2022 26th International Conference on Pattern Recognition (ICPR)
August 21-25, 2022, Montréal, Québec, Canada

Cyberbullying Detection in Code-Mixed Languages: Dataset and Techniques

Krishanu Maity
Indian Institute of Technology Patna
Bihar, India
Email: krishanu_2021cs19@iitp.ac.in

Sriparna Saha
Indian Institute of Technology Patna
Bihar, India
Email: sriparna@iitp.ac.in

Pushpak Bhattacharyya
Indian Institute of Technology Bombay
Bombay, India
Email: pb@cse.iitb.ac.in

2022 26th International Conference on Pattern Recognition (ICPR) | 978-1-6654-9062-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICPR56361.2022.9956390

***Abstract*—The advent of the Internet is a boon to society. However, many of its banes cannot be undermined, and cyberbullying is one of them. In this work, we have created a benchmark corpus for cyberbullying detection in code-mixed languages. In India, most communications on different social media platforms are based on Hindi and English languages, and language switching is a common practice in digital communication. To investigate how code-mixed data can be handled effectively, BERT language model and VecMap based bilingual embedding along with a two-channel convolutional neural network model, namely BERT+VecMap-CNN, have been used. The input to one channel is the BERT language model and that to the other is the bilingual word embedding based on VecMap. As a baseline, we used standard machine learning models, as well as deep neural network models like CNN and LSTM. Our proposed model outperforms the baselines with overall accuracy and F1-measure values of 81.12%, and 81.03%, respectively. Furthermore, a different benchmark code-mixed dataset has been considered to show the robustness of our proposed model.[1]**

***Index Terms*—Cyberbullying, Code-Mixed (Hindi-English) dataset, Bilingual Word-Embedding, VecMap, MuRIL-BERT.**

I. Introduction

Cyberbullying [1] is described as the intentional, serious and repetitive acts of a person's cruelty towards others using various digital technologies. This further instigates several adverse effects on victims, such as depression, hopelessness, psychosomatic problems, loss of self-esteem; also attempts, or actual suicide are seen in users. Previous research has found that nearly 43% of teens in the United States have been victims of cyberbullying [2]. Different studies have reported that cyberbullying affects between 10 to 40 percent of internet users [3]. State-of-the-art research mainly focuses on cyberbullying detection from the English language [4]. Indigenous languages have received little attention due to a lack of appropriate datasets.

In social media, aggression[5] is directed at a specific person or group in order to harm their identity and diminish their standing and reputation . Aggression also contains Hate Speech. Hate speech [6] is any communication that disparages a person or group on the basis of a characteristic such as color, gender, race, sexual orientation, ethnicity, nationality, religion, or other features. Whereas cyberbullying is more specific to the personal attack rather than the attack on any kind of community or class, as mentioned in hate speech. So in short, aggression is the more general form of attack, it could be cyberbullying or hate speech.

The process of seamlessly switching between two or more languages in a discussion is known as code-mixing(CM) [7]. This is mainly observed in multilingual countries like Russia and India [8] and this is often correlated with casual conversations like in chat or one-on-one conversations. The contingency of code-mixing is increasing very rapidly. Over 50M tweets were analyzed by [9], in which 3.5% of tweets are code-mixed. Hence this should be our primary focus at this point of time. Choudhury et al. [10] demonstrated that the language employed on these social media sites, which is the source of most code-mixed text, varies from that found in books.

One of the most challenging problems with the code-mixed data is noisiness (spelling variations, short-form) and not following any specific linguistic rules and standards [11]. The majority of the works that have been carried out so far on cyberbully detection are based on monolingual language. According to the National Crime Records Bureau data, incidents of cyberbullying in India surged by 36% from 2017 to 2018[2]. India is a country with the second largest internet users (734 million[3]) and huge language diversity. In India, Hindi, English, and Hinglish make up the majority of text interactions on social networking platforms. The depiction of Hindi language in Roman form is known as Hinglish.

The primary purpose of this paper is to establish a cyberbully identification strategy in code-mixed data, which is commonly used by users in a multilingual country. To the best of our knowledge, there is only one publicly available Hindi-English code-mixed corpus for cyberbullying detection against "CHILDREN" and "WOMEN" only [12]. In this work, we have created a new general purpose Hindi-English code-mixed labeled (bully/Non-bully) dataset for cyberbullying detection. Unlike the existing dataset [12], in this paper, the introduced corpus for cyberbullying detection is not gender-specific and the domain has been enhanced by incorporating different entities like "Politics", "Ethnicity", "Education", "Sports", "Movies" etc. Another novelty of this corpus is the inclusion

[1]Code available at https://github.com/MaityKrishanu/ICPR

[2]https://ncrb.gov.in/en/crime-india-2018-0

[3]https://en.wikipedia.org/wiki/List_of_countries_by_number_of_Internet_users

of a harmfulness score (HC) on a three-point scale (0, 1, 2) for each tweet.

