texts are between 40 and 200 characters long. The M-BATC consists of a total of 65,436 words. The dataset was partitioned into three parts for training and evaluation: 80% was allocated for training, while both validation and testing received 10% each. The total number of words (T_W) in the dataset is 142162, and the total number of sentences (T_S) is 15535.

Table I: Class-wise data distribution, where T_D, T_S, T_W, T_{UW}, and L_A denote the total amount of data, sentences, words, unique words, and average sentence length, respectively

Class	T_D	T_S	T_W	T_{UW}	L_A
ReAG	1894	3752	35586	9806	109
PoAG	1595	3132	28522	8017	105
VeAG	3453	6211	56086	13762	93
GeAG	1321	2130	19162	6040	81
RaC	224	3310	2806	1541	71
Total	8487	15535	142162	39166	459

Table II: Hyperparameters of the proposed CNN model. The symbols FS, ED, N, BS, AF, and LR represent the filter size, embedding dimension, number of neurons, batch size, activation function, and learning rate

H	**Hyperparameter (H) Space**	**Value**
FS	3, 5, 7, 9	3
ED	30, 35, 50, 70, 100, 150, 200, 250, 300	300
N	16, 32, 64, 128, 256	64
BS	16, 32, 64, 128, 256	128
Ptype	'max', 'average'	'max'
AF(DLayer)	'relu', 'softplus', 'sigmoid', 'tanh'	'relu'
AF(OLayer)	'relu', 'softplus', 'sigmoid', 'tanh'	σ
Optimizer	'RMSprop', 'Adam', 'SGD', 'Adamax'	'Adam'
LR	0.5, 0.1,0.01, 0.005, 0.001, 0.0005, 0.0001	0.001

IV. Methodology

The primary aim of this study is to develop an automatic system that can detect multiple aggressive texts or comments in Bengali. This work investigated distinct ML and DL approaches to perform the task. The proposed technique consists of four major components: preprocessing, textual feature extraction, classification model preparation, and prediction. Figure 1 illustrates an abstract process, which includes the employed models.

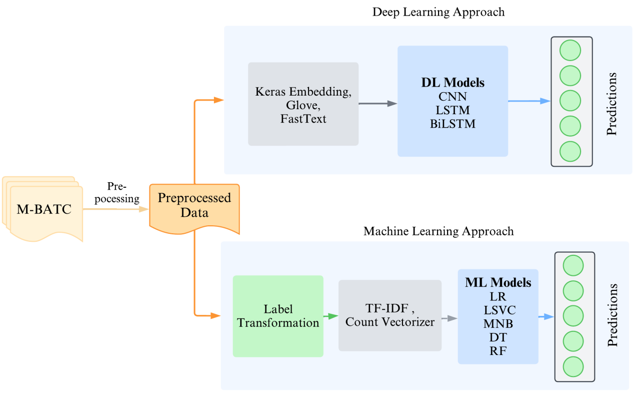


Figure 1: Abstract process of multilabel aggressive comment detection

A. Preprocessing

Raw data may contain inconsistencies, duplication, noise, and unnecessary information (e.g., stopwords). Preprocessing is necessary to reduce discrepancies and get precise analytical results. Automated data preprocessing removes punctuation, numbers, emojis, and other unnecessary characters from text, as they do not convey vital information. After that, texts are translated into a set of tokens. Tokenization is the process of breaking down a given text into meaningful tokens.

B. Textual Feature Extraction

A feature extraction technique converts text into a numerical representation in vector form for training the classifiers.

- **CountVectorizer:** It refers to separating a sentence into its constituent words [16]. Consequently, an encoding vector is returned, comprising an integer count of how many times each word appears in the sentence and the overall vocabulary size. The parameter features for CountVectorizer is set to 15k.
- **TF-IDF:** It is used to lessen the impact of less informative words that appear frequently in texts. TF denotes a word's frequency in a document, whereas IDF assesses the significance of the word throughout the entire corpus. To extract the combined unigram and bigram properties, the 15k most common words are considered.
- **Word Embeddings:** Several embedding techniques (i.e., Keras, Glove, and FastText) extract the textual features. Each word vectors are assigned to one-hot encoded words [17]. A vocabulary of 15k unique words is created to obtain the features. Finally, Word2Vec and FastText techniques utilize embedding dimensions of 300 to learn their features.

