International Journal of Computer Trends and Technology
ISSN: 2231 – 2803 / https://doi.org/10.14445/22312803/IJCTT-V71I2P101

Volume 71 Issue 2, 1-7, February 2023

Original Article

Hinglish Profanity Filter and Hate Speech Detection

Nirali Arora[1], Aartem Singh[2], Laik Shaikh[3], Mawrah Khan[4], Yash Devadiga[5]

[1,2,3,4,5]*Computer Engineering, Rizvi College of Engineering, Maharashtra, India.*

Received: 15 December 2022 Revised: 19 January 2023 Accepted: 01 February 2023 Published: 11 February 2023

Abstract *- Freedom of speech is highly valued on the Internet, yet it is frequently also abused there. Events such as social media applications have become necessary instead of luxury. Many children and young teenagers at a tender age are introduced to this content and are prone to verbal abuse or exposed to illegitimate content or deadlines. There are no constraints or regulations to prevent the flow of hatred and violent content; this nature of the Internet inevitably gives rise to soul stigmas such as cyberbullying and cybercrime, which can impact the minds of children and young teenagers in society. The use of a profanity filter censors out all the above content. The hate filter recognizes hate speech and blocks any hateful material, making the application suitable for kids[2]. The paper proposes a hate speech detector along with a profanity filter algorithm. One of the simulation findings demonstrates that when considering profanity as noise input in the sentiment classification for review data, accuracy decreased by roughly 2%[10].*

Keywords *- Censorship, Corpus, Filtering, Profanity filtering, Tokenization.*

1. Introduction

Hate speech could be defined as any speech that targets a group of people based on race, religion, ethnicity, nationality, sexual orientation, or gender. Hate speech is used to propagate violence. It can be used to threaten people. It can cause people to feel anxious. This can also damage relationships between groups of communities. Children who are naive and innocent should not be burdened by trauma or fear.

The Hindi language is the national language of India. It is the most widely spoken throughout all of Central India, up to the northernmost parts [15]. The Internet has highly influenced communications in our generation's applications. Every language has a set of foul words that are inappropriate and are used to provoke or insult people. Profanity filters work on a dictionary. It also has certain negatives in that it serves as a forum for some users who engage in "Profanes," or knee-jerk reactions, trolls, and persecution [9]. Every human being has the right to be spoken to with respect or courtesy, and there is a need to ensure such filters make the internet much healthier. Our daily lives are becoming more reliant on communication technology thanks to the explosive expansion of mobile app platforms. Social media and other large online communication venues allow users to express themselves freely and occasionally anonymously [46]. However, users of social networking networks see the risk of receiving offensive content, such as offensive language, offensive images, and hate-prone sensitive information.

India is one of the countries with the highest usage of social media. With the rise of social media in India, the risk of profanities escaping the usual algorithms by using Hinglish expletives increases. To prevent such things, we need profanity filters. Profanity and hate speech filters have a long list of expletives. When community members use these words, the algorithm automatically replaces them with synonyms, symbols, or mutes that part of the audio. People cheat the usual filters by adding extra letters or slightly misspelling the words. This project aims to create a profanity filter for the Hinglish language, i.e., Hindi words with English script. The project will be able to filter out words that attempt to bypass the filter by adding extra letters. It also has an interactive GUI to test the filter. It will be able to filter out words that aim to bypass that filter by adding extra letters. Implementing this hate speech and profanity filter can eliminate foul language in comments, chats, or posts on social media, gaming, and streaming platforms. It makes a lot of social media applications safer and non-toxic for children.

2. Literature Review

Social media is a platform that enables two-way communication between people. Thus anyone with an internet account can express their thoughts with other social media users [26]. Social Networking, according to Hartshorn, is "the act of interaction," whereas social media allows users to converse with one another without having to interact in person, encouraging a sense of liberalism without defying cultural norms and leading to an increase in profanity usage online. While the majority of social media sites have included profanity filters to limit swearing, cyberbullies have consistently updated their profanity-related tactics to undermine the effectiveness of the filters already in place. Due to this, social media has a significant issue that, if left unchecked, might have considerable negative impacts on young users.