In this work, we have proposed a two-channel BERT+VecMap-CNN framework to represent code-mixed data efficiently. The first channel uses BERT language model [13], pre-trained on Books Corpus (800M words) [14] and English Wikipedia[4] (2,500M words). BERT has different variances like medical BERT, base model, MuRIL, etc. We utilized MuRIL BERT[5] for our experiment, which was pre-trained on 17 Indian languages and their transliterated equivalents. VecMap, a multilingual word embedding mapping technology developed by Artetxe et al. [15], is used in the second channel. The basic idea is to train the embeddings of source and destination languages individually using monolingual corpora, then align them in a common vector space where comparable words are grouped together using a linear transformation matrix.

Further, we have considered a benchmark aggression dataset [16] consisting of 15,000 Facebook Posts and Comments written in Hindi-English code-mixed language to analyze our proposed model's robustness on other code-mixed datasets.

The following are the primary contributions of this work:

1) We developed a Hindi-English code-mixed annotated dataset for detecting cyberbullying.
2) We have examined the BERT language model and Bilingual embedding using the VecMap technique to investigate how effectively they can handle code-mixed data.
3) We have proposed the two-Channel Convolutional Neural Network Model, namely BERT+VecMap-CNN, where one channel's input is the BERT language model whereas another is bilingual word embedding based on VecMap.
4) We have developed different baselines based on standard machine learning and deep learning models and our proposed model obtains overall accuracy and F1-measure values of 80.91%, 80.57%, respectively, on our new benchmark code-mixed cyberbully dataset.
5) We have also extended our experiment on another benchmark Hindi-English code-mixed aggression dataset to check the robustness of our model and examined that our proposed model surpasses state-of-the-art with a significant margin.

II. Related Works

With the availability of different social media sites and other information sharing platforms, cyberbullying is one of the major issues which needs to be addressed.

A. Works on Monolingual Datasets

Dinakar et al. [17] proposed an experimental work by applying binary classifiers on a corpus of 4,500 YouTube comments for cyberbullying detection. They obtained an overall accuracy of 66.70% with SVM classifier and 63% with Naive Bayes classifier. Reynolds et al.[18] worked on data collected from the Formspring.me and labeled using web service to train their model, they had used a Weka tool kit and were able to achieve 78.5% accuracy by using C4.5 decision tree learner. Djuric et al. [19] proposed a methodology for distributed low dimensional representations of comments using paragraph2vec and continuous BOW (CBOW) approach for hate speech detection. They tested their method on a vast data set of user comments collected from the Yahoo Finance website and found it 80.01% accurate. Balakrishnan et al. [20] developed a strategy for detecting cyberbullying for Twitter users based on psychological characteristics and machine learning approaches in 2020. They examined that considering personalities and sentiment features with baseline features (text, user, network) improves the cyberbullying detection task and achieves a sound accuracy of 91.7%.

B. Works on Code-Mixed Datasets

Kumar et al. [16] developed an aggression-annotated corpus containing 18k tweets and 21k Facebook comments written in Hindi-English code-mixed form. Bohra et al. [21] developed a code-mixed dataset of 4,575 tweets and annotated with hate speech and normal speech. SVM classier achieved 71.7% accuracy score when word n-grams, punctuations, character n-grams, hate lexicon and negation words are taken into account as feature vectors. Authors in [22] developed a multilayer perceptron model as the classifier. They also created a code-mixed corpus collected from Facebook and labeled it manually. They extracted various features from this data and obtained an accuracy of 68.50%. The authors in [23] proposed a deep learning based approach to identify hate speech from Hindi-English code-mixed corpus. With the help of domain specific word embedding, they outperformed the base model by 12% F1 score.

After a thorough literature survey, we observed that there is no work available for gender agnostics cyberbullying detection from Hindi-English code-mixed text. This motivates us to work in this specific domain.

III. Code-Mixed Cyberbully-Annotated Corpora Development

In this work, we have introduced a new code-mixed Hindi-English dataset to identify cyberbully.