C. Classifiers

Five popular ML and three DL classifiers are investigated for the downstream task. The feature extractor converts unread text into feature sets, which are the input to the classifier model and are used to predict multiple labels for each given instance (e.g., PoAG, ReAG).

ML Models: LR, LSVC, MNB, DT, and RF models are exploited for the classification task. Since ML models cannot directly handle multilabel data, we convert the dataset into single-label multiclass data using the label powerset method. Several parameter combinations are evaluated and tweaked to train the classifier models. To train, the LR model 'lbfgs' optimizer with '10' regularization is employed, where the C value is fixed to the default 1.0. For LSVC, defision_function_shape was 'ovr', and gamma value 0.1 was used. The number of trees considered for RF was 100. The 'gini' and 'entropy' criteria are used to assess the quality of RF and DT.

DL Models: This work explored LSTM, BiLSTM, and CNN. This work exploited all possible combinations of word embedding and DL techniques, creating nine models.

- **CNN:** The embedded features from embedding layers is transmitted to the convolution layer, which has 64 filters with a kernel size of 7. The output of the convolution layer is passed to a max-pooling layer, and then a fully connected layer (64 neurons). In the relevant layers, 'ReLU' activation is employed. Finally, the probability distribution of the five classes was computed using an output layer with a 'sigmoid' activation function.
- **LSTM:** Three consecutive layers, an embedding layer, an LSTM layer, and a dense layer, are used. The 'sigmoid' activation function provided the highest result, while the 'Adam' optimizer was employed.
- **BiLSTM:** It includes an embedding layer, a BiLSTM (32 units), and a fully connected layer (16 neurons). The 'sigmoid' activation function was used in the output layer.

Table II describes the optimal hyperparameter values used to develop this CNN model.

V. Experiments

We experimented using the Google Colaboratory platform and the Python 3.7.13 module. Pandas 1.8.0 and numpy 1.21.6 were used for data preparation. Scikit-learn 1.0.2 was used to implement all models in ML, whereas Keras 2.8.0 and TensorFlow 2.8.2 were used to train DL models. In this study, the weighted f_1 score (WF1) was used to assess the effectiveness of the models. But weighted precision (WP) and recall (WR) were also provided for comparison of the results.

A. Results

Table III demonstrates the performance of ML models. With the best WF1 of 0.82, LR with CountVectorizer beats all other ML models.

Table IV demonstrates the performance of the employed DL models. The results indicate that the CNN model with GloVe embedding achieved the highest WF1 score (0.88) among all the ML and DL models. CNN with GloVe outperformed other DL models, likely because CNNs are adept at identifying local patterns in text, such as specific words or phrases that suggest aggression, whereas GloVe embeddings capture word meanings and contexts. Together, they enable the model to accurately and efficiently detect aggressive comments.

B. Error Analysis

Error analysis of the best-performing model is conducted in both quantitative and qualitative ways to better understand its performance.

Quantitative Analysis: Figure 2 illustrates the confusion matrix of the best-performed model (CNN+GloVe).

Figure 2a revealed that the model incorrectly identified 38 instances (out of 383) as others and 28 instances (out of 417) as ReAG texts, respectively. Figures 2b and 2d show that the accuracy of the PoAG and GeAG is 91% and 84%, which is also good. However, for the VeAG class, TNR and TPR are very high (Figure 2c), and for the RaC class, TPR and TNR are pretty low (2e). The model may not have had access to enough samples for learning, which prevented it from identifying the suitable class during testing.

