Hateful Speech Detection in Public Facebook Pages for the Bengali Language

Alvi Md Ishmam[1] and Sadia Sharmin[2]
[1,2]Department of CSE, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh
Email: [1]1305092.am @ugrad.cse.buet.ac.bd, [2]sadiasharmin.ss@gmail.com

***Abstract*—Online hateful speech detection and classification in social media for the various major languages other than English has drawn the attention of researchers recently. In this paper, we develop Machine Learning (ML) algorithms based model, as well as Gated Recurrent Unit (GRU), based deep neural network model for classifying users' comments on Facebook pages. We have collected, annotated 5,126 Bengali comments and classified them into six classes – *Hate Speech*, *Communal Attack*, *Inciteful*, *Religious Hatred*, *Political Comments*, and *Religious Comments*. The produced corpus is the first contribution to the field of hateful speech detection in the Bengali language for social media. Finally, we employ several machine learning algorithms, compare the performance, and attained 52.20% accuracy in Random Forest. The accuracy is improved in the case of GRU based model (70.10% accuracy) about 18%.**

***Index Terms*—Annotated corpus, Hate speech, Facebook page, Bengali language, Machine Learning, Gated Recurrent Unit**

I. Introduction

Social Networking Sites (SNS) such as Facebook, Twitter, etc., have facilitated modern communications with a handful of services. On an average with 2.27 billion [1] active users, Facebook is the most preferable medium of the interaction of people. People talk, argue, and express opinions with people or communities by enjoying different features of Facebook. However, arguments or debate often turn towards intolerance of other's opinions, that often lead to abusing or hate speech towards a person, community, culture or religion.

Dhaka, the capital of Bangladesh, has the second-highest active Facebook users which are about 1.1% of total monthly active SNS users all around the world [2]. People engage in different groups and often create pages for promoting products and ideas. For example, Facebook page of *Sakib Al Hasan*,[3] one of the topmost celebrity players has about 1 million likes and followers. Besides, actors, players, political, cultural or social parties maintain Facebook pages for different purposes. Different studies indicate that these pages are potential sources for spreading hateful speech [8] on political, religious, cultural aspects. According to Counter-Terrorism and Transnational Crime (CTTC) cyber unit of Bangladesh, about 2500 Facebook pages are responsible for spreading communal hatred [4]. Moreover, Facebook pages are public and anyone can see the posts. Therefore, public pages have become the honeypots to spread hateful speech and creating unfavorable situations.

In Bangladesh, Facebook usage is prominent than any kind of SNSs and public pages are potential resources for spreading hate speech. In a nutshell, in this paper we have made the following contribution–

- We have developed an annotated data set of about 5126 comments from popular public pages. The details of the data set development and annotation are described here [12]. We have identified linguistic as well as some quantitative features for the mentioned six classes on the social context of Bangladesh. Linguistic features from existing state-of-the-art based on Twitter and the English language are not applicable to the social context of this region. According to the rigorous search on the internet, we have found no study regarding hate speech detection for the Bengali language on Facebook or any other social media. Our approach can be the first approach regarding identifying and detecting hate speech.
- We have compared the performance of a couple of well known supervised machine learning algorithms such as linear SVC, SVC, Adaboost, Naive Bayes, and Random Forest. We have attained about 52.20% accuracy in the case of Random Forest (90% train and 10% test data). To improve the performance, we develop a GRU based deep neural network model exploiting the word2vec [18] algorithm and attained 70.10% accuracy.

II. Related Work

Researchers' have shown interest in detecting hate speech and abusive content on social media and web blogs. Binary classification (positive, negative) related to spiritual belief (Christianity, Islam, Judaism) based on the Twitter data set have been explored in this study [24]. The authors have used different machine learning algorithms, compared the result, and found SVM is performing better for binary classification. In another paper [21], the authors have annotated 5,143 comments from Twitter and Facebook videos and gained the best performance using linear SVM. With 78% accuracy SVM performs best in classifying twitter comments into three classes (hate, offensive, normal) using character n-grams, word n-grams, and word skip-grams [16]. However, the authors did not clear the difference between hate speech and offensive speech, and the distribution of the annotated comments is

[1]https://www.statista.com/statistics/264810/number-of-monthly-active-Facebook-users-worldwide/

[2]https://bdnews24.com/bangladesh/2017/04/15/dhaka-ranked-second-in-number-of-active-facebook-users