Most civilizations use profane language, although the definitions of what constitutes blasphemy differ by area. In the real world, using coarse language in conversation is widespread. People converse with one another online without necessarily meeting in person. Because of this, the use of profanity has increased in the virtual world, which encourages a sense of liberalism without defying cultural standards.

This opinion was supported by a study by [30], which compared statistics from the real world, where an approximation of 0.5% to 0.7% of obscene terms was spoken, with figures from social media, where one in every 13 tweets contained such phrases.

According to research [27], user-generated content that is derogatory in tone, cruel in intent, nasty, profane, and/or insulting is a common problem in online communities.

The majority of research studies have, however, run into a number of difficulties in addressing the issue of profanity in social media claims [45]. The three main issues affecting profanity detection are the different types of profanity filters, the methods used to detect profanity, and the categories of languages spoken across geographic boundaries. Meanwhile, according to [29], there are three categories for profane commercial filters: restricted entry whitelist filtering, free form whitelist filtering (for non-profane words), and blacklist filtering (for profane words) (text prediction of non-profane words).

Recently, the prevalence of profanity filters has become a daunting task. There is a stringent requirement for a robust technical model to keep profanity in check. Perusing various research papers makes it evident that there is a requirement for NLP and neural networks that would assist in building a technical model that would build a powerful neural network architecture coupled with the LSTM approach, which has shown to be a feasible solution. A study showed that deep learning is very effective in solving such problems. An impactful combination of CNN with LSTM helps construct the architecture that analyzes global semantics.[3]

The use of CNN with LSTM helps in constructing a strong architecture. The purpose of architecture was definitive; it flags the trained/textual contexts that are rude. The purpose of the architecture was quite definitive.

Another research employed word embedding using a fast text model. The character vector string was formed by adding the sub-string of the character vector.[4] Another research is pertinent to restrict cyberbullying. Their model was evaluated by two classification models, SVM (Support Vector Machine) and neural network.[5] Both were associated with TF/DF.

3. System Working

The system can be broken down into two different sections:

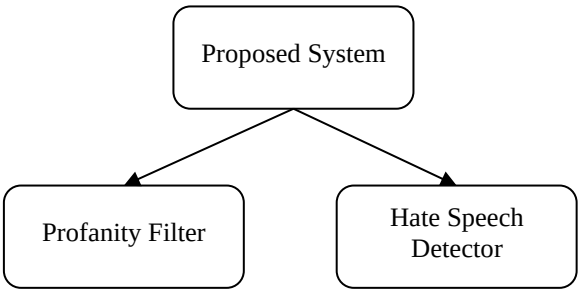

Fig. 1 Proposed System

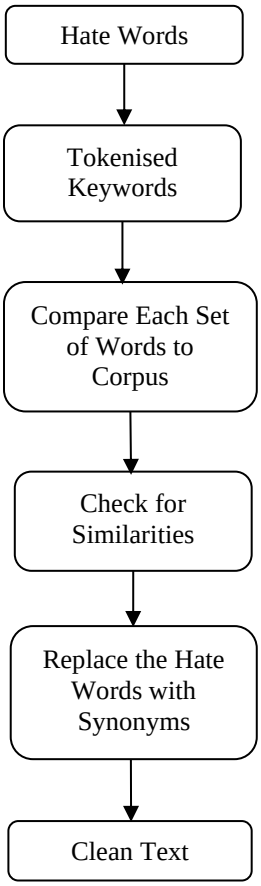

Fig. 2 Filtration Process

4. Existing Methods and a Comparison

The linguistic Rule-Based Approach is one of the methods used for this purpose. In 2014, C. J. Hutto et al. proposed an approach to classify sentiment using VADER, which is a rule-based approach [31]. At first, they created a list of lexical. Features that are highly sensitive to the sentiment of social media posts. Afterwards, they combined that list of lexical features with five general rules encapsulating syntactical and grammatical rules for presenting sentiment intensity.