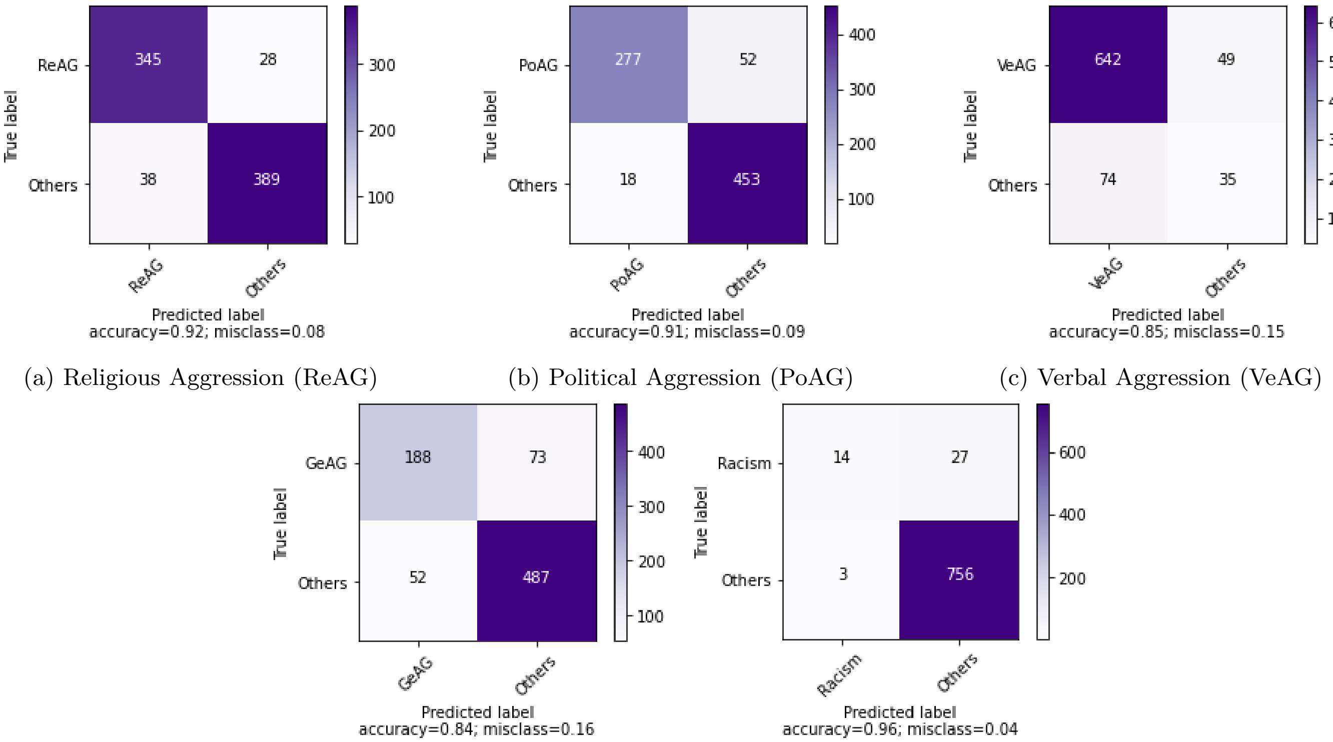
Qualitative Analysis: Table V depicts some sample predictions by the suggested model, showing that Samples 1, 2, and 5 were predicted correctly. According to the