A. Data Collection

With the help of Twitter streaming API[6] and Twitter Search API[7], we have collected tweets from Twitter. Using Twitter Search API, we can crawl historical tweets based on some specific keywords, while Twitter streaming API collects the real-time streaming data. Between March 2021 and June 2021, we have scraped roughly 75K raw tweets based on

[4] https://en.wikipedia.org/wiki/Wikipedia
[5] https://tfhub.dev/google/MuRIL/1
[6] https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data
[7] https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference/get-search-tweets

TABLE I: Some samples from our developed code-mixed cyberbully dataset

Tweets	HC	Class
T1: Tum to AliaBhatt k tara intelligent ho. **Translation**: You are as intelligent as AliaBhatt.	1	Bully
T2: Did you just call yourself a liberal? Tu gadhi hai gadhi, not liberal. **Translation**: Did you just call youself a liberal? You are sordid, not liberal.	2	Bully
T3:Itihas gawa hai har khubsurat ladki Kisi chutiya se he pyaar karte hai **Translation**: History says, every beautiful women likes brawler man.	1	Bully
T4: Bahut khubsurat lag rahe ho aap aur aap ki smile super hai ji **Translation**: You are looking beautiful and your smile is superb.	0	Non-bully

specified hashtags and keywords like r*ndi, MeToo, JusticeForSushantSinghRajput, IndiaAgainstAbuse, AliaBhatt, bitch etc. To extract only Hindi-English code-mixed tweets from 75K raw tweets, we utilised iNLTK [24], an open-source NLP tool for word label language identification, It's a pre-trained language model that supports tokenization, data augmentation, textual similarity, and text generation in 13 Indian languages, including Hindi and English.

B. Data Annotation

Three annotators having proficient linguistic background in both Hindi and English were involved in data annotation. We have given them detailed instructions of annotations with examples and closely monitor the annotation process. The term "closely monitored" indicates that we have randomly selected some samples during the annotation process to verify the quality of the annotated label. If we find some mistakes, we notify the corresponding annotator not to repeat the same mistakes. They have worked in isolation to avoid any biasness. For annotation, we followed the guidelines used in Van Hee et al. [25]. Annotators had tagged the cyberbully class (Non-bully/Bully) and the harmfulness score (HC) on a three-point scale (0, 1, 2) for each tweet. In Table I, there are some examples of annotated tweets. Score 0 signifies that there is no indication of cyberbullying and 1 indicates that the post contains indications of cyberbullying. However, they are not severe, and score 2 indicates that the post contains strong evidence of cyberbullying (e.g., physical threats or excitements to commit suicide). For finalizing the annotation label of each tweet, we use a majority vote procedure to resolve conflicts between annotators. We calculated the inter-annotator agreement (IAA) using Fleiss'[26] Kappa score to verify the quality of annotation. We attained the agreement scores of 0.81, 0.72 on the cyberbully class and HC, respectively. We have considered all the samples in our proposed corpus to calculate IAA scores.

C. Dataset Statistics

Our corpus contained 5,792 tweets, 2,975 categorized as non-bully, and 2,817 were tagged as bully. The percentages of non-bully and bully tweets in our corpus are 51.36% and 48.64%, respectively. Our corpus has 633 data points with a harmfulness score of 2, whereas the number of tweets having a harmfulness score of 1 is 2,184.

IV. Proposed Methodology

In this section, we have described the proposed methodology for cyberbullying detection in code-mixed languages. We have developed the two-channel convolutional neural network model, namely BERT+VecMap-CNN, where one channel's input is the BERT language model, and another is bilingual word embedding based on VecMap.

A. Proposed BERT+VecMap-CNN model

The proposed BERT+VecMap-CNN model has two channels. In the first channel, we have used BERT language model to get the embeddings of input tweets. On the other channel, we have used the Hindi-English bilingual align embedding matrix generated by the VecMap technique to represent the input text in numerical form. Let $X = \{x_1, x_2, ...x_n\}$ be the input sentence with n number of words. The input sentence is passed through two different channels with a series of operations which are described as follows:

1) **Channel 1:**
   a) **BERT:** We feed the input sentence X to BERT model. Here, we have considered only MuRIL BERT as it was pre-trained on Hindi-English language. Before passing X to BERT model, we need to add [CLS] token in the beginning and [SEP] token at the end of the sentence. The BERT model's final input is the concatenation of token embedding, segment embedding and position embedding.
   Pooled and sequence these two types of output have been generated by BERT. A $[batch\ size, 768]$ shaped pooled output represents the entire sentence using a special token called *CLS*. Let $W_B \in \mathbb{R}^{n \times d}$ be the sequence output obtained from the BERT model for input X, where n is the maximum sequence length and $d = 768$ is the dimension of each token. We have considered sequence output for our proposed model as in the next step, CNN will be applied. Pooled output has been taken into account for machine learning-based baselines.
   b) **CNN:** Convolution and pooling are the two primary components of CNN [27]. Convolution utilizes several filters to extract features (feature map) from data, while pooling layers is important for reducing the dimensionality of feature maps. The N-gram feature map is extracted by passing the output of the BERT model W_B through convolution

