# Bangla Hate Speech Detection System using NLP and machine learning techniques

Omar Faruqe
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
omar.faruqe15@northsouth.edu

Md.Faisal
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
md.faisal3@northsouth.edu

Mubassir Jahan
Departmennt of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
mubassir.jahan@northsouth.edu

Md. Shahidul Islam
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
shahidul.islam3@northsouth.edu

Riasat Khan
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
riasat.khan@northsouth.edu

***Abstract*—Anyone can express themselves freely on social media. Still, many fail to obey community standards and breach the line of self-limitation, causing harm to others and occasionally leading to cyberbullying. Hate speech spreads hatred toward a person or a particular group based on various characteristics, e.g., race, religion, gender, and so on is referred to as bias. The offensive speech detection system is the frontier where researchers are battling to provide secure internet using Natural Language Processing and machine learning approaches. In this research, we strive to achieve this goal for those who speak the Bangla language. Our vision is to create a Bangla Hate Speech Detection System using natural language processing and machine learning approaches. In this work, we have made our custom dataset and used some labeled data from the previous open-access dataset. Both were merged together and implemented in this project. We have used around 4,978 data for prepossessing and running the code. The labeled datasets were categorized into four distinct classes, i.e., geopolitical, personal, religion, gender-abusive and non-hates. This work uses traditional classifiers, deep learning, transfer learning-based classifiers, or a mix of both types of classifiers to detect hate speech. Deep learning is a sort of machine learning that uses data to learn. It may be used to search for patterns in data. Transfer learning is a machine learning that allows a machine-learning algorithm to learn from data that another machine learning algorithm has already learned. Pretrained approaches have significantly advanced machine learning and natural language processing disciplines, including hate speech identification. These methods were used to identify hate speech. However, we also used Google API to convert text from Bangla to English. After that the emojis were removed from the data-sets and again that datas were converted back to Bangla. Finally, we have got the results where the best accuracy is in GRU and Attention model which is 98% . However, the lowest accuracy we've got in BERT model is 94% for this research.***

***Index Terms*—Keywords—word embedding, BERT Fine Tuning, Bi-LSTM, cross-validation, Transformer, GRU, Attention Model.**

## I. Introduction

Hate speech is illegal in every country. Different faiths, countries, genders, and civilizations can all be targets of hate speech. The main issue with hate speech is attracting people with immoral characters or morals. Furthermore,[18] it motivates them to spread hatred in society. Bangla is one of the world's most widely spoken languages. However, hate speech detection in Bangla is uncommon. The target of the project is to detect hate speech in Bangla. Everyone has the right to free expression. However, in the guise of free speech, this right is used to physically or verbally discriminate against others and assault general people. Hate speech that displays hatred towards a person or group based on characteristics such as race, religion, gender, and so on is referred to as prejudice. Hate speech is described by J. T. Nickleby [1] as "any communication that denigrates a person or group based on certain characteristics such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other particularities." Hundreds of incidents have happened due to hate speech and crimes, resulting in disputes, riots, and several murders. Despite the government's efforts to combat or minimize the problem, they were unable to find effective answers for regulating or resolving it through legislation or harsh penalties against agitators. To improve hate speech identification, various social media sites such as YouTube, Instagram, Facebook, and Twitter are increasing artificial intelligence and utilizing

machine learning approaches. Data creating models consisting of inputs and projected outputs, referred to as training data, are covered by supervised machine learning. These label data are used by algorithms such as Decision Trees, SVM, Logistic Regression, k-Nearest Neighbors, and others to classify based on some aspects of the labeled data. If there are any emojis or emoticons in the dasets then it can be removed easily by using the Google API. Deep learning is a branch of machine learning in which neural networks are used. The utilization of layers is employed. Each layer receives data, which it learns and transforms into the desired output.

People nowadays are easily harassed or targeted on social media. Being harassed by each other for sex, racism, politics, or anything else has become a big problem in society. Bad [19] problems like cyberbullying and online harassment are also on the rise. Many are being deceived by blackmailing using social sites. Much research has been done in English and other languages to identify hate text. However, very few works were done in the Bangla language. So, we have decided to build an automatic Bangla hate speech detection system using machine learning. We will do our work using the Doc2vector, BERT Fine-Tuning, LSTM, and Deep hate Explainer models. We will try to get the best Accuracy from these models. This paper implements machine learning and NLP techniques for detecting Bangla hate speech. The significant contribution of the works is as follows.

• We have made a custom dataset build by own and preprocessed it.

• An automotive Bangla hate speech detection system is developed Utilizing the Bangla Annotated dataset.

• These automated, deep learning-based navigation assistive systems are deployed in the Google Colab framework.

• Google Translator used in Bangla Emoji detection for data preprocessing. Fist,convert data Bangla to English .Then Clean data Emoji and again convert English to Bangla.Then use Transformer based model GRU ,Attention and BERT model.

• The performance of the proposed system is evaluated in terms of mean average precision, frame rate, accuracy, etc. We do a comparative analysis in an English-based model on the Bangla dataset in our project. Though our proposed English model works in the English Language but in, Bangla language prediction is not good if we compare it to English language prediction. We are the first to implement a different algorithm to detect Bangla hate speech in the Bangla language. So our work has a significant influence and vital role in the Bangla language community.

## II. RELATED WORK

Recently, extensive research has been done to investigate the automatic detection of hate speech, especially from social media and news websites. In the following paragraphs, some of the recent papers on Bangla hate speech recognition are discussed. For instance, in [2], the researchers created a dataset corpus that contained the Bangla comments from the social media site Facebook and annotated them for negative, positive, and neutral categories to detect hate speech. First, the authors explored, analyzed, and preprocessed the data, then performed the exploratory data analysis based on the survey. Next, they used the traditional machine learning algorithm with SVM models, BNLP tools for data cleaning, and Term Frequency Inverse Document Frequency vectorizer for feature extraction to convert the data into a matrix of features. Finally, they used a confusion matrix to analyze the correct and misclassification numbers. Also, they used a vectorizer to measure the Accuracy using uni-gram, bigram, and trigram classifiers. Finally, they concluded that the highest Accuracy is obtained with the MNB classifier with unigram features, where 62.85% of cases were predicted correctly compared to the other classification methods.

In [3], the authors have created a data sheet on hate speech. They use their Dataset, first converting from video to audio and audio to text format, then applying natural language processing techniques to various models. Deep learning hyper-parameter adjustment was also applied to LSTM and GRU deep learning techniques. They evaluated and contrasted the findings of several models and analyzed them using various assessment criteria. Finally, this work obtained 98.89% and 86.67% accuracy for the GRU and LSTM models.

