Detection Of Hate Speech And Offensive Language Using Machine Learning And Neural Networks WithAI Explanation

Dr. Shanoli Pal, Anik Chakraborty, Raktim Chakraborty, Dr. Indranil Mitra

[1]Senior Associate,[2]Manager,[3]Director, [4]Managing Director,
[1]Advanced Analytics, Data & Analytics,
[1]PwC, India

Abstract - Natural language processing (NLP) has been successfully implemented in many text-mining applications. One of the most useful utilities of text mining analytics is hate speech detection and explanation of classified hate speech or offensive languages. Growing users of social media and their freedom of speech in all aspects of life make social media popular. But the used language, sentence formation, and context of the content have no pre-defined form, which complicates the detection ofany abusive, offensive language. This is also challenging due to multilingual texts, and paralinguistic signals (like emoticons, all other media files, and hashtags). These are some points fromthe implementation point of view. Hate speech can spread likea plague if it's not handled carefully. But it is evident that if it's not being handled mindfully, it can cause massive non- recoverable damage to society. So automated detection of hate speech is very useful to take necessary precautions against the toxic spread. Media platforms are full of several types of hate speeches. Precautions should be taken against hate speeches, offensive language, toxic words, etc. Machine learning (ML) and deep neural networks can classify toxic statements. But manual annotation based on the text meaning is another time-consuming process. Machine learning algorithms are useful to reduce humanefforts by learning and deciding from the annotated data. Sometimes, it happens that the decisions or outcomes of machine learning algorithms are not explainable or cannot be matched with human understanding. So, the study of such things is getting popularity due to massive data handling and managing mental health, and social conditions in a better way. Here, inthis work, advanced neural networks have been used to get the context from the messy contents and to detect hate speeches and offensive language. The used dataset is manually annotated in three labels hate, offensive, and neither. Different embeddings andclassification algorithms like Term Frequency - Inverse DocumentFrequency (TFIDF), Bidirectional Encoder Representations fromTransformers (BERT), Multinomial naive bayes, Long short term memory (LSTM), Generative Pre-trained Transformer 2 (GPT2), Transformer XLNet have been studied and detailed results as well as algorithm performances are explained using Local Interpretable Model-Agnostic Explanations (LIME) in thiswork. In the best model, 95% test accuracy has been achieved using XLNet.

Index Terms - NLP, TFIDF, BERT, Multinomial naive bayes, LSTM, GPT2, XLNet, LIME.

I. INTRODUCTION

Hate speech detection is a subfield of text mining, stilldeveloping in natural language processing. Social media like Twitter, Facebook, Instagram, and Snapchat are flooded with personalized human comments, judgments, opinions, etc. People are using these platforms to share every moment, every experience openly. These platforms provide the opportunity tospeak on everything, convey own thoughts, and get support. Itseems that these are good for mental health. But the darkest part of this is that these platforms have bad impacts on society if used with bad intentions. In human society, thereare several social restrictions followed regularly in our lifeor it can be said society is bounded by laws. But peopledo not follow legal bounding when expressing their views against anything, passing abusive, unrealistic language havinggreat impacts on others' image, emotion, or mental health. Detecting such human expressions expressed by languages andrestricting those from over-flooding in a mass population are huge responsibilities that should be considered by Governmentand these social media platforms.

Autodetect and tagging those abusive words by using ma- chine learning capability is a developing area of research inthe field of natural language processing. Machine learningalgorithms highly depend on the quality of the dataset, specifically when it is complicated to understand the context and content altogether. Also, human comments are mixed with non-standard variations in spelling and grammar, sentence formation. If the expressions are multilingual, then hate content becomes code-mixed form which makes hate speech detectionmore complicated.

Here, in this work, some advanced techniques for wordembeddings and model building to classify hate and offensive speeches have been used. Then model classification outcomes have been explained by using LIME. In the following literaturereview section, several existing works considering the dataset description and several proposed approaches with their objectives have been discussed. The aim and scope of the study have been briefly mentioned in Sections 3 and 4 respectively. In Section 5, the significance of the study has been discussed. In Section 6, methods and results have been discussed. Section7 includes the conclusion of the study.

II. LITERATURE REVIEW

Hate speech is nothing but a hateful expression of hu- mankind. So practically it can be of different forms based on the nature of expression. Some of the harmful online content can be easily identified as attacks or promotes hate towardsa group or an individual member based on their actual or perceived aspects of identity, such as ethnicity, religion, and sexual orientation. Hate speech detection is nothing but a classification problem, where the training dataset should be precise considering several hate contents. Also, the training dataset size should be limited to reduce the algorithm's time complexity. So, dataset creation or selection highly controls the performance of the proposed approaches. There are severalhate speech datasets available like Hate Speech [1], Hate Speech and

Offensive Language [2], Implicit Hate [3], ETHOS(multilabEl haTe speecH detectiOn dataSet) [4], HateXplain [5] to sustain this area of research work.