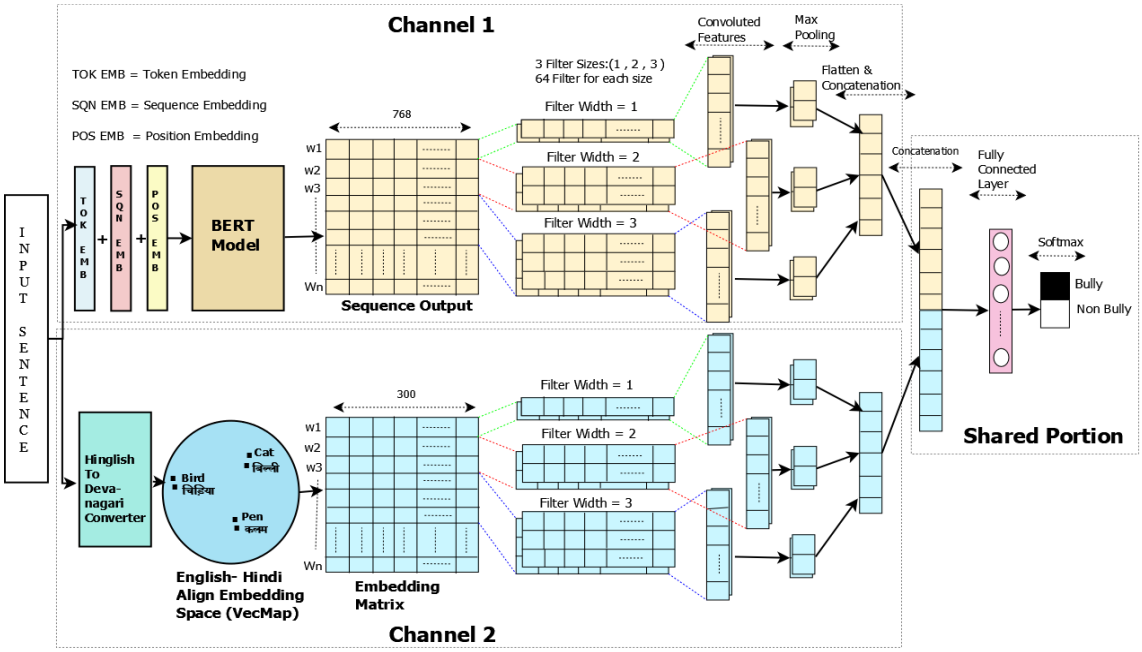

Fig. 1: Proposed Two Channel (BERT+VecMap-CNN) architecture.

layers. We have considered three different filters sizes 1, 2 and 3 for extracting unigram, bigram and trigram feature maps where the number of filters for each size being 64. To get the feature map, $\mathbf{c} \in \mathbb{R}^{\mathbf{n-k_1+1}}$ from filter $F \in \mathbb{R}^{k_1 \times d}$, an element-wise dot product over each possible word-window, $W_{j:j+k_1-1}$ has been performed. Each element c_j of feature map $\mathbf{c}$ is generated after convolution by

$$c_j = f(w_{j:j+k_1-1} * F_a + b) \tag{1}$$

Where f is a non linear activation function and b is the bias. Then we perform max pooling operation on $\mathbf{c}$. After applying t distinct filters of the same N-gram size, t feature maps will be generated, which can then be rearranged as

$$\mathbf{C} = [\mathbf{c_1}, \mathbf{c_2}, \mathbf{c_3}, ...\mathbf{c_t}] \tag{2}$$

We have concatenated all the feature vectors obtained from three different filter sizes and sent them to the next shared layer.

2) **Channel 2**:

a) **VecMap:** Fasttext [28] English and Hindi pre-trained word embeddings were used as mono-lingual embedding inputs to VecMap. We have converted Hinglish to Devanagari-Hindi by using the parallel dictionary developed by [29]. First, we have created a Hindi-English bi-lingual embedding vector using the VecMap technique. Then, this pretrained bi-lingual embedding has been utilized to generate the embedding vector of all the unique words present in the corpus. Now we have an embedding matrix of dimension $n \times 300$, an input to the next CNN layer.

b) **CNN:** We keep the same architecture of CNN as mentioned in channel 1. The only difference is that the embedding dimension d = 300. The concatenated output of the last layer of CNN is forwarded to the next shared layer of BERT+VecMAP-CNN.

3) **Shared Portion:** In the shared portion, a fully connected layer with 60 neurons is followed by a softmax layer with two outputs, i.e., bully and non-bully. The representations learned from both channels using CNN are concatenated and fed as an input to the next fully connected layer. As a loss function, we have used categorical cross-entropy $L(\hat{y}, y)$ to train the parameters of the network.

$$L_{CE}(\hat{y}, y) = -\frac{1}{N}\sum_{j=1}^{M}\sum_{i=1}^{N} y_i^j log(\hat{y}_i^j) \tag{3}$$

Where $\hat{y}_i^j$ is the predicted label and y_i^j is the true label. M and N represent the number of classes, and the number of samples, respectively.