Table III: Performance of ML models

Model	**Features**	**ReAG**	**PoAG**	**VeAG**	**GeAG**	**RaC**	**MF1**	**WF1**	**WP**	**WR**
LR	Count Vectorizer	0.86	0.76	0.93	0.68	0.34	**0.71**	**0.83**	**0.84**	**0.82**
LSVC		0.84	0.74	0.91	0.64	0.35	0.69	0.80	0.81	0.80
MNB		0.83	0.74	0.92	0.63	0.29	0.68	0.80	0.82	0.80
DT		0.81	0.67	0.90	0.62	0.37	0.67	0.78	0.78	0.78
RF		0.84	0.73	0.93	0.68	0.31	0.70	0.82	0.83	0.81
LR	TF-IDF	0.81	0.75	0.93	0.65	0.12	0.65	0.80	**0.84**	0.80
LSVC		0.79	0.73	0.93	0.67	0.24	0.67	0.80	0.82	0.80
MNB		0.78	0.70	0.93	0.43	0.00	0.57	0.75	0.79	0.77
DT		0.77	0.64	0.91	0.59	0.16	0.61	0.76	0.77	0.76
RF		0.82	0.73	0.93	0.68	0.25	0.68	0.81	0.84	0.81

Table IV: Performance of DL models

Model	**Embedding**	**ReAG**	**PoAG**	**VeAG**	**GeAG**	**RaC**	**MF1**	**WF1**	**WP**	**WR**
CNN	Keras	0.82	0.83	0.84	0.70	0.30	0.70	0.80	0.81	0.80
LSTM		0.56	0.36	0.86	0.39	0.00	0.43	0.61	0.60	0.63
BiLSTM		0.85	0.86	0.92	0.71	0.42	0.75	0.85	0.87	0.84
CNN	Glove	0.92	0.89	0.92	0.76	0.43	**0.79**	**0.88**	**0.89**	**0.87**
LSTM		0.82	0.83	0.92	0.72	0.00	0.66	0.82	0.80	0.81
BiLSTM		0.83	0.82	0.93	0.70	0.08	0.67	0.83	0.77	0.82
CNN	FastText	0.89	0.96	0.92	0.75	0.20	0.72	0.86	0.87	0.86
LSTM		0.82	0.81	0.92	0.76	0.00	0.66	0.84	0.82	0.84
BiLSTM		0.87	0.82	0.93	0.80	0.17	0.72	0.85	0.87	0.85



(a) Religious Aggression (ReAG)

(b) Political Aggression (PoAG)

(c) Verbal Aggression (VeAG)

(d) Gender Aggression (GeAG)

(e) Racism (RaC)

Figure 2: Confusion matrix of each class for the CNN+GloVe model

analysis, the sentences indirectly reflect aggressiveness, making it difficult for the model to identify multiple classes simultaneously (Samples 3 and 4). Specific terms have been frequently used in the fine-grained classifications. Perhaps these terms make it more difficult for the model to differentiate between the classes, thereby increasing the task's complexity.

VI. Conclusion

This study presented a DL-based method for detecting multilabel aggressive comments in Bengali. Various ML and DL methods have been investigated using differ-

Table V: Some correctly and incorrectly classified samples by the proposed model (CNN+GloVe)

Text	**Actual**	**Predicted**
Sample 1: মহিলা রাজনীতিবিদ দের ক্ষমতায় থাকার যোগ্যতা এবং অধিকারধ" একটাও নেই । (A woman politician has no right and ability to be in power.)	GeAG, PoAG	GeAG, PoAG
Sample 2: ফের যদি এইসব ভিডিও বানাস তোর চৌদ্দ গুষ্টিরে তোর বাবরি মসজিদের সাথে পুঁতে ফেলবো। (If you continue to make this kind of video, then I will bury your whole race under ground with your Babori mosque.)	VeAG, ReAG	VeAG, ReAG
Sample 3: কালার ঘরের কালা তুই ধর্মের কি বুঝিস, এইসব ভণ্ডামি ছাইড়া দিনের রাস্তায় আয়। গাধা কোথাকার। (You black piece of [expletive]. What do you know about religion? Stop this hypocrisy. Come to senses, you donkey.)	RaC, ReAG, VeAG	RaC, VeAG
Sample 4: লড়াই শুরু হলে পালানোর রাস্তা হায়িয়ে ফেলবি নাস্তিক গোষ্ঠী। (If the fight starts, you atheists will lost your way.)	ReAG, VeAG	ReAG, VeAG

ent feature extraction techniques to perform the downstream tasks. The evaluation results indicated that the CNN+GloVe model surpassed the other methods (0.88 weighted f_1 score). This work created a corpus (M-BATC) comprising 4002 Bengali text comments, each labeled with one of five classes. Future work includes expanding the dataset with more classes from other domains and exploring advanced techniques like hybrid DL models and BERT-based approaches to enhance generalizability and real-time deployment.