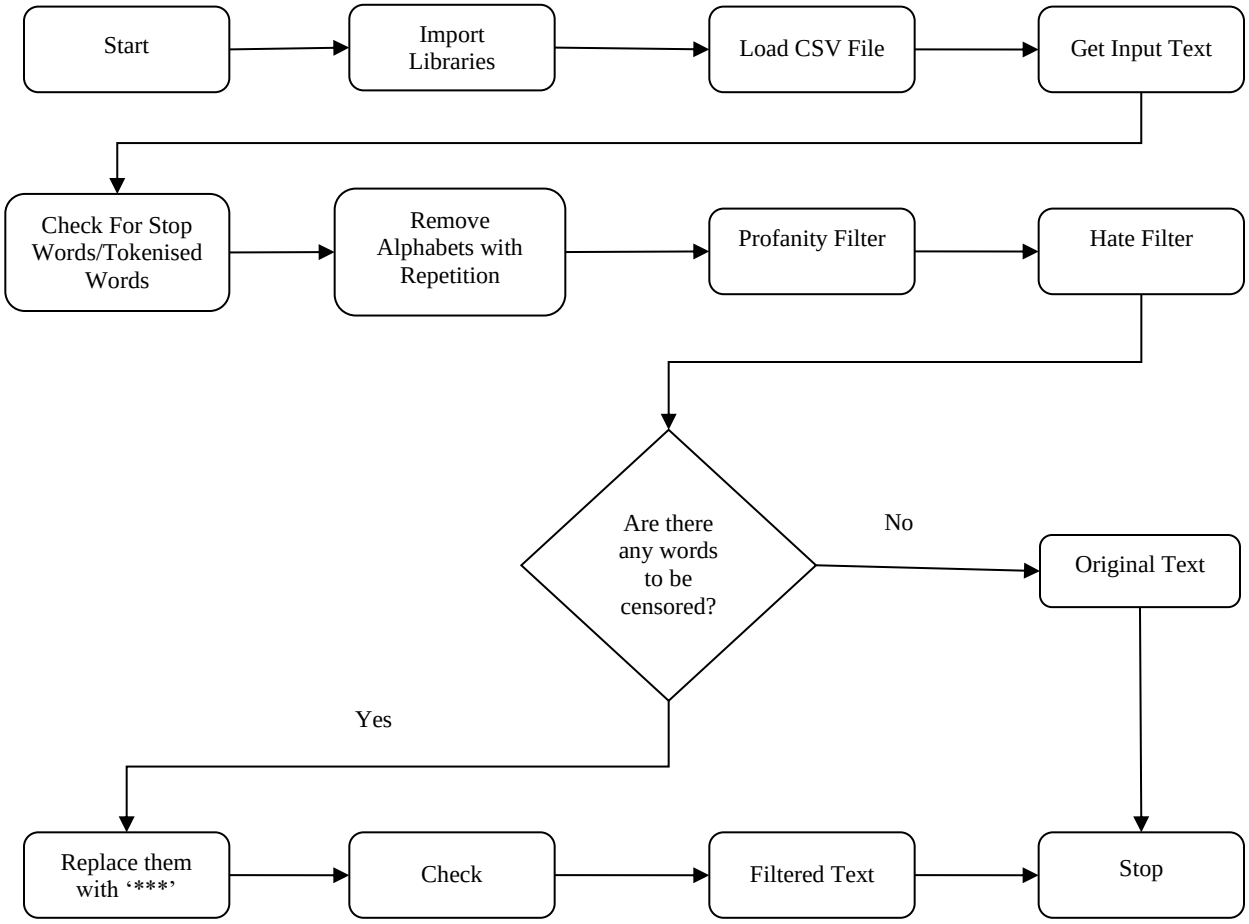

Fig. 3 Algorithm

At last, they found that VADER performed with 96% accuracy using the rule-based model on Twitter sentiments. In 2015, Dennis Gitariet al. introduced a rule-based approach to determine the sentiment analysis of social media text [32]. The three categories of nationality, religion, and race were used to categorise the hate speech issue. This paper's major goal is to create a classification model using sentiment analysis. The created model identifies and ranks the polarity of sentiment phrases in addition to detecting subjective sentences. Then they link the subjective and semantic characteristics to hate speech. Finally, they used the lexicon-based method to attain 71.55% precision.