[3]https://www.facebook.com/Shakib.Al.Hasan/

[4]https://www.dhakatribune.com/bangladesh/2017/11/16/hundreds-facebook-pages-spreading-communal-hatred-bangladesh

978-1-7281-4550-1/19/$31.00 ©2019 IEEE
DOI 10.1109/ICMLA.2019.00104

not uniform. In this study [10], four Convolutional Neural Network (CNN) models are employed to classify 6,655 twitter comments in English into four classes. The model was based on character n-gram with word2vector attained about 78.3 F-score. Crowd-sourcing based approached have been devised to label about 16k Twitter comments into three classes (Offensive, Hate Speech, Non-hate) in this study [5]. The authors ended up with that keyword-based hate content are easier to classify and used different machine learning algorithms (Logistic Regression, Naive Bayes, Decision Trees, Random Forests, and Linear SVM) to find a suitable model for classification. Furthermore, paragraph2vec for joint modeling of comments and words, using the continuous BOW (CBOW) neural language model has been designed to identify comments into binary classifications (hate or clean) [6]. However, the above-mentioned works based on Twitter data set are in English and these models are not applicable in our cases. Firstly, the definition and perspectives of hate speech, as well as other offensive content, are not the same in our cases. For example, *Muslim*, *Islam* are basic keywords for spreading religious or communal hatred in the West. On a note, ethnic minorities are the primary victim of the online hate [17]. On the other hand, হিন্দু (The Hindu) is used as hate content in many cases rather than মুসলিম (The Muslim) in our context. Therefore, social, cultural issues of geographical territories play a vital role in determining the peoples' behaviors and practices in social media.

A. Endeavour Other Than English

Beyond the English language-based studies, researchers' have worked on hate speech detection for major languages such as Italian, Arabic, Chinese, etc. An Italian corpus of $6,000$ tweets is annotated for hate speech against immigrants in Italian language [22]. In the case of Arabic language [19], tweets are annotated into three classes– obscene, offensive, clean. In this paper, a survey was accomplished among the higher educated German regarding how hate speech is classified by them [4]. The authors took two comments regarding a news post reporting the incidents on New Year's Eve and formulated questions on these comments. This can be considered an analysis of hate speech defining and identification in the German context. About 13,766 tweets have been gathered based on 10 hashtags on the refugee crisis in Germany in this paper [20].

B. Study of Bengali Language Based Detection on Facebook

In this study, 300 comments are collected from popular Facebook pages of celebrities [11]. These comments are annotated as abusive or non- abusive based on the survey on Facebook. Here, a couple of ML algorithms are used for classification purposes on 2500 comments [7]. Binary classification (Cyberbullying or not) was done through WEKA on 2400 Bengali tweets [1].

However, these studies do not clarify how they annotate these comments or what are the definitions they follow for annotating for the particular class. Secondly, apart from the small-sized dataset, the dataset is not described properly in the context of morphological, linguistic or any quantitative analysis as well as any feature-level analysis. Moreover, they did not describe any feature or data engineering methodology as well as the detailed description of dataset (how long sentences, word count, etc.)

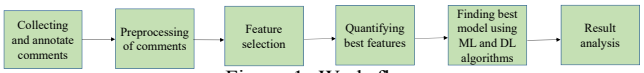


Figure 1: Work flow

III. Research Context and Methodology

The prior research works resemble the various methodology of hate speech detection. The authors have found that most of the works done on **Twitter** dataset based on **English** language (some works are done a couple of major languages like **Italic**, **Arabic**, etc). A report [5] indicates that about 2,500 pages are responsible for communal hatred in Bangladesh. Some recent studies also inspire the authors to look into the matter.

Unfortunately, Bengali language-driven hateful speech detection in any social media is yet to be accomplished except for some minor level sentiment analysis. However, no concrete dataset is available yet for Facebook or other social media. Therefore, we plan to gather publicly available datasets for hate speech research. We have accumulated posts, comments, and necessary metadata such as likes, reactions, replies of the commenters using Facebook Graph API from popular public pages under the privacy policy of Facebook. We have described the process in Section IV briefly. Moreover, the accumulated data set is unstructured, misspelled on Facebook. Therefore, we need to pre-process the collected data before further steps shown in Fig.1. After the pre-processing steps, we have employed five supervised learning algorithms (SVM, Linear SVC, Random Forest, Naive Bayes, Adaboost) including Gated Recurrent Unit (GRU) based neural network model using linguistic and quantitative features from users' comments. The features are following–

- N gram features, each weighted by its TF-IDF.
- character N gram feature
- Text quality: Flesch-Kincaid Reading Ease and Flesch Grade Level Score
- Count indicators for hashtags, mentions, and URLs.
- Total Number of characters in a word, words, syllables count in a comment and replies of a comment.