The architectural representation of the proposed BERT+VecMap-CNN model is shown in Figure 1.

V. Experimental Results and Analysis

The findings of several baseline models and our suggested model are presented in this section. We have run all models five times, and the average results have been reported.

A. Dataset Details

During experimentation, we have considered two datasets. Dataset-1 is our newly created benchmark Hindi-English code-

mixed dataset for cyberbullying detection. Dataset-2 is a code-mixed aggression dataset developed by Kumar et al. [30]. Dataset-2 has 15,000 Facebook Posts/Comments in Hindi and English, where each post is labeled into one of the three classes, namely, *Non-aggressive(NAG)*, *Overtly Aggressive(OAG)*, and *Covertly Aggressive(CAG)*.We randomly chose 80% of the data for training, 10% for validation, and the remaining 10% for testing for both Dataset-1 and Dataset-2.

B. Hyperparameters

We use Tanh activation in CNN and ReLU activation in all fully connected layers ($d_f = 100$). With a batch size of 32, we train our models for 10 epochs. We utilize Adam optimizer and set the learning rate to 0.001 to backpropagate the loss across the network. We have experimented with different learning rates like 0.01, 0.001, 0.0001, 2×10^5, as mentioned in literature where authors have done text classification using BERT+CNN [31] or BERT+LSTM [32], [33] or simply BERT+FC [34] models. In our proposed model, we get the best results concerning a learning rate of 0.001.

C. Comparison with the Baselines

Following baselines are introduced for comparison with our proposed *BERT+VecMAP-CNN model* :**(1) TFIDF+SVM** : The TF-IDF (term frequency-inverse document frequency) features of input tweets have been fed to SVM classifier. Hyperparameters of SVM: kernel=linear, class weight=balanced, regularization parameter C=0.8, tolerance=1e-3; **(2) TF-IDF+LR** : TF-IDF features have been fed to the LR classifier. Hyperparameters of LR: class weight=balanced, penalty=l1, solver=liblinear, multi class=ovr; **(3) BERT+SVM** : MuRIL BERT's pooled output with dimension 768 was fed to SVM Classifier; **(4) BERT+ LR** : BERT generated pooled output of dimension 768 fed to LR classifier; **(5) BERT-finetune** : MuRIL BERT's pooled output with dimension 768 was fed to an softmax output layer; **(6) BERT+LSTM** : The LSTM layer with 128 hidden states received a sequence output generated by the BERT model. The LSTM layer's outputs are then sent to a softmax layer for prediction. Hyperparameters of this model are: loss=categorical cross-entropy, batch size=32, optimizer=Adam, dropout probability=0.5; **(7) BERT+CNN** : We have kept the same CNN architecture as mentioned in proposed model. This is basically the channel-1 + FC(60) + Softmax; **(8) VecMap+SVM** : VecMap generated embedding vector has fed to SVM classifier; **(9) VecMap+LR** : VecMap generated embedding vector has fed to Logistic Regression (LR); **(10) VecMap+LSTM**: VecMap generated embedding matrix is passed through the same LSTM architecture followed by a softmax as mentioned in Baseline-3; **(11) VecMap+CNN** : This is basically the channel-2 + FC (60) + Softmax.

D. Result Analysis on Dataset-1 (Cyberbullying)

Table II shows the results for all of the baselines and the proposed model on Dataset-1 in terms of F1-score and accuracy. From the result table, we can conclude that VecMap+SVM (Baseline-9) achieves higher accuracy (74.92%) than other machine learning-based baselines. MuRIL BERT with CNN (Baseline-8) outperforms the other deep learning based baselines with overall accuracy and F1-measure values of 78.14%, and 78.03%, respectively.

TABLE II: Evaluation results of cyberbully detection (Dataset-1)

Model	Accuracy	F1-score
TF-IDF + SVM (Baseline-1)	64.78 (+/-0.06)	64.57 (+/-0.04)
TF-IDF + LR (Baseline-2)	67.55 (+/-0.08)	67.25 (+/-0.07)
BERT + SVM (Baseline-3)	74.12 (+/-0.10)	74.03 (+/-0.11)
BERT + LR (Baseline-4)	73.39 (+/-0.15)	73.11 (+/-0.09)
BERT-finetune (Baseline-5)	76.86 (+/-0.75)	76.72 (+/-0.88)
BERT + LSTM (Baseline-6)	76.37 (+/-0.58)	76.12 (+/-0.55)
BERT + CNN (Baseline-7)	78.14 (+/-0.86)	78.03 (+/-0.87)
VecMap + SVM (Baseline-8)	74.92 (+/-0.04)	74.88 (+/-0.05)
VecMap + LR (Baseline-9)	71.45 (+/-0.15)	71.27 (+/-0.14)
VecMap + LSTM (Baseline-10)	77.15 (+/-0.68))	76.45 (+/-0.95)
VecMap + CNN (Baseline-11)	77.56 (+/-0.78)	77.07 (+/-0.75)
BERT+VecMap-CNN	**81.12 (+/-0.47)**	**81.03 (+/-0.52)**