The deep learning approach is another approach that can be used for this. Using deep learning, Hugo Rosa et al. (2018) devised a method to identify cyberbullying [33]. The training and testing data sets used in this paper were obtained from Kaggle. They started with CNN, which has some parallels to the problem of cyberbullying. A single-layer CNN is used as the initial layer, followed by a fully linked layer with a 0.5 dropout and softmax performance. After that, CNN-DNN-LSTM was integrated for optimal accuracy. To represent vectors, they used TFIDF. With Google Embeddings, they achieved 64.9% accuracy. Using the Bi-GRU-CNN-LSTM Model, Tin Van Huynh et al. (2019) proposed a method to identify hate speech [34]. In this study, they gathered information from Twitter and divided it into three categories (OFFENSIVE, HATE, and CLEAN). After cleaning the data, they implemented three neural network models, BiGRU-LSTM-CNN, Bi-GRU-CNN, and TextCNN, to identify hate speech. They achieved a 70.57% of F1 score as a result. Gambäck et al. (2019) used a deep learning technique to find hate speech on Twitter [35]. In this study, they gathered information from Twitter and separated it into four categories: sexism, racism, sexism and racism together, and non-hate speech. They used four CNN models that were trained using a combination of character n-gram, word2vec, and random vectors (word2vec and character n-gram). The model's accuracy was increased using a 10-fold approach by the author. With a 78.3% F-score, the word2vec-based CNN model outperformed the other three models.

A Word Embedding Approach can also be used for this. For a computer to understand and process natural language efficiently, the language needs to be transformed into

numbers that the computer can process. As the performance of natural language processing varies considerably depending on word representation, the conversion of words into number types is being extensively studied. Among these, the commonly used method is word embedding, which represents a word as a dense vector [36]. Methods of representing a word by a vector include sparse and dense representations. One-hot encoding is a method of representing a vector with sparse representation. The vector value expressed in spare representation mostly includes the number "0," and the number of dimensions equals the number of words to be trained. However, as sparse representation includes as many dimensions as the number of words for training, considerable space is wasted, and the meaning of words cannot be appropriately represented. Of late, several algorithms that represent vectors through dense representation by improving the above disadvantages have emerged [37]. With dense representation, the vector dimension can be matched with the numbers set by the user, and the vectors, called dense vectors, have real values [47]. As indicated by the concept, word embedding refers to a method of representing words in the form of dense vectors [39]. Representative algorithm models that can adopt word embedding include Word2Vec [37], GloVe [47], and FastText. The Word2Vec model represents words in the vector space through distributed representation using their semantics and syntactic characteristics. However, this model is disadvantageous because the vector values cannot be obtained for OOV, and training is impossible for infrequent words [40–42]. The FastText model performs training by dividing words into character-level to supplement such limitations. As in the case of the Word2Vec model, the FastText model examines the preceding and subsequent contexts with reference to the target word and performs training on words; however, because it also learns words by dividing them to character-level, the model can also be trained on word morphology information. FastText model training is performed by representing words using the Bag-of-Words (BoW) or n-gram model. At the beginning and end of the words to be learned, "<", ">" is inserted as separators, and the entire world is also contained in the BoW with the separator to enable the model to learn the overall meaning [43].

Comparing our approach to those mentioned above, we can say that our methodology is much simpler, easier to implement, and more efficient. The Rule Based system uses lexicons and rules to identify and categorize various sentiments. Our approach is similar to it as a list comparison followed here. One advantage of our methodology is the updation of the list. By adding new words to our database, we can easily update the list to include new words in Hindi, English, and Hinglish. It can be done manually or automated using basic methods. The Deep Learning approach was first applied to counter cyberbullies. Our hate speech detection and filtering help towards this exact cause. The accuracy of our filtering method is 100% for the words in the database. In addition, our method detects offensive and hateful phrases even after altering them through the repetition of letters. Word Embedding Approach represents a word as a dense vector. Our approach, in comparison, is considerably simple. It is easier to implement, gives results faster, and immediately filters hateful and offensive words. It is also easier to scale and rarely gives false positives. It means that if a phrase exists in the database, it will be filtered.