In [4] this paper, the researcher works in automated abusive comment detection of Bangla alphabets, Bangla-English code mixed text, and translated Bangla text dataset based on public Facebook page comments and celebrity pages. The authors used an online scraper tool to avoid generating a biased dataset and made 2,000 labeled comments. For Dataset preprocessing, they used a profanity-based detection algorithm which helps to detect offensive words by replacing some characters of the words with special characters or digits. Next, this work employed an automatic model based on three classification machine learning algorithms, i.e., support vector machine, random forest, and AdaBoost. Random forest gives the best score in Accuracy with 72.1%, compared to other models.

In [5] this paper, the authors propose an explainable approach for different categories of Bangla hate speech detection. To achieve better performance, they trained their Dataset with Conv-LSTM, Bangla BERT, and XML-Roberta techniques. Because of their middle range dataset, some ML classifier does not give better performance on ML and DNN, but BERT Variants give a better F1 score of 88%. In this paper, the authors explain the Accuracy and show comparative analysis with baselines. They also observe the high classification accuracy and explainability. Therefore, they want to overcome the limitation by improving the significant area.

In [12], about 7,425 Bengali comments are collected in a dataset. The comments were gathered from various Facebook pages using the Facebook Graph API. We also manually collected the comments because the Graph API has limitations for us. The seven categories of hate speech, aggressive comment, religious hatred, ethnic attack, religious comment, political comment, and suicidal comment, were then applied to these remarks. A Bangla Emot Module was developed to assist in identifying the emotions concealed by emoticons and emojis. This makes it easier to understand what constitutes hate speech

in Bangla. Later, the pertinent specifics are covered. They used supervised machine learning techniques, including attention LSTM and Gated Recurrent Unit (GRU) based decoders. The algorithms produced an accuracy of 74%. The Attention mechanism subsequently enhanced the model, which had a 77% accuracy.

In this research[13], we have provided a framework for speech acquisition combined with the speaker's position, translating the talks into texts, and finally, we have suggested a system based on long short-term memory (LSTM), a variation of Recurrent neural networks (RNN) are used to categorize talks as suspicious or not. We considered Bangla speeches. Five thousand suspicious and non-suspicious samples make up our dataset, which we produced for training and to confirm our model. In comparing the precision of various machine learning techniques, including logistic regression, The effectiveness of the system is assessed using SVM, KNN, Naive Bayes, and decision trees. Experimental Results indicate that when compared to other models, our suggested deep learning-based model offers the highest accuracy.

In this paper[14], most research on abusive text or comment detection is done in English, and part of it aims to find demeaning or degrading texts. However, there are a few works in Bangla as well. Finding offensive material in Bangla can help to stop cybercrimes such as online blackmailing, harassment, and bullying, which are currently Bangladesh's top concerns. their objective is to find offensive Bangla comments that have been collected from various social media platforms where users express their thoughts, feelings, ideas, etc. They discuss performing abusive text analysis using Bangla comment data and manually gathering the training and test data. By applying our suggested categorizing algorithm, we attempted to achieve that. We get a satisfying maximum accuracy of 71.7% for 300 comments and an average of 68.9% for the entire test data set, even though this root-level approach is outdated today. We anticipate that our future strategy will produce a superior outcome to previously produced concepts. There are still lots of ways to make our experimental methodology better. Natural language processing employs a variety of machine learning algorithms and approaches. As the lack of mentorship, work in the Bangla language has not increased as anticipated.

From the above paragraphs, we can claim that much-related work has been implemented based on the natural language process and deep learning method. The algorithm gives significant thought to developing the idea of the new model and implementing Bangla hate speech detection. These paragraphs also elaborate significance of these models and which model would be the best and most suitable for our related work.

The following is the sequence in which our report was presented: Title and author; Abstract; Introduction; Experimental Procedures(Methodology); Results; Discussion; Acknowledgements; references, tables, figures and Future work one per page, each with a legend.

## III. METHODOLOGY

The suggested system's dataset, preparation approaches, and detection algorithms are all described in this section. We used Google collab to run and test the code. Pytorch was used to bug fix and further test the programming codes. And necessary Python dependencies were downloaded to perform specific operations in this project. The working sequences of the proposed Bangla hate speech detection system are represented in Fig. 1. This paper uses a fine-tuned BERT model, Bi-LSTM, and several machine learning approaches to detect hate speech in Bangla.

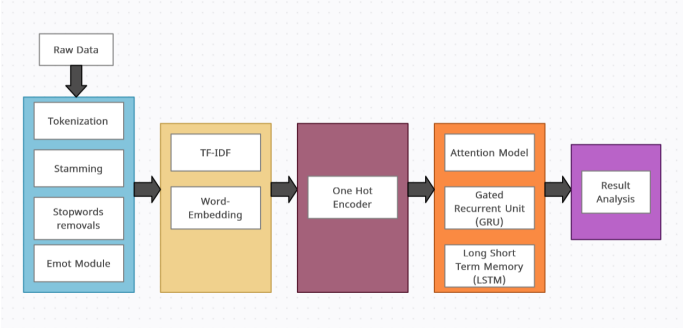



Fig. 1. Working flowchart of the proposed Bangla hate speech detection system.

### A. Software Tools/Algorithms

Google Colab, Pytorch, Python Packages.

### B. Models

• BERT Fine Tuning: BERT is first developed by Google for research purposes. It is a transformer-based model. It is mainly used for Linguistic prediction and following sentence prediction purposes [6]. In our model, first, we run our reference model code. In our reference code, they used the Cross-Validation method, and then we imported our Dataset on this model. Finally got the Accuracy. BERT is a large neural network design with many parameters ranging from 100 million to 300 million. Overfitting would emerge from training a BERT model from scratch on a tiny dataset, which is why it's better to start with a pre-trained BERT model that was trained on a large dataset. We may then train the model again on our reduced dataset, a technique known as model fine-tuning

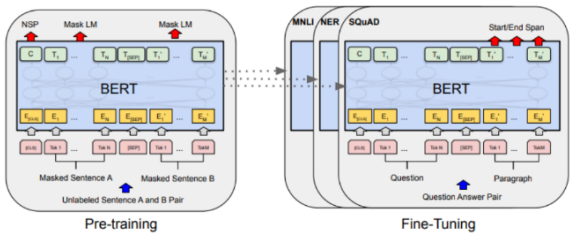



Fig. 2. BERT Model Tuning.