Apart from dataset research, machine learning practitioners are mitigating the challenges of automatic hate speech detection by studying advanced neural networks. In [6], Kovács et al. proposed a deep neural network (DNN) to mitigate the poorly written text as DNN can learn various features. Convolution neural network (CNN) and recurrent layers had been combined in their deep NLP approach for auto-detection of social media hate speech data. The model had been implemented on HASOC2019 corpus, obtained F1 score was 0.63. In [6], result comparisons of several machine learning algorithms on HASOC dataset have been discussed.

In [7], MacAvaney et al. pointed out that different hate speech detection solutions suffer from the actual interpretationof the founding. So, decision-making seems unable to explain the reality. In [7], a multi-view support vector machine (SVM)was proposed to reduce such limitations discussed before by providing better interpretable decisions. They discussed several shortcomings of auto detect hate speech which could not be removed without proper context of the contents. Datasets like Stormfront, TRAC (Facebook) were explored using SVM and adetailed result discussion was given in [7].

Social media provides freedom of speech, people express their opinion without any bound. Sometimes this causes aconflict of opinions and hate speech which creates an un- wanted environment. Hate speech is a problem on multiple platforms, but there is not sufficient research on this. To address these issues, Salminen et al. [8] proposed several classification machine learning algorithms on datasets collected from four platforms: YouTube, Reddit, Wikipedia, and Twitter. They implemented logistic regression, Na¨ıve Bayes, SVM, XGBoost, and neural networks and Bag-of-Words, TF-IDF, Word2Vec, BERT, and their combination as word embedding strategies. They concluded that BERT feature importance analysis capability was most impactful for predictions, whereas XGBoost performed well with the best F1 score of 0.92. They claimed that the proposed concepts could be implemented as universal online hate speech detection applicable to multiple social media platforms.

Putri et al. [9] studied Twitter data where tweets relatedto region, race, politics, and ethnicity in Indonesia had been considered for hate speech categorization. They implemented classification algorithms like Na¨ıve Bayes, Multi-Level Perceptron, AdaBoost Classifier, Decision Tree and Support Vector Machine. They also studied the performance of the algorithms using SMOTE to handle imbalanced data. From model comparisons, it was concluded that the Multinomial Naive Bayes algorithm outperformed with the highest recall value of 93.2% and accuracy value of 71.2% in classifying hatespeech. Finally, Multinomial Naive Bayes without SMOTE was recommended for social media hate speech detection.

In some literature, researchers reviewed existing research works from different aspects using several algorithms to detectsocial media hate speeches. Jahan and Oussalah [10] provided a systematic literature review on NLP and deep learning technologies, NLP-specific terminologies, and text processing pipelines. Yin and Zubiaga [11] provided a detailed discussionon generalizable hate speech detection and a review studyon obstacles as well as solutions for social media platforms. Future directions to improve generalization has also been discussed in hate speech detection.

Text mining using NLP looks promising due to the useof pre-trained transformer models like BERT, GPT2, XLNet. Transformer models are deep learning models used in text embedding. BERT [12] was introduced by Devlin et al. to consider the context on the left and right sides in all deep layers. GPT2 [13] is a transformer-based language model introduced by Radford et al. (Open AI research) trained ona dataset of 8 million web pages. Another one is XLNet [14], which was introduced by Yang et al. to overcome the limitations of BERT. Model explanation is another important area getting popularity as one can understand the model behavior. Local Interpretable Model-Agnostic Explanations (LIME) [15] is such a model explanation tool introduced by Ribeiro et al. that can learn implemented algorithms locally around the predicted values. LIME has been implemented in many works. Like in [16], Park and Lee presented LIME as a weekly-supervised text classification to get more streamlined and effective predictive models. In another work, Mehta and Passi [17] proposed Explainable Artificial Intelligence (XAI) to detect hate speeches in social media data. They worked on Google Jigsaw data and HateXplain data. BERT + ANN and BERT + MLP were studied and explained by LIME.

It has been observed from the literature reviews that capturing the context of the contents is challenging as socialmedia texts are poorly written, have multi-linguistics, having emoticons, hashtags, and grammatical errors. Proper word embedding as well as methodologies which are good at capturingthe context can be utilized in these scenarios. AI explain is recently getting popularity due to its strength in explaining. It has been observed that robust explainable tools are required toexplore with better capability. Also, these kinds of tools need to develop to support several languages.