5. Algorithm

Step 0: START
Step 1: Designing the GUI using a custom tkinter.
Step 2: Getting the input text from the user.
Step 3: Pre-processing of the input text.
Step 4: Converting the input text to lowercase.
Step 5: Creating word tokens for the given input text.
Step 6: Removing the extra letters from word tokens which swear words.
Step 7: Creating a new csv file with modified swear words.
Step 8: Reading the csv file using pandas
Step 9: Converting the pandas dataframe to a numpy array
Step 10: Converting the numpy array to a 1-Dimensional array
Step 11: Creating an empty list "data_set1" and add all the extra letter swear words to this list.
Step 12: Creating another empty list "output" for storing the filtered(free from any hate or swear or bad words) text.
Step 13: Running a for loop from 0 to (length of filtered word token)
Step 14: Setting a flag "found" to True.
Step 15: Within this for loop, run another for loop from 0 to (length of a modified swear word list)
Step 16: if a word in the filtered word token and swear word list matches, then
Step 16 A: set "Found" to False.
Step 16 B: replace the founded swear word with ***.
Step 16 C: append this replaced word in the "output" array.
Step 16 D: break
Step 17: if there are no swear words found, then append the tokenize word to the "output" list.
Step 18: Convert the "output" list into the string and store it in the varibale "final_text"
Step 19: Return the "final_text"
Step 20: Display the "final_text" on the front end.