Fig. 2 depicts the BERT framework's overall pre-training and fine-tuning methods. Whereas, Table I demonstrates the confusion matrix of BERT model. The same designs are used in pre-training and fine-tuning, in addition to the output layers. Models for multiple downstream processes use the same pre-trained model parameters. During fine-tuning, all parameters are fine-tuned. [CLS] is a unique separator token (for example, separating questions and answers), and [SEP] is a special symbol that appears before each input example. The intuition behind BERT's fine-tuning is that many researchers who try to get better Accuracy used BERT's fine-tuning for an experiment purpose. In our project, we also try to get experiments in BERT fine-tuning. It is easy for us to fit our Dataset into a reference project. Also, I previously tried to experiment on the "Bangla POS tagging" project using BERT fine-tuning for experiment purposes[7]. We got better Accuracy.

Fig.3 shows In our code we choose reference code which are IMDB movie review detection.On that refernce code we select our Bangla based hate speech detection.And we got good accuracy On Bert Fine Tuning 94%. The pre-training/fine-tuning paradigm was introduced by Google's BERT, which marked a paradigm shift in natural language modeling. After pre-training the model unsupervised on a large amount of text data, the model can be quickly fine-tuned on a particular downstream task with few labels because the general linguistic patterns have already been learned during pre-training. In Fig 22, Authors According to Jacob Devlin et al., fine-tuning BERT is "straightforward," requiring only the addition of one more layer after the final BERT layer and brief training of the entire network. After just 2-3 epochs of fine-tuning with the ADAM, the authors show strong performance on the common NLP benchmark problems GLUE, SQuAD, and SWAG, which look at various facets of natural language inference.in Fig 23, A recipe that has gained widespread acceptance in the research community is an optimizer with learning rates ranging from 1e-5 to 5e-5. This pre-training/fine-tuning paradigm is now accepted in the industry due to its astounding success. But scientifically speaking, we don't really comprehend the fine-tuning procedure all that well. What layers alter when you fine-tune? Do we need to make any adjustments? How reliable are the outcomes, too? Let's explore some of the more recent "BERTology" studies that were conducted after the initial BERT study.

• Bi-LSTM: The Bi-LSTM is a recurrent neural network that processes natural language [9]. The input travels in both directions, and it can use information from both sides, unlike ordinary LSTM. Table II refers the confusion matrix. It is also helpful for simulating the sequential relationships between words and phrases in both directions. In short, the bi-LSTM model adds an LSTM layer, which reverses the information flow. In a word, it implies that the input sequence is changed in the extra LSTM layer. The outputs of both LSTM layers are then merged using different methods, e.g., sum, average, multiplication, and concatenation.

• HateXplain: HateXplain is a more explainable part of hate speech detection . Because one sentence is either hate speech or Abusive speech. Which is not precisely wise predicted. In using BERT's pre-trained model, abusive and hate speech detection. Which is also known as the HateXplain model. HateXplain model also uses the best base uncase train model[10].

• Transformer: A transformer is a deep learning model that uses the self-attention process and weights each component of the incoming data differently based on its significance. The main applications are computer vision (CV) and natural language processing (NLP).[11]

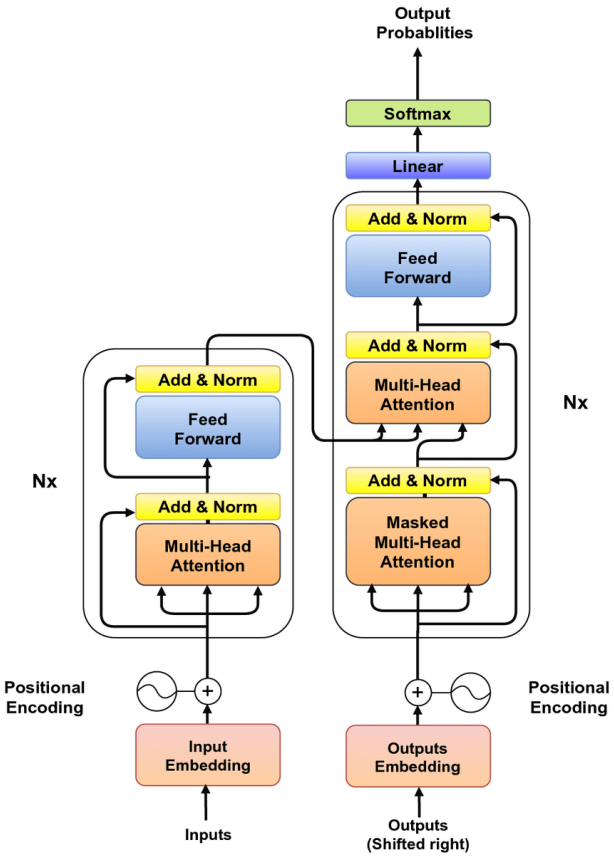



Fig. 3. Transformer model architecture

Data are preprocessed using some preprocessing methods, and the related information was extracted from emoticons and emojis to detect the type of speech. Bangla natural language tokenization was used to split sentences into words. Features were extracted using TF-IDF vectorization, and word embedding was also used. Furthermore finally, classification approaches such as CNN, Bidirectional LSTM, and GRU were applied to compare their performance. Here we have Applied a Recurrent neural network (RNN) with an Attention Mechanism for text classification. Finally, the performances of all the classification approaches have been analyzed and compared.

• GRU: A gated recurrent unit (GRU) is part of a specific recurrent neural network model that intends to use connections through a sequence of nodes to perform machine learning tasks associated with memory and clustering.There is a similarity between GRU and the lstm model. In lstm, there is only an

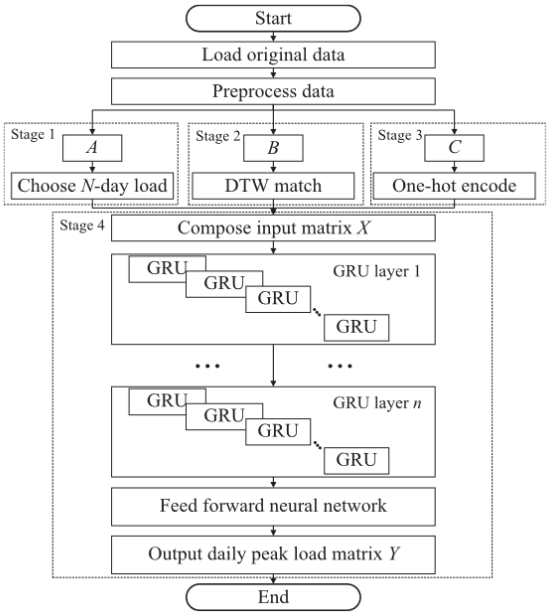



Fig. 4. Architecture of tokenization.