III. AIM AND OBJECTIVE

Cyberbullying is a growing menace in the world of theinternet. People often misuse the advantage of anonymity they have while using the internet. Some users simply write offensive language with no intention to violate human rights. This consists mainly of colloquial offensive words. Hate speech is what a person uses to harass other users targeting their religious beliefs, ethnicity, sexual preferences, etc. This study aims to explore different deep learning-based approachesto detect offensive and hate speech effectively by classifying English texts. Another objective here is that AI explanation hasbeen added to understand the predicted outcome as algorithmsare mostly black-box not explainable with bare eyes.

IV. SCOPE OF THE STUDY

Automated toxic speech detection in social media platformsis the best implementation of this work. If training data consider data collected from multiple platforms, then the trained model can be implemented in multiple platforms. Also,the trained model can be implemented in business forums for business-related discussions like brand awareness, customer connection, and product promotion

in English language. Plat- forms for social news, micro-blogging sites, and community blogs are good examples where viewers can freely provide opinions.

In case of social news platforms, hate speech on race, religion, and gender can be automatically flagged to reducethe spread of hate plague. Though toxic contents are quite uncommon in community blog platforms but still have some scopes or requirements to manage the spread of hate speech.

V. SIGNIFICANCE OF THE STUDY

This study is significant from the application point of view in social media platforms, and medical applications to autodetect hate and offensive speeches. It reduces human annotation efforts. This study also helps in explaining algorithms' test outcomes. The applicability of this study has already been discussed in the Scope of the study section.

VI. PROPOSED APPROACHES

In our work, tweet messages are explored. Basic EDA has been covered to understand the data. Word embeddings like TFIDF, keras encoding layer, BERT, GPT2, XLNet are used as different methodologies. Classifying algorithms like multinomial Naive Bayes, LSTM, BERT sklearn, TFGPT2Model (along with Dense and Dropout layer), and XLNetForSequenceClassification classifiers have been studied and finally, LIME has been implemented to check the model explanation.

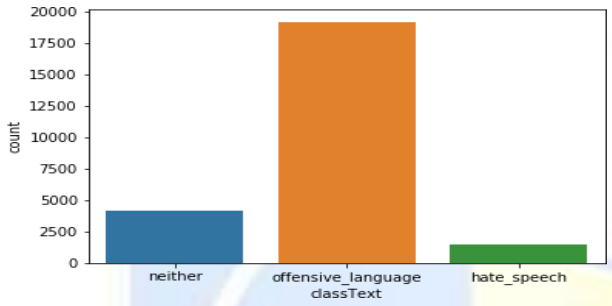

Fig. 1. Value counts plot of each label

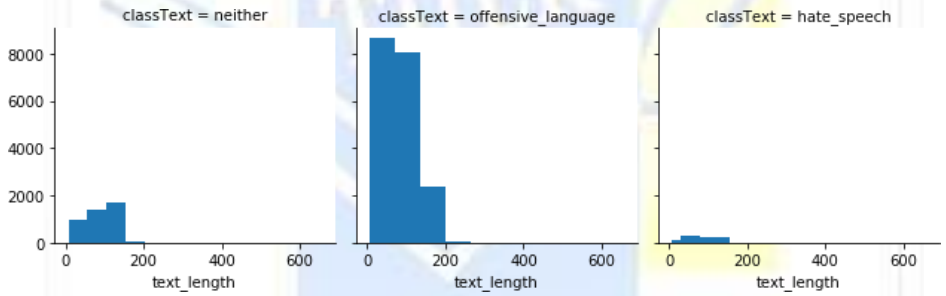

Fig. 2. Text lengths with number of tweets in each category

Dataset descriptions, methodologies, and results are describedin the following sections.

1) Dataset description

The dataset contains tweets with annotations 0, 1, and 2. Theclass label representations are 0 - hate speech, 1 - offensive language, and 2 - neither. There are 24784 non-null tweets withtheir defined classes. In this study, hate speech and offensive labels have been converted into 'hate & offensive' and neitherinto 'normal' to obtain the binary classification problem. Fig. 1depicts the value counts of each label 0, 1, 2.

The length of most of the tweet messages is less than 200. From Fig. 2, it can be observed that most of the tweets with lengths greater than 300 are belongs to offensive language category.

2) Method descriptions

1) Pre-processing

After converting the three classes into a binary classification problem, the value counts of each label are given in Fig. 3. In the pre-processing of the texts, emoticons, emojis, special characters, urls, hashtags, numbers, etc. all have been removed.