6. Results

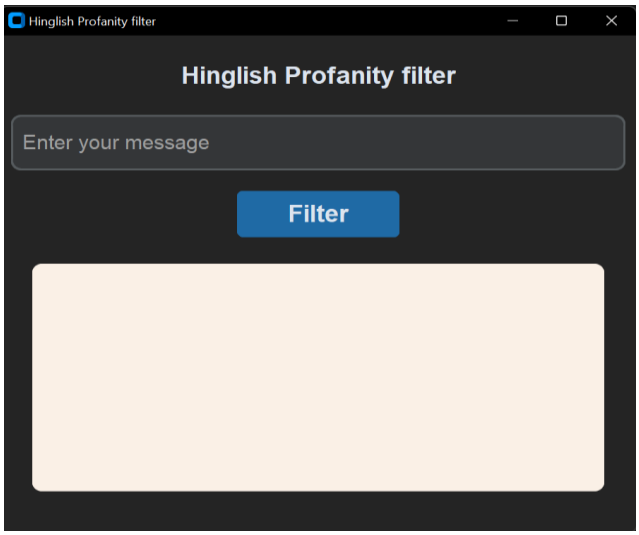

Fig. 4 GUI

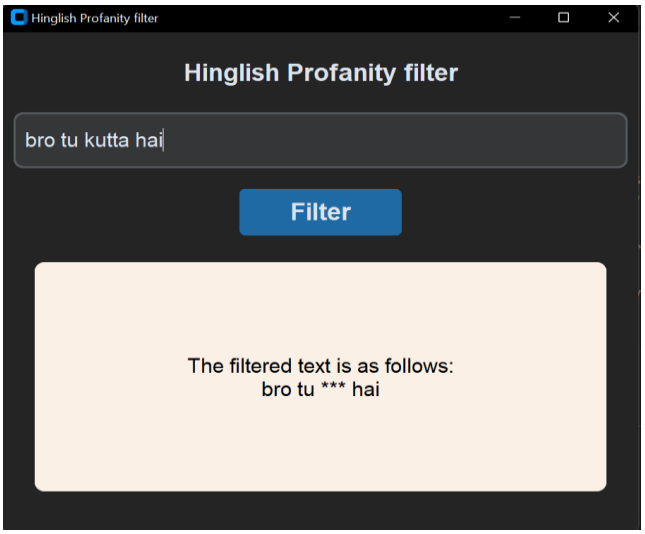

Fig. 6 Example 2

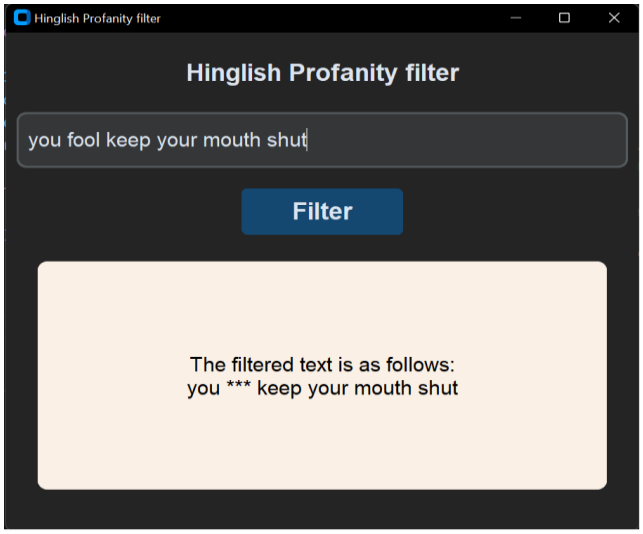

Fig. 5 Example 1

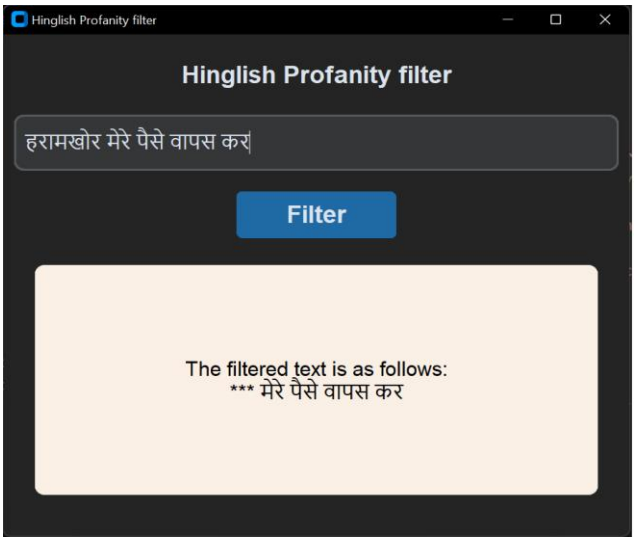

Fig. 7 Example 3

6. Conclusion

The machine learning and NLP-based profanity filter are successfully implemented. It can work seamlessly with any application. To reduce the misuse of social media platforms in daily conversation, chat profanity filtering software can be used to stop toxicity[8]. It could also be implemented in any e-commerce/entertainment/gaming application. Thus, this filter, if implemented into an application with many active users, leads to a positive impact on the community of users. This application will play a vital role in making the internet a much safer environment for people prone to being cyberbullied or swarmed with a series of toxic messages.

It is possible to improve the Profanity filter by increasing the extent of the corpus or the dataset used. Other future work may include resolving the different ways a user might bypass the given filter. It should be possible to prevent the methods like using alternate words and misspelling them to the extent that it would be difficult for the algorithm to capture but would still convey the intention, using short worms, etc.

References

[1] Elisabeth Métais et al., "Natural Language Processing and Information Systems," *26th International Conference on Applications of Natural Language to Information Systems*, vol. 12801, 2021.

[2] "Profanity Filters: Everything You Need to Know + Our Top 5 Picks," 2021.[Online]. Available: https://vpnoverview.com/internet-safety/kids-online/profanity-filters/

[3] A. D. Moore, "Python GUI Programming with Tkinter," 2021.

[4] Sanjana Kumar, Srikrishna Veturi, and Varun Sreedhar, "Profanity Filter and Safe Chat Application using Deep Learning," *International Research Journal of Engineering and Technology*, vol. 08 no. 07, 2021.

[5] MoungHo Yi et al., "Method of Profanity Detection Using Word Embedding and LSTM," *Mobile Information Systems*, vol. 2021, pp. 1-9, 2021. *Crossref*, https://doi.org/10.10.1155/2021/6654029

[6] Nur Chamidah, and Reiza Sahawaly, "Comparison Support Vector Machine and Naive Bayes Methods for Classifying Cyberbullying in Twitter," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 7, no. 2, pp. 338, *Crossref*, https://doi.org/10.10.26555/jiteki.v7i2.21175

[7] Sean MacAvaney et al., "Hate Speech Detection: Challenges and Solutions," *Plos One*, 2019. *Crossref*, https://doi.org/10.10.1371/0221152

[8] F Razali1 et al., "Implementation of Anti-Profanity Words in Mobile Application Platform," *International Colloquium on Computational & Experimental Mechanics*, vol. 1062, *Crossref*, https://doi.org/10.1088/1757-899X/1062/1/012026

[9] Raktim Chatterjee, Sukanya Bhattacharya, and Soumyajeet Kabi, "Profanity Detection in Social Media Text using a Hybrid Approach of NLP and Machine Learning", *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 7, no. 1, 2021.