The first three are linguistic features where the latter two are quantitative features. Both types of features will be discussed later. The workflow is shown in Fig 1.

IV. Data Collection

Popular public pages are the best potential sources for spreading hateful speech towards particular people, culture, ethnicity or community. According to our investigation [2],

[5]https://www.dhakatribune.com/bangladesh/2017/11/16/hundreds-facebook-pages-spreading-communal-hatred-bangladesh

[9], popular pages of different classes such as celebrities, actors, political parties, politicians, and players are honeypots for haters to spread hateful speech. These pages have millions of likes and interact with thousands of people every day through the posts and comments. Therefore, we have selected at least one page from these criteria for dataset development initially. We cannot collect dataset from all pages rather select pages which are the most popular pages in terms of Bangladesh. We select **noyon chatterjee5**, **Basherkella**), **Awami League**, **Sakib Al Hasan** as sample pages for developing our corpus. These pages belong to popular Facebook celebrities, extremist political parties, the official page of the ruling party and the famous cricketer of Bangladesh. These pages have likes of about 0.086, 0.175, 2.5 and 10 million respectively and found on average 3 posts per day having 2k reactions and 80 comments per posts. We have used Facebook graph API (version 2.9) from 22 December 2017 to April 2018 and collected about 3000 comments from these pages along with various metadata such as replies of comments, reactions, etc. However, Facebook recent policies have reduced the access of Graph API for gathering metadata of pages. Therefore, from May 2018 to November 2018, we have manually collected about 2000 comments from these pages. We have classified the Facebook comments into hateful and non-hateful comments and these are later classified into six categories– *hate, inciteful, commmunal hatred, religious hate, political, religious*. The first four categories belong to the hateful speech and later two are non hateful speech. The process of development and annotation of the data set is described here [12]. The data set is available here [6].

V. Pre-processing of Accumulated Data

After the collection of comments, we process the raw text to remove bad characters, punctuations, stemming, etc. In our pre-processing steps, we have three parts–

1) Removal of bad characters, punctuations, etc.: After getting the raw text, we remove the punctuations such as (comma(,), semicolon (;), dash(–), hyphen (–), dot(.), question mark (?), exclamatory mark (!), etc.). We have also removed distorted, noisy, bad characters that do not support unicode encoding.

A. Tokenization of string

Our second pre-processing step is tokenization of the pure Bengali text. Tokenization is the process of demarcating and possibly classifying sections of a string of input characters.

B. Stemming of the tokens

Stemming is the process of reducing a word to its word stem. In Bengali, if we consider the word 'কর', 'করি, 'করলাম','করেছি','করব', all have the same semantic meaning 'কর' (do). Since Bengali is a highly inflected language, the authors of these papers [13], [15] propose a rule base stemming technique, where the suffix is stripped from higher to lower precedence. We have used the java-based stemmer [7] for finding the root form of words.

C. Stopwords Removal

We have used the stopwords list available here [8] and updated it with many words. Currently, the list consists of 440 words. We have updated the stopword list with different prepositions, quantifiers, linkers, conjunctions, etc. Moreover, there are many misspelled words used in social media. Therefore, we have incorporated frequently used misspelled formats of stopwords.

VI. Feature Selection

We have employed three linguistic and two quantitative features from generated pre-processed tokens. These are discussed below–

A. N gram features, each token weighted by its TF-IDF

The keyword is *N gram* token and *TF-IDF*. For example, bi-gram means consecutive two tokens. In our case, we have taken N-gram, which means N (3 or 5 tokens) consecutive tokens to be considered. The second keyword is *Term Frequency (TF)* and *Inverse Document Frequency (IDF)*. The formal definition of the two parameters are –

$$TF(t) = \frac{Number\_of\_times\_term\_t\_appears\_in\_document}{Total\_number\_of\_terms\_in\_the\_document}$$

$$IDF(t) = \log \frac{Total\_number\_of\_documents}{Number\_of\_documents\_with\_termt\_in\_it}$$

$$TF - IDF(t) = TF(t) \times IDF(t).$$

For example, if we say 'হিন্দুদের' (The Hindus), that does not resemble hate content, however, if we say 'হিন্দুদের বের করে দাও' (The Hindus should be banned from the country), this represent a communal hate towards the Hindu community. Therefore, we have determined the TF-IDF of each of the N-gram tokens (3 or 5 consecutive tokens) in which particular class (among the six class) it resembles most. For this purpose, we have used sklearn **TfIdfVectorizer** with n-gram modeling where n=3 or 5.