2) Model building

In this step, all the used algorithms and the three pipelines used in this study will be discussed. In thecase of natural language processing, word embedding is the next important step after pre-processing as computeror machine learning algorithms understand the texts through embedding. Several embeddings like TFIDF,transformers like BERT, GPT2, XLNet and several classifying algorithms are briefly discussed below.

2.1. TFIDF

In TFIDF, the score for each word has been calculated to quantify based on the word's importance in the document.

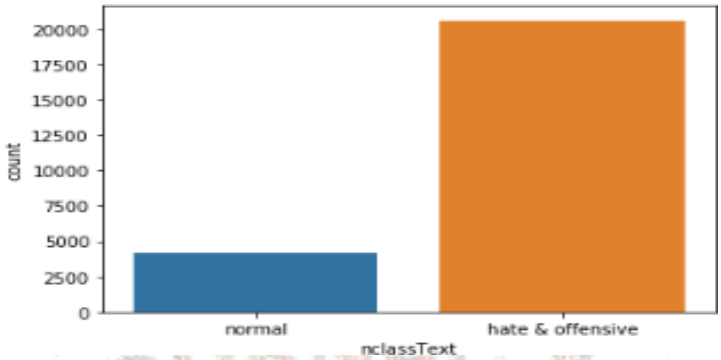

Fig. 3. Value counts plot of each label after converting three class into binary classification problem

TF represents the frequency ofa term that appeared in the document and IDFrepresents relatively rare occurrence information ofa word in the corpus. TFIDF together representthe importance of a term in the given text. TFand document frequency (DF) for each term in the document can be formulated as follows:

$$tf(t, d) = count\ of\ t\ in\ d\ /number\ of\ words\ in\ d\ df(t) = occurrence\ of\ t\ in\ N\ documents \quad (1)$$

To minimize the explode effect in the case of a large corpus and to avoid the divide by 0, IDF(t)has been calculated as $log(N/(df(t) + 1))$. So finally the TFIDF formula is given below:

$$TFIDF(t, d) = tf(t, d) * log(N/(df(t)+1)) \quad (2)$$

Here, TFIDF has been used together with Multinomial NB to hate and offensive speech detection in one of the implemented algorithms.

2.2. BERT

This is a transformer-based model. It is only an encoder-like encoder of transformers. The BERTis designed in such a way that it can capture the word's importance from the context. Here, contextmeans the most occurrence of the word after whichword. Language models read languages in one direction, either left-to-right or right-to-left, butBERT works differently. It has been designed ina way that it reads in both directions at once. Open-source BERT was introduced by Google. It has been implemented in multiple domains andmany languages. It has been used for sequential language generation as well as natural language understanding.

Here, BERT has been used as BERT encoder and BERT classifier based on python sklearn package.

2.3. Multinomial NB

This is a probabilistic approach to the data classification problem. It is a popular Bayesian learning approach for NLP classifications. The likelihoodof each tag of words has been considered and the maximum chance for the tag is considered further.

2.4. LSTM

Long short-term memory is a recurrent neural network (RNN) type architecture developed to avoid gradient vanishing and gradient exploding problems. LSTM is capable of recognizing pat-terns in long sequence-like data. LSTM consistsof multiple modules, in each module there are four interacting layers. Four layers are forget gate, input gate, memory cell, and output gate. There aremultiple variants of LSTM, e.g. Peephole connectedLSTM, Coupled forget and input gate LSTM, gatedrecurrent unit (GRU), etc.

In this work, one of the implemented algorithms is LSTM used inside tensorflow keras layer, where keras embedding layer has been used as word embedding tool.

2.5. GPT2Tokenizer, TFGPT2Model (Dense- Dropout)

GPT2 is a unidirectional language modelling pre- trained on a very large text dataset. GPT2Tokenizeruses byte-level byte-pair embedding. It treats spaces like parts of the tokens, a bit like sentencepiece. GPT2Tokenizer provides absolute embedding positions of the texts padded onthe right side. GPT2Model provides outputs in raw hidden states without a specific head on top. It has a pre-trained transformer architecture using attention mechanisms to focus on the selective texts that seem to be more relevant for classification in case of text classification problems.

In this work, GPT2Tokenizer, TFGPT2Modelalong with one mean reduce layer, Dense, and Dropout layer have been added in the model pipeline.

2.6. XLNetTokenizer, XLNetForSequenceClassification

XLNet is a permutative language modelling based on a generalized autoregressive model to createa bidirectional contextual representation of texts. XLNetForSequenceClassification is a normal XL- Net model with an added single linear layer on top of the XLNet outcome.

In this work, "xlnet-base-cased" pre-trained model has been implemented where the number of labels is 2.