[10] Cheong-Ghil Kim, Young-Jun Hwang, and Chayapol Kamyod, "A Study of Profanity Effect in Sentiment Analysis on Natural Language Processing Using ANN", Journal of Web Engineering, vol. 21, no. 3, 2022. *Crossref*, https://doi.org/10.13052/jwe1540-9589.2139

[11] Taijin Yoon, Sun-Young Park, and Hwan-Gue Cho, "A Smart Filtering System for Newly Coined Profanities by Using Approximate String Alignment", *10th IEEE International Conference on Computer and Information Technology*, pp. 643-650, 2010. *Crossref*, https://doi.org/10.1109/CIT.2010.129

[12] Abdulrehman A. Mohamed, George O.Okeyo, and Michael W. Kimwele, "Literature Survey: Data-driven Approach for Selection of an Ensemble Model of Profane Words Detection in Social Media", *International Journal of Scientific & Engineering Research*, vol. 9 no. 10, 2018.

[13] Zeerak Waseem, and Dirk Hovy, "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter," *In Proceedings of the NAACL Student Research Workshop*, *Association for Computational Linguistics*, pp. 88–93, 2016. *Crossref*, https://doi.org/10.18653/v1/N16-2013

[14] Sourya Dipta Das, Soumil Mandal, and Dipankar Das, "Language Identification of Bengali-English Code-Mixed Data Using Character & Phonetic Based LSTM Models," In Proceedings of the 11th Forum for Information Retrieval Evaluation, pp. 60–64, 2019. *Crossref*, https://doi.org/10.1145/3368567.3368578

[15] Shervin Malmasi, and Marcos Zampieri, "Challenges in Discriminating Profanity from Hate Speech," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 30, no. 2, pp. 187–202, 2018. *Crossref*, https://doi.org/10.1080/0952813X.2017.1409284

[16] Prashanth Kannadaguli, and Vidya Bhat, "Phoneme Modeling for Speech Recognition in Kannada using Multivariate Bayesian Classifier," *SSRG International Journal of Electronics and Communication Engineering*, vol. 1, no. 9, pp. 1-4, 2014. *Crossref*, https://doi.org/10.14445/23488549/IJECE-V1I9P101

[17] Sara Sood, Judd Antin, and Elizabeth F. Churchill, "Profanity use in Online Communities," *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1481-1490, 2012. *Crossref*, https://doi.org/10.1145/2207676.2208610

[18] Geetika Gautam, and Divakar Yadav, "Sentiment Analysis of Twitter Data Using Machine Learning Approaches and Semantic Analysis," *Seventh International Conference on Contemporary Computing*, pp. 437- 442, 2014. *Crossref*, https://doi.org/10.1109/IC3.2014.6897213

[19] Hate Speech - ABA Legal Fact Check - American Bar Association, [Online]. Available: https://abalegalfactcheck.com/articles/hate-speech.html.

[20] What are Profanity Filters? How to Implement Them? [Online]. Available: https://caseguard.com/articles/what-are-profanity-filters/

[21] NoSwearing.com. Noswearing.com - List of Swear Words, Bad Words, & Curse Words. 2019. [Online]. Available: https://www.noswearing.com/dictionary

[22] Ekaterina Chernyak, "Comparison of String Similarity Measures for Obscenity Filtering", aclanthology, vol. 04 no.06, 4 April 2017.