From the table, it can be shown that the suggested BERT+VecMap-CNN model attained significantly better results than the other baselines. Compared to the best baseline, i.e., baseline-7, which is basically channel-1 with one fully connected layer followed by a softmax layer, the proposed BERT+VecMap-CNN attained almost 3% improvement. The joint optimization of channel-1(BERT+CNN) and channel-2 (VecMap+CNN) in the proposed model lead to the classifier's better performance and the gain in accuracy. VecMap with LSTM in baseline-10 achieved higher accuracy than MuRIL BERT with LSTM, i.e., Baseline-6. But on the other hand, MuRIL BERT with CNN outperforms VecMap embedded with CNN with an accuracy value of 0.58%. We have also examined that baseline-7 and baseline-11 (based on CNN) outperform baseline-6 and baseline-10 (based on LSTM) with accuracy values of 1.77% and 0.41%, respectively. In either of the above two cases, where we have applied a deep learning algorithm (LSTM or CNN) as a baseline, CNN achieves improvements compared to LSTM. One possible reason why CNN performs better than LSTM may be that noisiness in code-mixed data is generally higher than monolingual data. As we know, CNNs are performing well for extracting local and position-invariant features [35]. In contrast, RNNs are better for long-range semantic dependency-based tasks (machine translation, language modeling) than some local key-phrases. That's why in our proposed architecture, we have included CNN instead of LSTM. Another observation from the experimental results is that LR outperforms SVM when considering word label features (TF-IDF) as an input to the classifier. On the other hand, the scenario is reversed, i.e., SVM outperforms LR when deep learning-based pre-trained embedding (BERT and VecMap) is used for input representation. When we compare BERT with VecMap followed by either LSTM or CNN, LSTM performs better when embedded with VecMap than BERT. On the other hand, the BERT+CNN combination outperforms the VecMap+CNN combination.

TABLE III: Evaluation results of Aggression detection (Dataset-2)

Model	Accuracy	F1-score
TFIDF + SVM (Baseline-1)	50.78 (+/-0.07)	50.11 (+/-0.06)
TFIDF + LR (Baseline-2)	53.18 (+/-0.12)	52.95 (+/-0.15)
BERT + SVM (Baseline-3)	51.25 (+/-0.12)	51.13 (+/-0.15)
BERT + LR (Baseline-4)	52.34 (+/-0.14)	52.11 (+/-0.12)
BERT-finetune (Baseline-5)	56.36 (+/-0.89)	56.15 (+/-0.87)
BERT + LSTM (Baseline-6)	57.14 (+/-0.56)	57.11 (+/-0.55)
BERT + CNN (Baseline-7)	59.45 (+/-0.74)	59.33 (+/-0.75)
VecMap + SVM (Baseline-8)	56.46 (+/-0.21)	56.23 (+/-0.17)
VecMap + LR (Baseline-9)	54.86 (+/-0.17)	54.58 (+/-0.19)
VecMap + LSTM (Baseline-10)	55.19(+/-0.53)	55.02 (+/-0.57)
VecMap + CNN (Baseline-11)	57.12 (+/-0.63)	57.08 (+/-0.53)
BERT+VecMap-CNN	**62.12 (+/-0.56)**	**62.05 (+/-0.53)**

E. Analysis of Results on Dataset-2 (Aggression)

To further illustrate the efficacy of our proposed model, we have continued our experiments on another Hindi-English code-mixed dataset (Dataset-2). Table III shows the results for all of the baselines and the proposed model on Dataset-2 in terms of F1-score and accuracy. From table III, we have observed that our proposed model (BERT+VecMap-CNN) outperforms all the baselines with a significant margin. Compared to the best baseline, i.e., baseline-7, the proposed BERT+VecMap-CNN showed almost 3% improvements. We have also noticed that machine learning-based baselines (baselines- 8, 9 outperform baseline-1, 2) performed well after incorporating the VecMap technique for input data representation. On the other hand, deep learning-based baselines (baselines- 6, 7 outperform baselines-10,11) performed well when BERT was used for input data representation.