2.7. LIME

It has been introduced by Marco Ribeiro in 2016 [15] to explain the prediction performances by machine learning algorithms, which cannot be explained or interpreted by human understanding. It works for any machine learning algorithms as per the name model-agnostic and it explains a small part of ML function as per the name local interpretable. For humans to trust the ML model out- come, it is necessary to understand why the model is giving a specific outcome. Model interpretabilityreveals some important issues like data leakage, model bias, and robustness. LIME provides themodel explain capability by explaining black box ML algorithms. One can fine-tune the model after checking the outcome explanation of ML models.

3) Proposed Approaches and result discussions

Above-described methodologies have been used to buildfive pipelines, which are described below. In each modeldiscussion, the obtained result has been given.

1) TFIDF - Multinomial NB

After pre-processing the labeled data, train and testdata are split into 80%-20% ratio. Then, count vectorizer and TFIDF transformation have been used to get word embeddings. Then sklearn MultinomialNB has been used as a classifier where $\alpha = 1.0$. Similarly, test data are vectorized and transformed and used for prediction. Model accuracy for testingdata is 84%.

2) keras encoding - LSTM

Same splitting rule for training and testing datahas been followed the same set of pre- processing as the previous model. In this model, to create LSTM model, KerasClassifier has been used together with sklearn pipeline. Tokenizer and pad sequences from keras pre-processing have been used in keras embedding layer. A dense layerwith one node and sigmoid activation has beenused as the original dataset has been convertedto a binary classification problem. Then model has been compiled with loss 'binary crossentropy', 'adam' optimizer and 20 epochs. Model accuracy for testing data is 92.8%.

3) BERT sklearn classifier

Same splitting and pre-processing have been followed in this case. This model is scikit-learn wrapper to fine-tune the BERT model for text and token sequence developed by Charles Nainan and Ezequiel Medina [18]. The whole package is nicely optimized with coding structure, and execution time. Configurable multilayer perceptron(MP) has been used as a classifier. It includes token sequence classifier for NER, PoS, and chunking tasks. Default bert model 'bert-base-uncased' has been used together with some parameters (= input values) like max seq length (= 128), train batch size (=16), epochs (=20), etc. Testing accuracy for this dataset is 94.05%. Thismodel is easy to implement and can be interpreted easily by LIME.

4) GPT2 based classifier

Same splitting and pre-processing have beenfollowed in this case. This model is using GPT2Tokenizer, TFGPT2Model from pre-trained transformer model "gpt2" with MAX LENGTH 17, one Dense and Dropout layers, epochs (=20), etc. Adam optimizer with base learning rate has been used as a model optimizer and SparseCategoricalCrossentropy as loss function has been used. Testing accuracy for this datasetis 90.48%. This model is easy to implement and can be interpreted easily by LIME.

5) XLNet based hate speech classifier

Same splitting and pre-processing have been followed in this case. This model is using XLNetTokenizer, XLNetForSequenceClassification from pre- trained transformer model "xlned-base-cased" with MAX LENGTH 128. In the XLNetForSequenceClassification architecture, there are two Dropout layers on top of one LayerNorm, two linear layers, and one Dropout layer with 'gelu' activation in feed-forward step. AdamW as an optimizer and 15as epoch numbers have been used. Testing accuracyfor this dataset is 95.64%. This model is classifying test inputs correctly in comparison to other implemented algorithms in this study. This model is easy to implement and can be interpreted easily by LIME.

Test Accuracy: 84.47%

	precision	recall	f1-score	support
normal	0.97	0.08	0.14	749
hate & offensive	0.84	1.00	0.91	3712
accuracy			0.84	4461
macro avg	0.90	0.54	0.53	4461
weighted avg	0.86	0.84	0.79	4461

Fig. 4. TFIDF-Multinomial NB classification report

Test Accuracy: 92.8%

	precision	recall	f1-score	support
normal	0.79	0.77	0.78	833
hate & offensive	0.95	0.96	0.96	4124
accuracy			0.93	4957
macro avg	0.87	0.87	0.87	4957
weighted avg	0.93	0.93	0.93	4957

Fig. 5. keras encoding - LSTM classification report

Test Accuracy: 94.05%

	precision	recall	f1-score	support
normal	0.84	0.79	0.82	833
hate & offensive	0.96	0.97	0.96	4124
accuracy			0.94	4957
macro avg	0.90	0.88	0.89	4957
weighted avg	0.94	0.94	0.94	4957