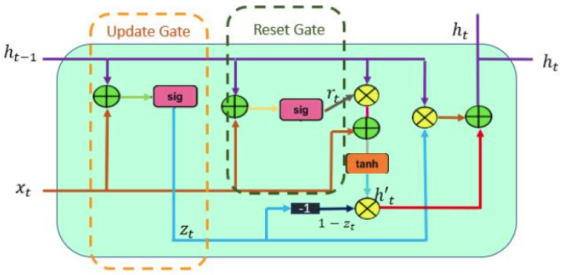



Fig. 5. Architecture of GRU model.

input gate and output gate, but in the GRU model, it has three gates input, output, and update gate. The GRU model uses the update gates to control the flow of information, and because of this feature, the GRU model is more significant than the last model.

Encoder hidden state is defined as:

$$h_t = f(W^{h^h})h_{t-1} + W^{h^x}x_t \tag{1}$$

Decode hidden state can be expressed as:

$$h_t = f(W^{h^h})h_{t-1} \tag{2}$$

Finally, the ouptut state is given by:

$$Y_t = softmax(W^S \times h_t) \tag{3}$$

Fig.7 represents the architecture of GRU model which trains faster and perform better than LSTM. In addition to being more straightforward to modify, GRU is also faster and uses less memory than LSTM when adding new gates. However,

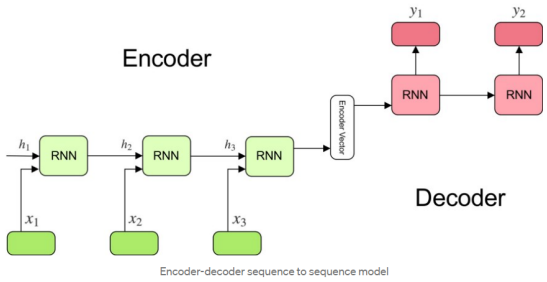



Fig. 6. Encoder and Decoder Architecture.

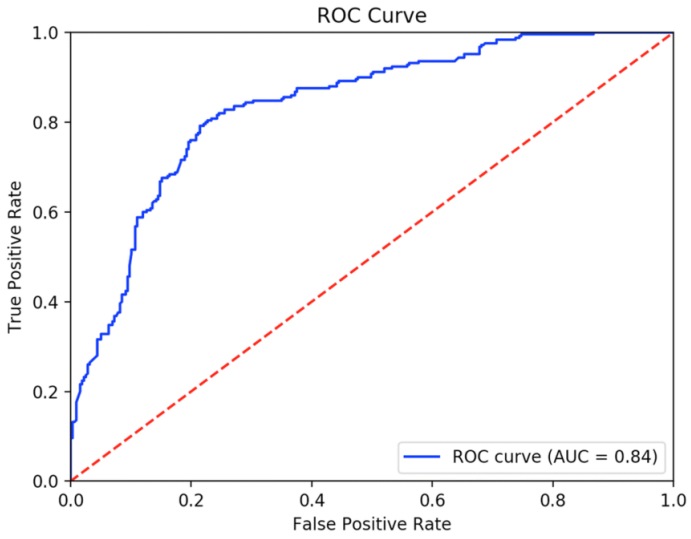



Fig. 7. Architecture of GRU model.

LSTM is more accurate when using datasets with longer sequences, all input to the network. It is just less code in general. An indicator of performance for classification issues at different threshold levels is the AUC-ROC curve.AUC stands for the level or measurement of separability, and ROC is a probability curve.It reveals how well the model can differentiate across classes. The model performs better at predicting 0s as 0s and 1s as 1s the higher the AUC. By analogy, the model is more effective at differentiating between patients with the condition and those who do not have it the higher the AUC. TPR is plotted against FPR on the ROC curve, with FPR on the x-axis and TPR on the y-axis.

• Attention based model: Attention-based model is a sequence-to-sequence model whose aim is to produce an output sequence according to the given input sequence in different lengths. The primary mechanism of this model is to improve the performance of the encoder to decoder in machine translation. The idea of the attention-based model is that it permits the decoder to utilize the most relevant part of the input sequence in a specific manner.In every output sequence, the decoder generates a word at a time, then takes the word in the previous step(t-1) as input for generating the next word in output.This method is more reliable in short-term sequences

but has complexity in long-term sequences. The architecture of the proposed attention model for Bangla hate speech detection has been demonstrated in Fig. 8.

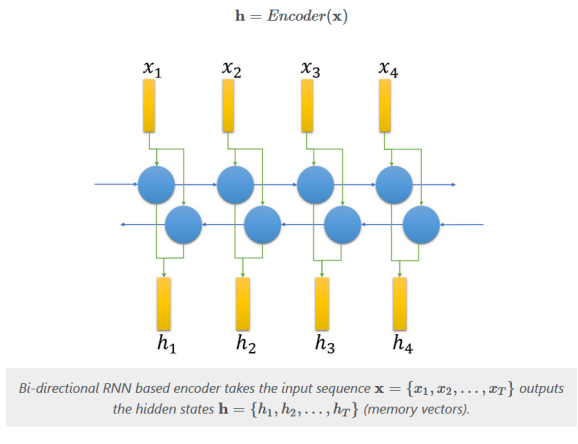



Fig. 8. Attention model architecture.

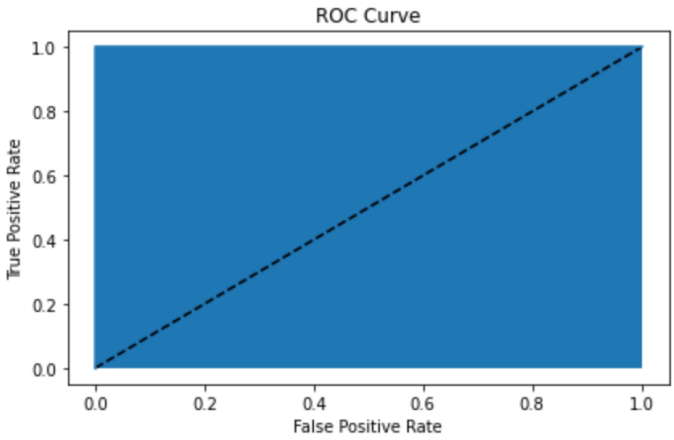


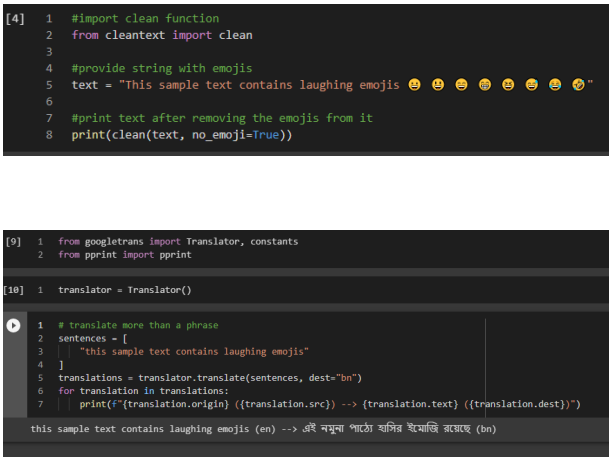

Fig. 9. ROC Curve of Attention Model.