B. Text quality: Readability Scores

Readability means how comprehensive a content is to understand. Flesch-Kincaid Grade Level and Flesch Reading Ease scores [14] are widely accepted matrices to determine the readability of any text content.

In both cases, the higher the score, the better it is to understand the content. In hate speech detection on the Twitter data set, these two readability scores are taken as a feature [5]. Therefore, we were also inspired by the prior work of Davidson et al. [5] to find the readability matrices for the Bengali language. A statistical analysis shows that [23], Flesch Reading Ease and Flesch-Kincaid Grade-Level are not

[6]https://github.com/IshmamAlvi/Hate-Speech-for-Bengali-language

[7]https://lucene.apache.org/core/7_{10}/$analyzers$ – $common/org/apache/lucene/analysis/bn/BengaliStemmer.html$

[8]https://github.com/stopwords-iso/stopwords-bn

applicable for the Bengali language and a new model has been devised based on regression analysis. The following model has been formulated– $-5.23 + 1.43 \times AWL + .01 \times PSW$ and $1.15 + .02 \times JUK - .01 \times PSW30$

Here, AWL = Average Word Length, Number of Polysyllabic Words, PSW = Polysyllabic words are the words whose count of syllable exceeds 2. Number of Jukta-akshars (JUK): jukta-akshar or consonant-conjunct is consonants occurring together in clusters. When a consonant with a halant (hasanta) is followed by another consonant, we consider it as one jukta-akshar.

C. *Hashtags*, *Mentions*, and *URLs*

We have also considered some quantitative features. In Twitter, hashtag(#), annotations (@) are important for any classification task, since these symbols are used especially to mention something. In spite of, these features are not used as widely on Facebook, sometimes hashtags or annotations or URL may be used to mention some bad words, offensive or abusive contents.

D. *Number of characters, words, and syllables in a string or comment*

We are employing some other quantitative features for feature selection.

- *Comment Length* Comment length means the total number of tokens in a comment. We find that sometimes very long comments or very short comments tend to spread different hateful contents. Therefore, comment length might be a feature in this aspect.
- *Word Length* We have also taken word length as well as average word length as our features.
- *Average Syllables* We have used the average syllables of a particular word in some cases. The average syllable length of words in a comment can be a feature for this study.

VII. Model Selection

After the pre-processing steps and features selections, we need to feed our data and features into suitable models. We are planning to use popular algorithms such as Linear or Logistic Regression, Random Forest, Support Vector Classifier (SVC), Linear SVC, Naive Bayes. The objective is to find the most suitable model and compare among the algorithms.

A. Reduction of the Dimensionality of Data

Feature selection can be an important part of the machine learning process as it can greatly improve the performance of our models. We have used a couple of algorithms for this purpose.

1) Logistic Regression and Principle Component Analysis: We first use logistic regression with L1 regularization to reduce the dimensionality of the dataset. We have generated the topmost 60 features from 4000 features with this regression. In the case of Principle Component Analysis (PCA), we get a similar type of result alike logistic regression.