Fig. 6. BERT sklearn classifier classification report

Test Accuracy: 90.48%

	precision	recall	f1-score	support
normal	0.75	0.65	0.70	833
hate & offensive	0.93	0.96	0.94	4124
accuracy			0.90	4957
macro avg	0.84	0.80	0.82	4957
weighted avg	0.90	0.90	0.90	4957

Fig. 7. GPT2 classifier classification report

Test Accuracy: 95.64%

	precision	recall	f1-score	support
normal	0.87	0.86	0.87	833
hate & offensive	0.97	0.98	0.97	4124
accuracy			0.96	4957
macro avg	0.92	0.92	0.92	4957
weighted avg	0.96	0.96	0.96	4957

Fig. 8. XLNet classifier classification report

4) Hate speech explains

To explain the hate detection capability of the used algorithms, two tweets have been selected and all the respective results are discussed following. Two tweets after pre-processing are ’khloee s new backpack that we colored last night http tco tyndjcsq ’ and ’the homie reallly love fat bitches like in love’, which are annotatedas 0 (normal) and 1 (hate & offensive).

1) TFIDF - Multinomial NB

In this modelling, accuracy is around 84%, Test accuracy, Precision, Recall, F1-score are given in Fig. 4. For the selected examples, LIME explanations are given in Fig. 9 and 10.

Fig. 9. TFIDF-Multinomial NB LIME explain example 1, actual label ‘normal’

Fig. 10. TFIDF-Multinomial NB LIME explain example 2, actual label ‘hate & offensive’

Fig. 11. keras encoding - LSTM LIME explain example 1, actual label 'normal'

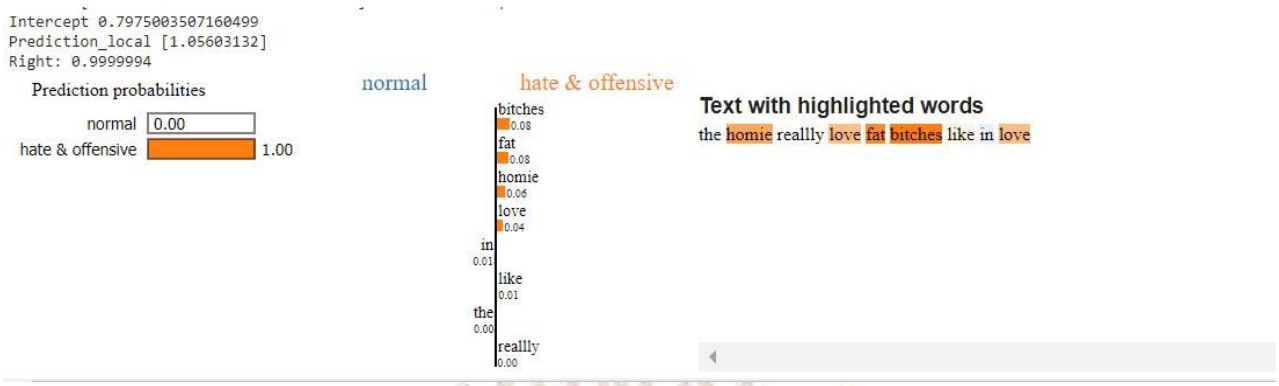

Fig. 12. keras encoding - LSTM LIME explain example 2, actual label 'hate & offensive'

Fig. 13. BERT_sklearn classifier LIME explain example 1, actual label 'normal'

Fig. 14. BERT sklearn classifier LIME explain example 2, actual label 'hate & offensive'

Fig. 15. GPT2 classifier LIME explain example 1, actual label 'normal'