Let input sequence be $x = x_1, x_2, ..., x_T$ and output sequence be $y = y_1, y_2, ..., y_U$. There are two parts of sequence-to-sequence models, viz. encoder and decoder, both of which are RNNs. Encoder encodes the given input producing a series of hidden states $h = h_1, h_2, ..., h_T$ of input $x$.

We have also worked on a general model implementation like Logistic Regression and XGBoost giving good results based on different attributes. Fig 9 represents ROC curve of Attention model. However, Fig 10 represents the techniques of detecting emojies.

### C. Dataset Preprocessing

The working process involves collecting hate speech from social media sites and performing exploratory data analysis, data cleaning, modeling, and evaluating the proposed system. In this work, we used about 4,000 classified instances to the Bengali Hate Speech Dataset from [8]. Previously, the dataset classified political, personal, geopolitical, religious, and gender abusive hate observations. However, based on our research and analysis, we discovered that distinguishing between private and gender abusive hate is sometimes tricky since they frequently overlap linguistically. Consider the following hateful comments as examples. For data collecting, they used a bootstrap strategy. Only writings containing widely used phrases, whether aimed at a single person or entity or generalized to a group, are evaluated. Texts were gathered from Facebook comments, YouTube comments, and newspaper articles.



```
[4]
#import clean function
from cleantext import clean

#provide string with emojis
text = "This sample text contains laughing emojis 😀 😃 😄 😁 😆 😅 😂 🤣"

#print text after removing the emojis from it
print(clean(text, no_emoji=True))
```

```
[9]
from googletrans import Translator, constants
from pprint import pprint

[10]
translator = Translator()

# translate more than a phrase
sentences = [
    "this sample text contains laughing emojis"
]
translations = translator.translate(sentences, dest="bn")
for translation in translations:
    print(f"{translation.origin} ({translation.src}) --> {translation.text} ({translation.dest})")

this sample text contains laughing emojis (en) --> এই নমুনা পাঠ্য হাসির ইমোজি রয়েছে (bn)
```

Fig. 10. Emoji and emoticons detection.

Since we have collected our data from different sources, we have many emojis and emoticons. To detect emoticons and emojis, we have used some techniques to detect and clean those tags. Mainly, we have implemented the translator method from the Googletrans package.

Furthermore, the samples were divided into political, personal, geographical, religious and none hatred. Fig. 11 demonstrates a sample distribution and definition of several types of hate. Shows several examples of dataset labeling for distinct hatred classifications.

| text | label |
|---|---|
| ইনিই হচ্ছেন ভারতের প্রতিরক্ষামন্ত্রী মনোহর পারিকর এই কদিন আগেই যিনি ঘোষনা দেন বাংলাদেশ দখল করে মহাভারত গঠন করবেন আশ্চর্য তিনিই কিনা আমাদের বরেন্য রাষ্ট্রীয় অতিথি | Geopolitical |
| রেন্ডিয়াকে পৃথীবির মানচিএ থেকে মুচে ফেলতে হবে | Geopolitical |
| এই মালাউনরা বাংগালিদের সাফল্য দেখে হিংসা করে বাহির দেশে বাংলাদেশীদের দেখতে পারেনা এই একটি মাএ জাতি সুতরাং সাবধান | Religious |
| আমরা বলতে কারা ভারত তাইতো | Geopolitical |
| পাকিস্তান আমার বাল | Geopolitical |

Fig. 11. Sample of labeled datasets.

**Algorithm: Algorithm of the One-Hot Encoder**

**Start:**

**Initialize:** for Length BanglaHateSpeechDetection Non-Hate ; **do**

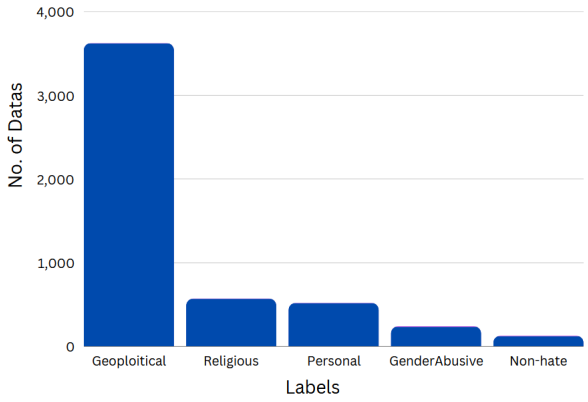



Fig. 12. Histrogram of Datasets

**Compute:** $Label'$ using ('BanglaHateSpeechDetection Non-Hate ' n= BanglaHateSpeechDetection - Non-Hate ';)

**Apply** Length BanglaHateSpeechDetection Hate'n ;

**Enable** Label' BanglaHateSpeechDetection Hate n= BanglaHateSpeechDetection =Hate';

**if not**(missing('BanglaHateSpeechDetection'))

then

**do** (BanglaHateSpeechDetection Non-Hate =0 && 'BanglaHateSpeechDetection Hate=0)' ;

**Employ** if kleft ('BanglaHateSpeechDetection') Hate' then BanglaHateSpeechDetection Non-hate=1;

**end;**

**Start:**

**Initialize:** Length 'Edlevel Hate'n ;

**Compute:** $Label'$ 'Edlevel-Hate' n= Edlevel- Hate ';

**Apply** Length Edlevel-Non-Hate'n :

**Compute:** Ed-$'Label'$ Non-Hate'n = 'Edlevel-Non-Hate';

**if not** (missing('Edlevel'n))
then
**do** ('Edlevel-Hate'n=0 && 'Edlevel Non-Hate'n=0 );

**if** kleft('Edlevel'n) - Hate'
then
('Edlevel-Hate'n=1);
else

**if** (kleft('Edlevel'n)= 'Non-Hate'
then
('Edlevel Non-Hate'n=1);

**end;**

• In our projects, we faced another problem with Software tools. In free Google Colab, we have limited free access. However, the most concerning part are that if our training dataset size is large, we have to find out paid access in Google Colab. We do comparative analysis on some English Models and infer them in Bangla Datasets. Few models give us better Accuracy in our chosen techniques. However, still, we have to find other models that give us better prediction and Accuracy.