Table IV shows the results of all the state-of-the-art approaches on Dataset-2 (Aggression detection). As is evident from the table, our proposed model outperformed all other state-of-the-art techniques by a significant margin. All the reported results for the proposed model and baselines are statistically significant.

TABLE IV: Results of state-of-the-art models and the proposed model on Dataset-2

Model	F1-score
Dense Neural Network(Input-1024-512-256-Output) [36]	59.51
Ensemble method(CNN, LSTM, and Bi-LSTM) [37]	60.37
BERT+VecMap-CNN (Our model)	**62.05**

F. Error Analysis

We have manually checked those data instances for error analysis which were misclassified by the proposed *BERT+VecMap-CNN* model. We have considered the following examples for error analysis.

Example 1: Tum to Alia Bhatt k tara intelligent ho.
Translation : You are as intelligent as Alia Bhatt. Originally this sentence was labeled as bully, but our proposed model has predicted it as non-bully. Though there is no such profane or vulgar word present in Example-1, it is still indicated as bullying because it is trying to humiliate somebody based on intelligence. Our proposed classifier is unable to identify the underlying sarcasm of a sentence. The possible reason for this misclassification is the lack of sarcasm data in our dataset.

Example 2: Kiso ko paana Phir khona phir uski yaad me Randi Rona Agar yahi love h toh gaand maraye aisa love ka.
Translation : If love means crying for someone in remembrance after having and loosing somebody then I must say, I hate love. The proposed model had predicted example-2 as bully though its gold label is non-bully. Here, we can notice that example-2 has two vulgar words, *"Randi"*(whore) and *"gaand maraye"*(Ass Fucking) still, it is not bullying because that sentence does not humiliate someone. The possible reason for the wrong prediction could be the presence of multiple abusive or profane words in a sentence.

Example 3: Reha ki chut jayegi jail se?
Translation: Will Reha be released from jail. Example-3 is predicted as bully though it is a non-bully sentence. The reason for this misclassification could be that the model has been confused with the Hinglish word *"chut"*. In this example, *"chut"* means released, but in general, *"chut"(p*ssy)* is mostly used as a vulgar word in the Hindi language. As *"chut"* is frequently present in bullying tweets, so model considers the sentence as a bully.

VI. Conclusion

The requirement of cyberbullying detection is strongly determined by many of the researchers. In this study, we have created a benchmark Hindi-English code-mixed corpus annotated with Bully/Non-bully. Further, the severity of the cyberbullying post is also quantified by incorporating a *harmfulness* score into our dataset. We have examined the performance of fine-tuned MuRIL BERT with CNN and VecMap with CNN separately and noticed that they achieved overall accuracies of 78.14% and 77.56%, respectively, on our proposed dataset. We have proposed the two-channel BERT+VecMap-CNN model with the intuition that the joint optimization of channel-1 (BERT+CNN) and channel-2 (VecMap+CNN) leads to the development of a better classifier. The overall accuracy of 81.12% on the proposed dataset proves the effectiveness of the proposed technique. To further check our proposed model's robustness, we have conducted experiments on another benchmark Hindi-English code-mixed aggression dataset and examined that our model outperforms all other state-of-the-art approaches by a significant margin.

In future, we would like to build a pre-trained embedding for the Hinglish language as its use in India is increasing rapidly. As data annotation is very time consuming and error-prone work, we would like to develop a robust unsupervised learning approach for identifying cyberbullying.

Acknowledgement

Dr. Sriparna Saha acknowledges the support of ASEAN-India Science & Technology Development Fund (AISTDF) for conducting this research.

REFERENCES

[1] P. K. Smith, J. Mahdavi, M. Carvalho, S. Fisher, S. Russell, and N. Tippett, "Cyberbullying: Its nature and impact in secondary school pupils," *Journal of child psychology and psychiatry*, vol. 49, no. 4, pp. 376–385, 2008.

[2] C. Moessner, "Cyberbullying, trends and tudes," *NCPC. org*, 2014.

[3] E. Whittaker and R. M. Kowalski, "Cyberbullying via social media," *Journal of school violence*, vol. 14, no. 1, pp. 11–29, 2015.

[4] H. Rosa, N. Pereira, R. Ribeiro, P. C. Ferreira, J. P. Carvalho, S. Oliveira, L. Coheur, P. Paulino, A. V. Simão, and I. Trancoso, "Automatic cyberbullying detection: A systematic review," *Computers in Human Behavior*, vol. 93, pp. 333–345, 2019.

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

[6] J. T. Nockleby, "Hate speech in context: The case of verbal threats," *Buff. L. Rev.*, vol. 42, p. 653, 1994.

[7] C. Myers-Scotton, *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press, 1997.