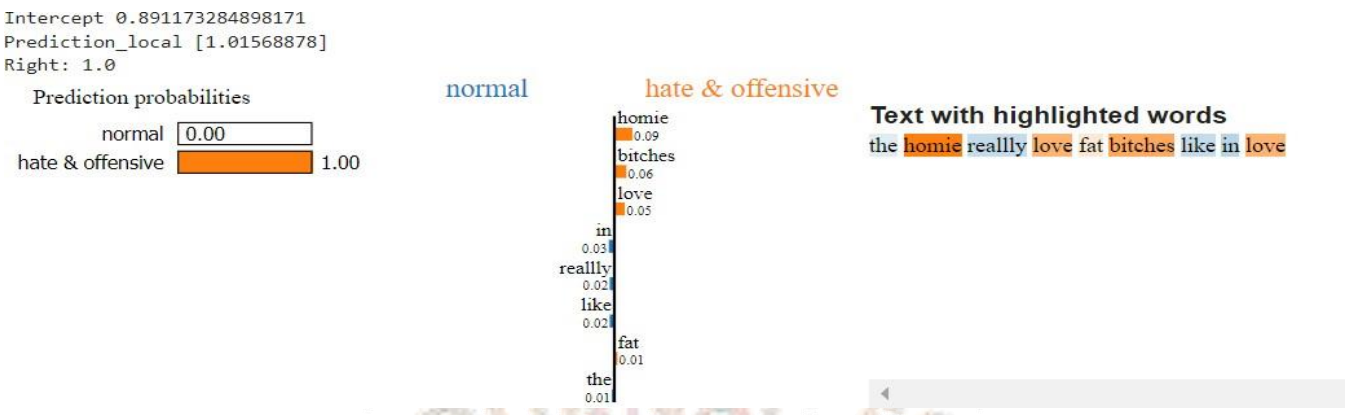

Fig. 16. GPT2 classifier LIME explain example 2, actual label 'hate & offensive'

Fig. 17. XLNet classifier LIME explain example 1, actual label 'normal'

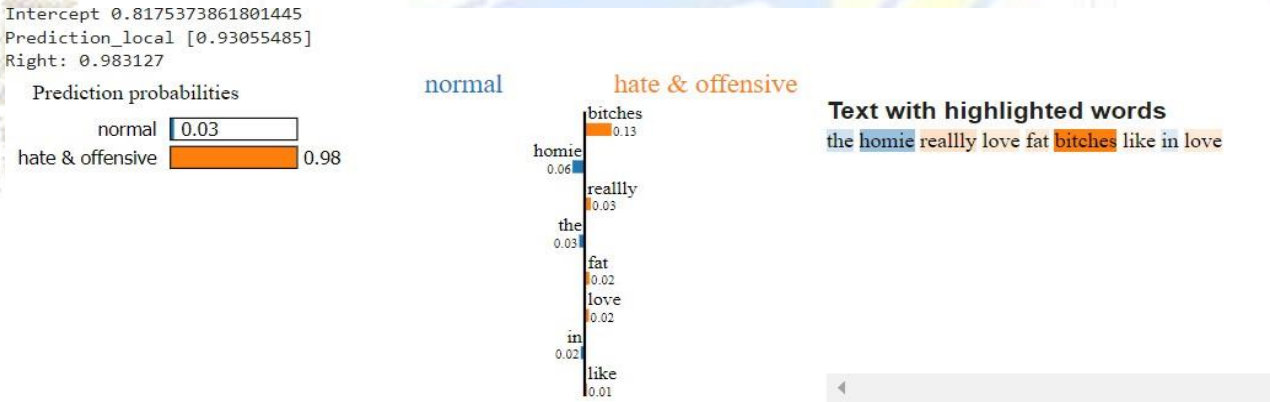

Fig. 18. XLNet classifier LIME explain example 2, actual label 'hate & offensive'

2) keras encoding - LSTM

This algorithm is performing better than TFIDF- Multinomial NB. It is correctly identifying the tweets' labels. Test accuracy, Precision, Recall, andF1-score are given in Fig. 5. LIME explanationsfor both examples depend on many words. LIME explanations are shown in Fig. 11 and 12.

3) BERT sklearn classifier

This algorithm is the second best in our study. Testing accuracy is 94.05%. Test accuracy, Precision, Recall, and F1-score are given in Fig. 6. FromLIME explanations in Fig. 13 and 14, it can be said that the algorithm correctly identifies the words which are responsible for the annotated categories.

4) GPT2 based classifier

Testing accuracy is 90.48%. Test accuracy, Precision, Recall, and F1-score are given in Fig. 7. LIME explanations for both examples depend on many words. LIME explanations are shown in Fig. 15 and 16.

5) XLNet based hate speech classifier

This algorithm outperforms the other algorithms. Testing accuracy is 95.64%. Test accuracy, Precision, Recall, and F1-score are given in Fig. 8. FromLIME explanations in Fig. 17 and 18, it can be saidthat the algorithm correctly identifies the words locally responsible for the predicted categories.

VII. CONCLUSION

In this study, social media hate and offensive speech detection have been explored, which is an NLP problem getting popularity due to flooded non-structured media data. Another important thing of this study is that online hate and offensive speech detection can remove the difficulty due tothe toxic spread of hate or offensive language. Advanced text embedding and deep ML algorithms as well as manually annotated data can be life savers in such situations. Also, hate to explain or the deep learning algorithms outcome explain is an important study to get robust classification or identificationof the hate and offensive speeches.

Several embedding and deep learning networks for model building have been used. Good accuracy has been observed forthe social media tweet messages dataset, which is a labeled data, and the messages are not properly structured. This work also includes AI explanation of all the models for some fixed number of messages using LIME. It has been observed that theperformance of models, XLNet and BERT sklearn classifier are outperforming. LIME provides a good explanation of the model behaviors discussed already.