• Sometimes we have to install some python libraries individually to support import packages.

## IV. RESULTS AND DISCUSSION

In BERT's fine-tuning, we got an accuracy of 94 percent. Compared to another model's Accuracy, BERT's fine-tuning gives better results. Fig. 13 demonstrate the performance metrics of the BERT Fine-tuned model for train and test

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.95 | 0.94 | 0.95 | 2534 |
| 1 | 0.94 | 0.95 | 0.94 | 2466 |
| accuracy | | | 0.95 | 5000 |
| macro avg | 0.95 | 0.95 | 0.95 | 5000 |
| weighted avg | 0.95 | 0.95 | 0.95 | 5000 |

Fig. 13. BERT Fine tuning on bangla hate speech.

Nevertheless, in BERT's fine-tuning, we have no required training parts. Take more time to fit the reference code training part because we have limited free GPU access in google co-lab. Another problem is Fine Tuning. Most of the layers in the BERT model freeze and a few layers are activated. We could not get better predictions in this Fine-tuning process.Fig.14 We have shown the BERT Accuracy Level Graph.

In Logistic Regression, baseline accuracy is 60%, and the SVM baseline is 58%.

In Bi-LSTM model summary has 2,630,154 parameters. With the dictionary-based approach, we got an accuracy of 71%, which is shown in Fig. 15. Nevertheless, we face a problem when we give the input a sentence included in the Dictionary; we get a better prediction. However, giving random sentences in the input model can not recognize. In HateXplain Model, we used the best-pretrained model. After the Normalization, we got classification results. Furthermore, Accuracy is 87%.

Nevertheless, in the pretrain model, we do not need to train or test data split. Without training and test data, we can not get accurate predicted results. This model is the same. Fig. 16

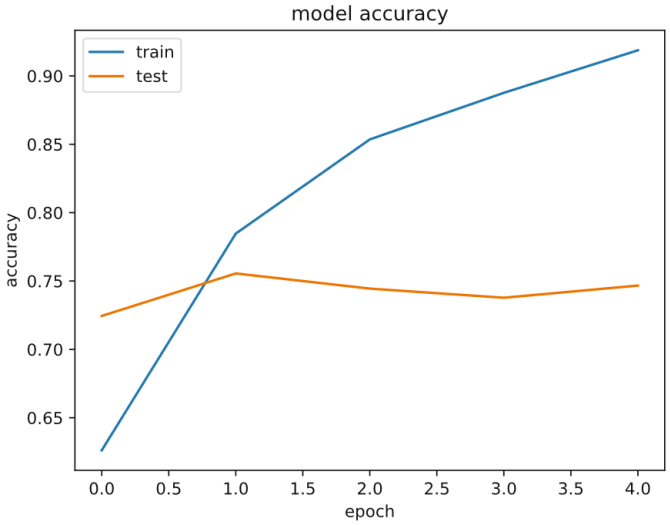



Fig. 14. BERT Accuracy Level Graph (Train and Test)

demonstrate that the stemmed and lemmazed version of the change of the epochs.

```
u = 'রোহিঙ্গাকে পৃথীবির মানচিত্র থেকে মুছে ফেলতে হবে'
inputs = tokenizer(u, return_tensors="pt")
labels = torch.tensor([1]).unsqueeze(0)  # Batch size 1
outputs = model(**inputs, labels=labels)
label = categories[np.argmax(outputs.logits.softmax(dim=-1).tolist())]
```

```
outputs.logits.softmax(dim=-1).tolist()
```

```
[[0.04814157634973526, 0.7389839887619019, 0.21287444233894348]]
```

```
label
```

```
'normal'
```

Fig. 15. HateExplain Model.

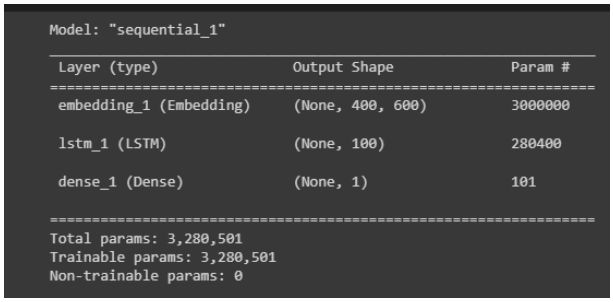



Fig. 16. Stemmed and Lemmatized

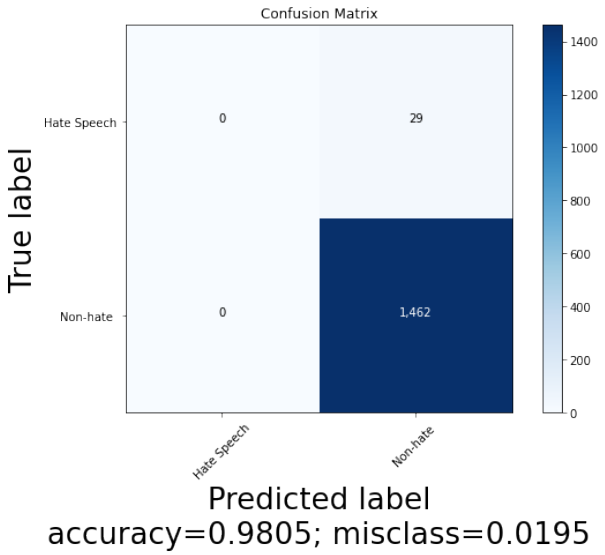



Fig. 17. GRU Model Confusion Matrix (Without Normalization)

In this code, we input a Bangla sentence which labeled with hate word. In some cases when a sentence refers Normal sentence with no hate speech. Consequently, this prediction is incorrect. However, if we convert this sentence into English, it gives an actual forecast. So, this English pre-trained model can not provide accurate predictions on Bangla sentences. Finally, Fig 15 demonstrates the evaluation metrics of the proposed HateExplain model.