Acknowledgment

This work was supported by the Directorate of Research and Extension (DRE), Chittagong University of Engineering & Technology (CUET), Chittagong, Bangladesh.

References

[1] J. Culpeper, *Impoliteness: Using language to cause offence.* Cambridge University Press, 2011, vol. 28.

[2] R. Kumar, A. N. Reganti, A. Bhatia, and T. Maheshwari, "Aggression-annotated corpus of hindi-english code-mixed data," *arXiv preprint arXiv:1803.09402*, 2018.

[3] A. M. Ishmam and S. Sharmin, "Hateful speech detection in public facebook pages for the bengali language," in *Proc. IEEE ICMLA*. IEEE, 2019, pp. 555–560.

[4] O. Sharif and M. M. Hoque, "Identification and classification of textual aggression in social media: resource creation and evaluation," in *Proc. CONSTRAINT*. Springer, 2021, pp. 9–20.

[5] N. K. Al-harbi and M. Alghieth, "Fine-tuning arabic and multilingual bert models for crime classification to support law enforcement and crime prevention." *International Journal of Advanced Computer Science & Applications*, vol. 16, no. 5, 2025.

[6] S. Madisetty and M. S. Desarkar, "Aggression detection in social media using deep neural networks," in *Proc. TRAC*, 2018, pp. 120–127.

[7] M. O. Ibrohim and I. Budi, "Multi-label hate speech and abusive language detection in indonesian twitter," in *Proc. of 3rd Workshop on Abusive Language Online*, 2019, pp. 46–57.

[8] P. Mathur, R. Shah, R. Sawhney, and D. Mahata, "Detecting offensive tweets in hindi-english code-switched language," in *Proc. of SocialNLP*, 2018, pp. 18–26.

[9] A. K. Das, A. Al Asif, A. Paul, and M. N. Hossain, "Bangla hate speech detection on social media using attention-based recurrent neural network," *J. of Int. Sys.*, vol. 30, no. 1, pp. 578–591, 2021.

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

[11] R. Kumar, A. K. Ojha, S. Malmasi, and M. Zampieri, "Evaluating aggression identification in social media," in *Proc. TRAC*, 2020, pp. 1–5.

[12] M. N. Hoque and M. H. Seddiqui, "Detecting cyberbullying text using the approaches with machine learning models for the low-resource bengali language," *Int J Artif Intell ISSN*, vol. 2252, no. 8938, pp. 358–367, 2024.

[13] S. Ghosal and A. Jain, "Hatecircle and unsupervised hate speech detection incorporating emotion and contextual semantics," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 4, pp. 1–28, 2023.

[14] N. Islam, R. Haque, P. K. Pareek, M. B. Islam, I. H. Sajeeb, and M. H. Ratul, "Deep learning for multi-labeled cyberbully detection: Enhancing online safety," in *2023 International Conference on Data Science and Network Security (ICDSNS)*. IEEE, 2023, pp. 1–6.

[15] J. Cohen *et al.*, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[16] S. T. Zhang, F. F. Wang, F. Duo, and J. L. Zhang, "Research on the majority decision algorithm based on wechat sentiment classification," *J. of Int. & Fuzzy Sys.*, vol. 35, no. 3, pp. 2975–2984, 2018.

[17] F. Bogale Gereme and W. Zhu, "Fighting fake news using deep learning: Pre-trained word embeddings and the embedding layer investigated," in *Proc. 3rd CIIS*, 2020, pp. 24–29.