[8] R. D. Parshad, S. Bhowmick, V. Chand, N. Kumari, and N. Sinha, "What is india speaking? exploring the "hinglish" invasion," *Physica A: Statistical Mechanics and its Applications*, vol. 449, pp. 375–389, 2016.

[9] S. Rijhwani, R. Sequiera, M. Choudhury, K. Bali, and C. S. Maddila, "Estimating code-switching on twitter with a novel generalized word-level language detection technique," in *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers)*, 2017, pp. 1971–1982.

[10] M. Choudhury, R. Saraf, V. Jain, A. Mukherjee, S. Sarkar, and A. Basu, "Investigation and modeling of the structure of texting language," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 10, no. 3-4, pp. 157–174, 2007.

[11] U. Barman, A. Das, J. Wagner, and J. Foster, "Code mixing: A challenge for language identification in the language of social media," in *Proceedings of the first workshop on computational approaches to code switching*, 2014, pp. 13–23.

[12] K. Maity and S. Saha, "Bert-capsule model for cyberbullying detection in code-mixed indian languages," in *International Conference on Applications of Natural Language to Information Systems*. Springer, 2021, pp. 147–155.

[13] 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.

[14] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 19–27.

[15] M. Artetxe, G. Labaka, and E. Agirre, "Learning bilingual word embeddings with (almost) no bilingual data," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 451–462.

[16] 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.

[17] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of textual cyberbullying," in *Proceedings of the International Conference on Weblog and Social Media 2011*. Citeseer, 2011.

[18] K. Reynolds, A. Kontostathis, and L. Edwards, "Using machine learning to detect cyberbullying," in *2011 10th International Conference on Machine learning and applications and workshops*, vol. 2. IEEE, 2011, pp. 241–244.

[19] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, "Hate speech detection with comment embeddings," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 29–30.

[20] V. Balakrishnan, S. Khan, and H. R. Arabnia, "Improving cyberbullying detection using twitter users' psychological features and machine learning," *Computers & Security*, vol. 90, p. 101710, 2020.

[21] A. Bohra, D. Vijay, V. Singh, S. S. Akhtar, and M. Shrivastava, "A dataset of hindi-english code-mixed social media text for hate speech detection," in *Proceedings of the second workshop on computational modeling of people's opinions, personality, and emotions in social media*, 2018, pp. 36–41.

[22] S. Ghosh, S. Ghosh, and D. Das, "Sentiment identification in code-mixed social media text," *ArXiv*, vol. abs/1707.01184, 2017.

[23] S. Kamble and A. Joshi, "Hate speech detection from code-mixed hindi-english tweets using deep learning models," *arXiv preprint arXiv:1811.05145*, 2018.

[24] G. Arora, "inltk: Natural language toolkit for indic languages," *arXiv preprint arXiv:2009.12534*, 2020.

[25] C. Van Hee, B. Verhoeven, E. Lefever, G. De Pauw, W. Daelemans, and V. Hoste, "Guidelines for the fine-grained analysis of cyberbullying," version 1.0. Technical Report LT3 15-01, LT3, Language and Translation ..., Tech. Rep., 2015.

[26] J. L. Fleiss, "Measuring nominal scale agreement among many raters." *Psychological bulletin*, vol. 76, no. 5, p. 378, 1971.

[27] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

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

[29] M. M. Khapra, A. Ramanathan, A. Kunchukuttan, K. Visweswariah, and P. Bhattacharyya, "When transliteration met crowdsourcing: An empirical study of transliteration via crowdsourcing using efficient, non-redundant and fair quality control." in *LREC*. Citeseer, 2014, pp. 196–202.

[30] 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.

[31] T. Saha, S. R. Jayashree, S. Saha, and P. Bhattacharyya, "Bert-caps: A transformer-based capsule network for tweet act classification," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 5, pp. 1168–1179, 2020.

[32] T. Saha, A. Upadhyaya, S. Saha, and P. Bhattacharyya, "Towards sentiment and emotion aided multi-modal speech act classification in twitter," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 5727–5737.

[33] ——, "A multitask multimodal ensemble model for sentiment-and emotion-aided tweet act classification," *IEEE Transactions on Computational Social Systems*, 2021.

[34] S. Paul and S. Saha, "Cyberbert: Bert for cyberbullying identification," *Multimedia Systems*, pp. 1–8, 2020.

[35] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," *arXiv preprint arXiv:1702.01923*, 2017.

[36] K. Raiyani, T. Gonçalves, P. Quaresma, and V. B. Nogueira, "Fully connected neural network with advance preprocessor to identify aggression over facebook and twitter," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, 2018, pp. 28–41.

[37] S. Madisetty and M. S. Desarkar, "Aggression detection in social media using deep neural networks," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, 2018, pp. 120–127.