We have work stemmed and lemmatization for implementing the method. The lemmatization technique requires putting together a word's inflected components so they can be recognized as a single entity. Similar to stemming, but with meaning attached to the root words. Stemming is a method that uses rules to generate variations of a root or base word. Simply put, it boils down to an essential word to its stem. Given that this heuristic procedure entails randomly clipping the ends of the words, it is the easier of the two. In order to make sentences easier to grasp, stemming shortens the look-up process.

In the GRU model, we got the best result from all of the results applied. In our previous work in BERT, we got the 94% accuracy, but in the GRU model, we have the highest accuracy, 98.87% and better results than another classification report. We have faced many problems while implementing the GRU model. Among them, we are in trouble training the dataset. The GRU model could read our train set from the dataset and fit with the model, but after several approaches, we have implemented it successfully.

Fig. 16 We have shown the stemmed and lemmatization for our used model. Fig. 17 We have shown the GRU Model Confusion Matrix with out using Normalization. Fig. 18 The GRU MODel Confusion Matrix with using Normalization. Fig. 20 We have shown the BERT Model Confusion Matrix Plot where Normalizer is true. Fig. 19 We have shown the BERT Model Confusion Matrix Plot where Normalizer is false. Fig.21 We have shown the value loss graph for values 1. Fig.22 We have shown the value loss graph for values 0.

Finally, Table I, shows the testing accuracy of several machine learning and natural language processing hate speech identification approaches(precision, Recall and F1-score) .The fine-tuned BERT and GRU model got the highest Accuracy for the aforementioned classification job, according to the analysis and findings.

Here we have shown comparatively our applied different models and their related classification report in Table II. However, in table III refers Performance comparison of the

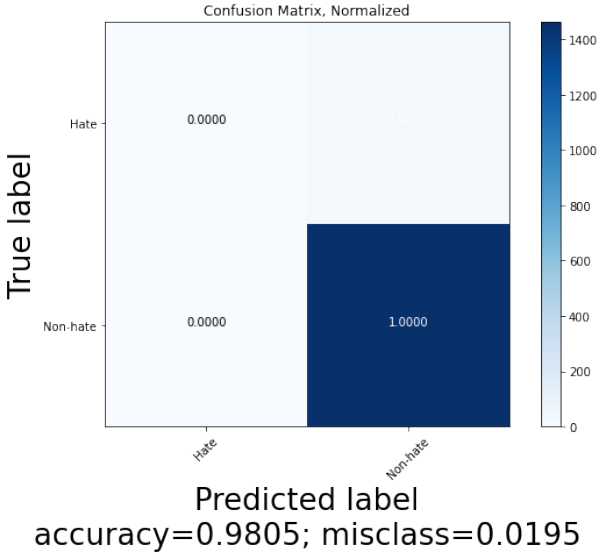



Fig. 18. GRU Model Confusion Matrix (With Normalization)

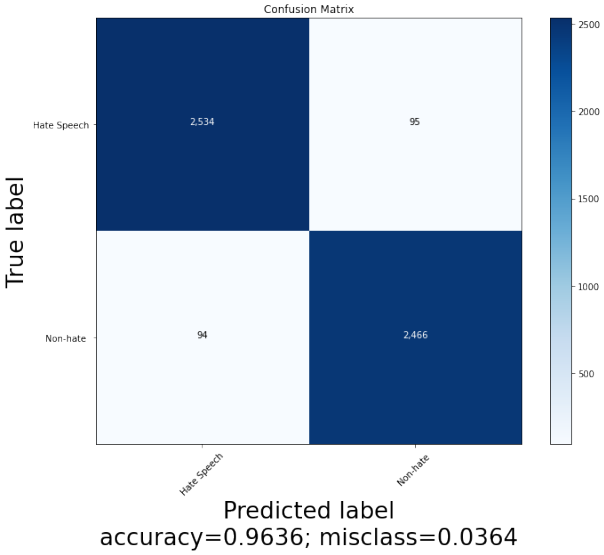



Fig. 19. BERT Model Confusion Matrix (without Normalization)

| Model | | Precision | Recall | F1-Score | Accuracy |
|---|---|---|---|---|---|
| GRU | 0 | 0.00 | 0.00 | 0.00 | 98% |
| | 1 | 0.98 | 1.00 | 0.99 | |
| Attention | 0 | 0.00 | 0.00 | 0.00 | 98% |
| | 1 | 0.98 | 1.00 | 0.99 | |
| BERT | 0 | 0.95 | 0.94 | 0.95 | 95% |
| | 1 | 0.94 | 0.95 | 0.94 | |

TABLE I
USED MODELS AND THEIR PRECISION, RECALL AND F1-SCORE

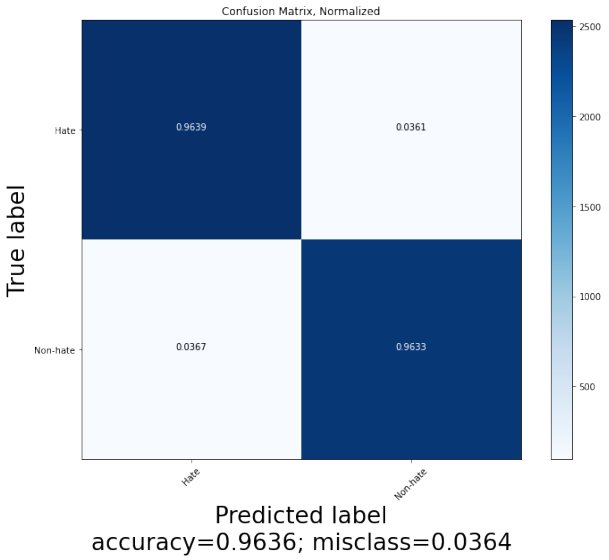



Fig. 20. BERT Model Confusion Matrix (with Normalization)