Machine learning algorithms and LIME are black-box tools. Implementing LIME or similar tools for advanced modelsincluding convolution neural networks, and recurrent neural networks in the model-building layers can be studied as such pipelines are good in text classification problems. Also, similarplatforms like social media for different domains can be explored in the near future.

VIII. REFERENCES

[1] O. Gibert, N. Perez, A. Garćıa-Pablos, M. Cuadros, ”Hate Speech Dataset from a White Supremacy Forum”, https://arxiv.org/pdf/1809.04444.pdf.

[2] T. Davidson, D. Warmsley, M. Macy, I. Weber, ”Automated Hate Speech Detection and the Problem of Offensive Language”, https://arxiv.org/pdf/1703.04009v1.pdf

[3] M. ElSherief, C. Ziems, D. Muchlinski, V. Anupindi, J. Seybolt, M. De Choudhury, D. Yang, ”Latent Hatred: A Benchmark for Understanding Implicit Hate Speech”, https://github.com/gt-salt/implicit-hate

[4] I. Mollas, Z. Chrysopoulou, S. Karlos, G. Tsoumakas, ”ETHOS: an Online Hate Speech Detection Dataset” , https://arxiv.org/abs/2006.08328

[5] B. Mathew, P. Saha, S. M. Yimam, C. Biemann, P. Goyal, A. Mukherjee, ”HateXplain: A Benchmark Dataset for Explainable Hate Speech Detec- tion”, https://arxiv.org/pdf/2012.10289v2.pdf

[6] G. Kovács, P. Alonso, R. Saini, ”Challenges of Hate Speech Detection in Social Media”, SN COMPUT. SCI. 2, 95 (2021), https://doi.org/10.1007/s42979-021-00457-3

[7] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, O. Frieder, ”Hate speech detection: Challenges and solutions”, PLoS ONE 14(8):e0221152, 2019, https://doi.org/10.1371/journal.pone.0221152

[8] J. Salminen , M. Hopf, S. A. Chowdhury, S.-gyo Jung, H. Almerekhi, B. J. Jansen, ”Developing an online hate classifier for multiple so- cial media platforms”, Hum. Cent. Comput. Inf. Sci. 10, 1 (2020), https://doi.org/10.1186/s13673-019-0205-6

[9] T. T. A Putri, S. Sriadhi, R. D. Sari, R. Rahmadani, H. D. Hutahaean,”A comparison of classification algorithms for hate speech detection”,IOP Conf. Series: Materials Science and Engineering 830 (2020) 032006, https://doi:10.1088/1757-899X/830/3/032006

[10] Md S. Jahan, M. Oussalah, ”A systematic review of hatespeech automatic detection using natural language processing”, https://paperswithcode.com/paper/a-systematic-review-of-hate-speech- automatic/review/

[11] W. Yin, A. Zubiaga, ”Towards generalisable hate speech detection: a review on obstacles and solutions”, PeerJ Comput Sci. (2021); 7: e598, https://doi: 10.7717/peerj-cs.598

[12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, ”BERT: Pre-training of Deep Bidirectional Transformers for Language Under- standing”, Proceedings of NAACL-HLT (2019), pages 4171–4186,https://doi.org/10.48550/arXiv.1810.04805

[13] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, ”Language Models are Unsupervised Multitask Learners”, https://d4mucfpksywv.cloudfront.net/better-language-models/language models are unsupervised multitask learners.pdf

[14] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q. V. Le, ”XLNet: Generalized Autoregressive Pretraining for Language Understand-ing”, NIPS’19: Proceedings of the 33rd International Conference on Neural Information Processing Systems, December 2019, 517, pages 5753–5763, https://doi.org/10.48550/arXiv.1906.08237

[15] M. T. Ribeiro, S. Singh, C. Guestrin, “Why should I trust you?: Explaining the predictions of any classifier.” Proceedings of the 22ndACM SIGKDD international conference on knowledge discovery and datamining. ACM (2016)

[16] S. Park, J. Lee, ”LIME: Weakly-Supervised Text Classification Without Seeds”, Proceedings of the 29th International Conference on Compu- tational Linguistics, October 2022, 2022.coling-1.91, pages 1083–1088, https://doi.org/10.48550/arXiv.2210.06720

[17] H. Mehta, K. Passi, ”Social media hate speech detection using ex- plainable artificial intelligence (XAI)”, Algorithms, 15, 291 (2022), https://doi.org/10.3390/a15080291

[18] C. Nainan, E. Medina, https://github.com/charles9n/bert-sklearn