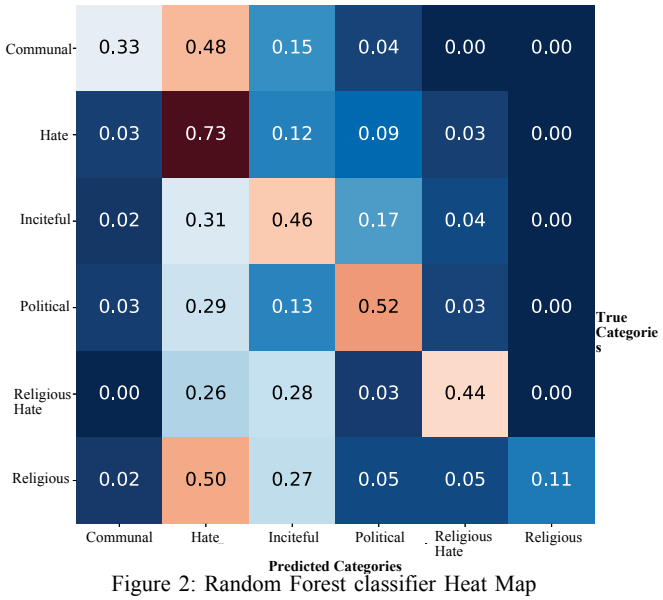


Figure 2: Random Forest classifier Heat Map

Table I: Random Forest

class	precision	recall	f1 score
CommunalAttack	0.45	0.37	0.41
HateSpeech	0.48	0.75	0.59
Inciteful	0.48	0.46	0.47
Political	0.62	0.52	0.57
Religious	0.59	0.44	0.50
ReligiousHatred	1.00	0.14	0.24
avg / total	0.57	0.52	0.51

2) SGDclassifier: Stochastic Logistic Gradient (SGD) classifier is also employed for topmost features selection. However, it cannot reduce the dimensionality at an expected level. It generates about 9800 features which increase the dimensionality of the data, as a result, it increases the chances of getting overfitted. Therefore, we are not interested in this classifier.

B. Performance of Classifier Algorithms

After defining the topmost features, we test with a variety of classifiers to find a suitable model for training. We have taken accuracy, precision, recall and f1 score as performance measurement matrices to compare the performance of the algorithms. In table II, we have compared the performance (precision, recall, f1-score, accuracy) of the ML algorithms and GRU model.

In Table I, we have incorporated the performance of the Random Forest with the mentioned matrices where the train and test data set were 90% and 10% respectively.

C. Test Accuracy of Classifier Algorithms

The test accuracy for ML algorithms are shown in Table II. We find that Random Forest gives the best accuracy over test data and it is about 52.20%. Adaboost and Naive Byes are not good classifiers since having poor accuracy on test data. In Fig. 2, indicates the heat map of Random Forest, which performs best.

Table II: Precision, Recall , F1 score, Accuracy, of different classifier algorithms

Algorithm	Precision	Recall	F1 score	Accuracy
SVC	0.50	0.48	0.47	0.48
Linear SVC	0.52	0.51	0.50	0.50
Naive Bayes	0.53	0.25	0.24	0.24
Random Forest	0.56	0.52	0.50	0.52
Adaboost	0.43	0.39	0.37	0.39
GRU based model	0.68	**0.70**	**0.69**	**0.70**

Table III: Feature ablation study: F-scores calculated on each algorithm when one feature is removed at a time

Algorithm	RF	Linear SVC	SVC	NB	Adaboost
F1 score	50	**51.36**	45	21	37
token n gram	-10.69	**-15.23**	-7.13	-9.49	-9.45
char n gram	-3	**-3.36**	-1.57	-2.5	-3.1
Readability matrices	+2	**+2.51**	+2.1	+1.25	+1.36
URl, mentions, hashtags	0	0.01	0.01	0.02	0
word count, unique word count, char count	+2	**+2.94**	+2.1	+ 1.91	+2.63

D. Neural Network Model

In Fig. 3, we show the architecture of the GRU based neural network. We use GRU with the word Embedding model word2vector. We go for GRU instead of LSTM for faster training and achieve good performance over small dataset i.e GRUs may have better ability to generalize and less tendency to overfit on small datasets [3]. To implement word2vector, we use gensim. The longest comment in our dataset has 377 tokens. Then, comments are converted into the sequences of integer indices and these sequences are padded with zeros so that they have an equal length of 377. Then, the sequences are fed into an embedding layer with an output dimension of 1000. The output of the embedding dimension (377, 1000) is then fed into a dropout layer with a rate of 0.2. The layer is used as regularization to avoid overfitting. The GRU layer with 300 hidden layers are followed by another dense layer with a dimension of 64 having a dropout rate of 0.2. Then the output is fed into a dense layer with a softmax activation function. We perform training in batches of size 32, and we used 'adam' as our optimizer.

We evaluate the experiment ten times for each configuration and report the average performance. In each iteration, we reserve 10% of our data for testing and the rest is further divided into 90% train and 10% validation set for fine-tuning the hyperparameters (dropout rate, hidden layers) .

VIII. Results and Discussion

We have used 5 ML algorithms and attain around 51% accuracy at best (precision=56%, recall=52%, f1 score=50%) in case of Random Forest. Then, we go for the GRU based neural network model, where we attained about 70% accuracy (precision=68%, recall=70%, f1 score=69%)) with a loss of around 38%.