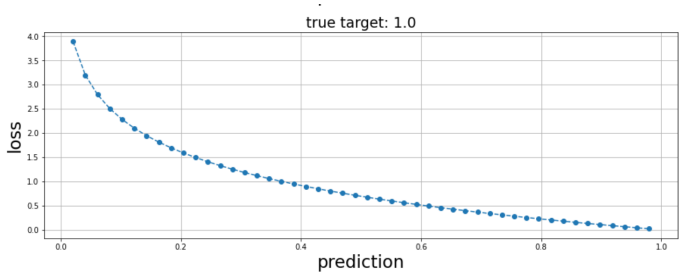



Fig. 21. GRU MODEL VALUE LOSS graph for Values 1

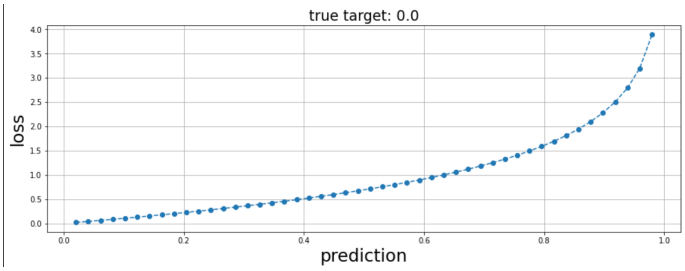



Fig. 22. GRU MODEL VALUE LOSS graph for Values 0

| References | Model | Validation |
|---|---|---|
| Proposed System | BERT | 94.04% |
| | Bi-LSTM | 71.68% |
| | HateXplainer | 87% |
| | GRU | 98.87% |
| | Attention Model | 98% |
| Number of Classes | Geopolitical | 3610 |
| | Religious | 557 |
| | Personal Hate | 502 |
| | Gender Abusive | 216 |
| | Non-Hate | 98 |

TABLE II
COMPARISON TABLE OF RELATED PROPOSED SYSTEM AND NUMBER OF CLASSES

| Name of the approch | BERT | Random Forest | LSTM | GRU | Attention | SVM |
|---|---|---|---|---|---|---|
| Paper[27] | - | 78% | - | - | - | 82% |
| Paper[28] | - | - | 77% | - | - | 78% |
| This paper | 94% | - | - | 98% | 98% | - |

TABLE III
PERFORMANCE COMPARISON OF THE PROPOSED APPROACH

proposed approach.

## V. CONCLUSION AND FUTURE WORK

In Bangladesh, the number of social network users increases with internet users. That is why it is necessary to detect Bangla hate speech. We explored several methods for detecting Bangla hate speech, and we trained them with the Data set for the experiment. In the Data-set, there were anomalies, and we processed them to remove their anomalies. After all, we got different results from the proposed method, and we analyzed them according to the evaluation. Although some models gave us good Accuracy, some could not provide good results according to our expectations.All of the models we have got best accuracy from attention based GRU model (accuracy 98.87%) and Bert model (accuracy 94%). Hence we have some limitations and faced obstacles during our work. we have noticed that have got same results in transformer method becuase of one hot encoder.We will work in the future for better Accuracy by implemented different method.

In the future, we will try increase the amount of data-set and level the Data set according to the category. We will try to enrich our Data-set by collecting different sources, especially the video content, and preprocessing them for better results. We will also work on the real time platform that we can show the results and visualize the implementation. We are deterministic that we can overcome these limitations and will give good results hopefully.

## REFERENCES


[1] J. T. Nickleby, "Hate speech Encyclopedia of the American constitution," New York: Macmillan Reference USA, Second edition, 2000, pp. 1277-1279.

[2] S. A. Kaiser, S. Mandal, A. K. Abid, E. Hossain, F. B. Ali and I. T. Naheen, "Social Media Opinion Mining Based on Bangla Public Post of Facebook," International Conference on Computer and Information Technology, 2021, pp. 1-6.

[3] M. I. Hossain Junaid, F. Hossain, and R. M. Rahman, "Bangla Hate Speech Detection in Videos Using Machine Learning," Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, 2021, pp. 0347-0351.

[4] M. R. Karim *et al.*, "DeepHateExplainer: Explainable Hate Speech Detection in Under-resourced Bengali Language," International Conference on Data Science and Advanced Analytics, 2021, pp. 1-10

[5] M. Jahan, I. Ahamed, M. R. Bishwas and S. Shatabda, "Abusive Comments Detection in Bangla-English Code-mixed and Transliterated Text," International Conference on Innovation in Engineering and Technology, 2019, pp. 1-6.

[6] X. Han and L. Wang, "A Novel Document-Level Relation Extraction Method Based on BERT and Entity Information," IEEE Access, vol. 8, pp. 96912-96919, 2020..

[7] . R. Cai *et al.*, "Sentiment Analysis About Investors and Consumers in Energy Market Based on BERT-BiLSTM," IEEE Access, vol. 8, pp. 171408-171415, 2020

[8] M. R. Karim, B. R. Chakravarti, J. P. McCrae and M. Cochez, "Classification Benchmarks for Under-resourced Bengali Language based on Multichannel Convolutional-LSTM Network," International Conference on Data Science and Advanced Analytics, 2020.

[9] A. A. Sharfuddin, M. N. Tihami and M. Saiful Islam, "A Deep Recurrent Neural Network with BiLSTM model for Sentiment Classification," International Conference on Bangla Speech and Language Processing, 2018, pp. 1-4.

[10] B. Mathew *et al.*, "HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection," AAAI Conference on Artificial Intelligence, pp. 14867-14875, 2021.

[11] https://en.wikipedia.org/wiki/Transformer machine learning model

[12] A. K. Das, A. Al Asif, A. Paul, and Md. N. Hossain, "Bangla hate speech detection on social media using attention-based recurrent neural network," Journal of Intelligent Systems, vol. 30, no. 1, pp. 578–591, Jan. 2021

[13] Md. R. Rahman, M. S. Arefin *et al.*,"Towards a Framework for Acquisition and Analysis of Speeches to Identify Suspicious Contents through Machine Learning," Complexity, vol. 2020, pp. 1–14, Nov. 2020.

[14] M. G. Hussain and T. A. Mahmud, "A technique for perceiving abusive Bangla comments," GUB Journal of Science and Engineering (GUBJSE)

[15] 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," Journal of Intelligent Systems, vol. 30, no. 1, pp. 578–591, 2021.

[16] B. Mathew, P. Saha, S. Yimam, C. Biemann, P. Goyal and A. Mukherjee, "HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection", Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 17, pp. 14867-14875, 2021. Available: 10.1609/aaai.v35i17.17745 [Accessed 29 September 2022].

[17] H. Saleh, A. Alhothali, and K. Moria, "Detection of hate speech using Bert and hate speech word embedding with deep model," arXiv.org, 02-Nov-2021. [Online]. Available: https://arxiv.org/abs/2111.01515. [Accessed: 29-Sep-2022].

[18] N. I. Remon, N. H. Tuli and R. D. Akash, "Bengali Hate Speech Detection in Public Facebook Pages," 2022 International Conference on Innovations in Science, Engineering and Technology (ICISET), 2022, pp. 169-173

[19] A. M. Ishmam and S. Sharmin, "Hateful Speech Detection in Public Facebook Pages for the Bengali Language," 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 2019, pp. 555-560