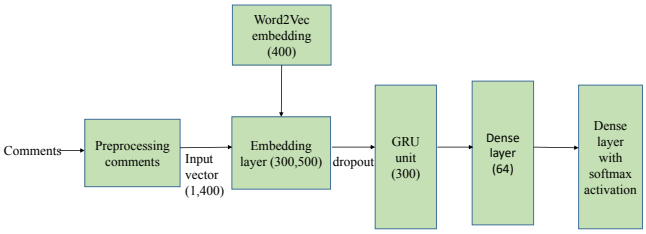


Figure 3: Architecture of GRU based neural network with word embedding

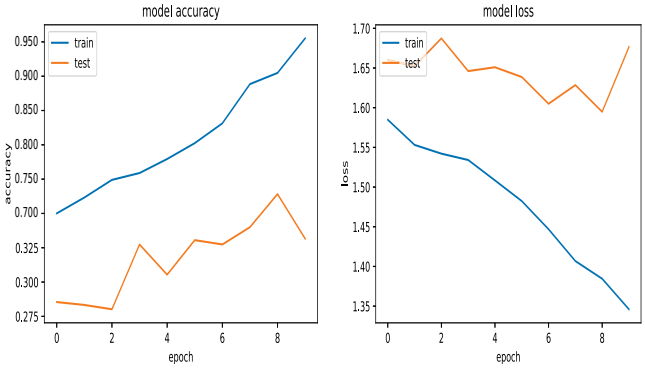
A. Feature ablation

In Table III, we have mentioned f1 score after the feature ablation for ML algorithms. We have divided our features into linguistic and quantitative features. Hashtags, URL counts, and mentions are not important features on the f1 score. Unlike Twitter, hashtags (#) and mentions (@) are not used widely on Facebook. Removing these features hardly affects the f1 scores. Removing the readability score from the feature set improves the scores (+2.51 in the case of Linear SVC). Moreover, discarding other quantitative features such as word count, unique word count, char count improve the scores. Finally, token n-gram and character n-gram with Tf-IDF have a significant contribution. Removing these can drop around -15.23 and -3.36 respectively in case of linear SVC.

B. Performance comparison between Neural Network Model and ML algorithms

In the case of GRU based model with word2vector, we have achieved 70.21% accuracy. From Fig. 2, it shows, religious and communal attack are misclassified as hate speech. It causes a downfall on the f1 score. Less training data (less than 1000 comments in case of inciteful, communal attack, religious) and low inter-annotator agreement is a factor for this. Moreover, religious hatred and communal attack constitute words or phrases, which are closely related to a religious comment or hate speech. Therefore, the model fails to determine a semantic meaning to some extent. In other cases of hateful content (inciteful or religious hatred), most of the misclassifications are recognized as hate speech. Most importantly, the rate of misclassification of hateful content (4 class) as non-hateful content (2 class) is very low excluding the religious class.

Moreover, the GRU based model develops better understandings of context and semantics of words and has better performance in terms of accuracy, precision, recall, and f1 score than other algorithms. Unlike the n-gram model, the word embedding model can grow the semantic relationship among words. Word embeddings allow semantically similar words to have dense vector representation. The higher agreement rate can help to improve the misclassification cases. Besides, getting more training data (especially religious and communal attack) might leap up the performance to some extent. In Fig. 4, epoch vs accuracy and epoch vs loss is shown for the GRU model. From Fig. 4a, we have attained maximum validation



(a) Epoch Vs Accuracy (b) Epoch Vs Loss

Figure 4: Effect of epoch number on train and validation accuracy and loss

accuracy of 70.21% in epoch 8, train accuracy is more than 90% at epoch 10. From Fig. 4b, the loss for the validation set increases while increasing the epoch number. Therefore, we confine our epoch number to 8 to attain optimum accuracy, precision, recall, and f1 score by avoiding overfitting.

IX. Conclusion

Hateful speech detection in online social media for various major languages are challenging task because of the diversity of the languages and usages pattern of the users. However, hateful speech detection in any type of social media for the Bengali language is not embraced in any study. In this paper, we have introduced the methodology of data set collection, annotation process form the social context of Bangladesh and divided them into six classes. We have generated the topmost uni-gram, bi-gram and tri-gram features for each class. Besides, hateful content, we have classified the non-hateful content into two classes (political or religious), that represents a more rigorous investigation of people's usage pattern on Facebook. Finally, we have investigated different ML algorithms as well as a deep neural network model to find suitable models and compare the performance of the algorithms. Our data set can be enhanced for further investigation and the final result can be used for the benchmark for future endeavor.

References

[1] Shahin Akhter et al. Social media bullying detection using machine learning on bangla text. In *2018 10th International Conference on Electrical and Computer Engineering (ICECE)*, pages 385–388. IEEE, 2018.

[2] Anat Ben-David and Ariadna Matamoros-Fernández. Hate speech and covert discrimination on social media: Monitoring the facebook pages of extreme-right political parties in spain. *International Journal of Communication*, 10:1167–1193, 2016.

[3] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[4] Tuba Ciftci, Liridona Gashi, René Hoffmann, David Bahr, Aylin Ilhan, and Kaja Fietkiewicz. Hate speech on facebook. In *Proceedings of the 4th European Conference on Social Media, ECSM 2017*, pages 425–433, 2017.

[5] Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515, 2017.

[6] Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, pages 29–30. ACM, 2015.

[7] Shahnoor C Eshan and Mohammad S Hasan. An application of machine learning to detect abusive bengali text. In *2017 20th International Conference of Computer and Information Technology (ICCIT)*, pages 1–6. IEEE, 2017.

[8] Johan Farkas, Jannick Schou, and Christina Neumayer. Cloaked facebook pages: Exploring fake islamist propaganda in social media. *New Media & Society*, 20(5):1850–1867, 2018.

[9] Johan Farkas, Jannick Schou, and Christina Neumayer. Cloaked facebook pages: Exploring fake islamist propaganda in social media. *New Media & Society*, 20(5):1850–1867, 2018.

[10] Björn Gambäck and Utpal Kumar Sikdar. Using convolutional neural networks to classify hate-speech. 2017.

[11] Md Gulzar Hussain, Tamim Al Mahmud, and Waheda Akthar. An approach to detect abusive bangla text. In *2018 International Conference on Innovation in Engineering and Technology (ICIET)*, pages 1–5. IEEE, 2018.

[12] A. M. Ishmam, J. Arman, and S. Sharmin. Towards the development of the bengali language corpus from public facebook pages for hate speech research. In *Asian HCI Symposium*. ACM, May 2019. doi: 10.1145/3309700.3338457, ISBN:978-1-4503-6679-3/19/05, (To be appeared).

[13] Md Islam, Md Uddin, Mumit Khan, et al. A light weight stemmer for bengali and its use in spelling checker. 2007.

[14] J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. 1975.

[15] Md Redowan Mahmud, Mahbuba Afrin, Md Abdur Razzaque, Ellis Miller, and Joel Iwashige. A rule based bengali stemmer. In *Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on*, pages 2750–2756. IEEE, 2014.

[16] Shervin Malmasi and Marcos Zampieri. Detecting hate speech in social media. *CoRR*, abs/1712.06427, 2017.

[17] Tarlach McGonagle. Minorities and online hate speech: A parsing of selected complexities. *Eur. YB Minority Issues*, 9:419, 2010.

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[19] Hamdy Mubarak, Kareem Darwish, and Walid Magdy. Abusive language detection on arabic social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 52–56, 2017.

[20] Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. *arXiv preprint arXiv:1701.08118*, 2017.

[21] Joni Salminen, Hind Almerekhi, Milica Milenkovic, Soon gyo Jung, Jisun An, Haewoon Kwak, and Bernard J Jansen. Anatomy of online hate: Developing a taxonomy and machine learning models for identifying and classifying hate in online news media. In *ICWSM*, pages 330–339, 2018.

[22] Manuela Sanguinetti, Fabio Poletto, Cristina Bosco, Viviana Patti, and Marco Stranisci. An italian twitter corpus of hate speech against immigrants. In *Proceedings of LREC*, 2018.

[23] Manjira Sinha, Sakshi Sharma, Tirthankar Dasgupta, and Anupam Basu. New readability measures for bangla and hindi texts. In *COLING*, 2012.

[24] T Zia, MS Akram, MS Nawaz, B Shahzad, AM Abdullatif, RU Mustafa, and MI Lali. Identification of hatred speeches on twitter. In *Proceedings of 52nd The IRES International Conference*, pages 27–32, 2016.