[23] Tobias Renwick, and Denilson Barbosa, "Detection and Identification of Obfuscated Obscene Language with Character Level Transformers," *The 34th Canadian Conference on Artificial Intelligence*, pp. 1–8, 2021. [Online]. Available: https://caiac.pubpub.org/pub/5uqi2h7k/

[24] Pushkar Mishra, "Author Profiling for Abuse Detection," *27th international conference on computational linguistics*," pp. 1088–1098, 2018. [Online]. Available: https://aclanthology.org/C18-1093

[25] Yi Chang et al., "Abusive Language Detection in Online User Content," *25th international conference on world wide web*, pp. 145–153, 2016. *Crossref*, https://doi.org/10.1145/2872427.2883062

[26] Sood S O, Antin J and Churchill E 2012 Conference on Human Factors in Computing Systems ACM 978-1-4503-1015

[27] Abdulrehman A Mohamed, Dr George O Okeyo and Dr Michael W Kimwele 2018 International Journal of Scientific & Engineering Research 9 (10) 2229-5518

[28] A. Abitha, and K Lincy, "A Faster RCNN Based Image Text Detection and Text to Speech Conversion," *SSRG International Journal of Electronics and Communication Engineering*, vol. 5, no. 5, pp. 11-14, 2018. *Crossref*, https://doi.org/10.14445/23488549/IJECE-V5I5P103

[29] Kate Knibbs, "Curses! People swear a lot on Twitter, and here are the most popular words," 2014. [Online]. Available: http://www.digitaltrends.com/socialmedia/popular-curse-words-twitter/

[30] C. J. Hutto, and Eric Gilbert, "VADER : A Parsimonious Rule-Based Model for Sentiment Analysis Of Social Media Text," *The Eighth International AAAI Conference on Weblogs and Social Media*, vol. 8, no. 1, pp. 216–225, 2014. *Crossref*, https://doi.org/10.1609/icwsm.v8i1.14550

[31] N. D. Gitari, Z. Zuping, H. Damien, & J. Long.
[32] Hugo Rosa et al., "A 'Deeper' look at Detecting Cyberbullying in Social Networks," *International Joint Conference on Neural Networks*, pp. 1–8, 2018. *Crossref*, https://doi.org/10.1109/IJCNN.2018.8489211T
[33] Tin Van Huynh et al., "Hate Speech Detection on Vietnamese Social Media Text using the Bi-GRU-LSTMCNN Model," *Computation and Language*, 2019. *Crossref*, https://doi.org/10.48550/arXiv.1911.03644
[34] Bjorn Gambäck, and Utpal Kumar Sikdar, "Using Convolutional Neural Networks to Classify Hate-Speech," *The First Workshop on Abusive Language Online, Association for Computational Linguistics*, pp. 85–90, 2017. *Crossref*, https://doi.org/10.18653/v1/W17-3013
[35] Tomas Mikolov et al., "Efficient Estimation of Word Representations in Vector Space," *Computation and Language*, 2013.[Online]. Available: http://arxiv.org/abs/1301.3781.
[36] Tom Young et al., "Recent Trends in Deep Learning Based Natural Language Processing, *Computation and Language* , 2017. [Online]. Available: http://arxiv.org/abs/1708.02709.
[37] Ayush Jain et al., "Detection of Sarcasm through Tone Analysis on video and Audio files: A Comparative Study On Ai Models Performance," *SSRG International Journal of Computer Science and Engineering*, vol. 8, no. 12, pp. 1-5, 2021. *Crossref*, https://doi.org/10.14445/23488387/IJCSE-V8I12P101
[38] Jeffrey Pennington, Richard Socher, and Christopher D. Manning "Global Vectors for Word Representation," *Conference on Empirical Methods in Natural Language Processing*, pp. 1532-1543, 2014. *Crossref*, https://doi.org/10.3115/v1/D14-116
[39] Piotr Bojanowski et al., "Enriching Word Vectors with Subword Information," 2017. [Online]. Available: http://arxiv.org/abs/1607.04606.
[40] Armand Joulin et al., "Bag of Tricks for Efficient Text Classification," 2016. [Online]. Available: http://arxiv.org/abs/1607.01759.
[41] Armand Joulin et al., "Compressing Text Classification Models," 2016. [Online]. Available: http://arxiv.org/abs/1612.03651.
[42] Tomas Mikolov et al., "Advances in Pre-Training Distributed Word Representations," 2017. [Online]. Available: http://arxiv.org/abs/1712.09405.
[43] ZENG Runhua, and ZHANG Shuqun, "Improving Speech Emotion Recognition Method of Convolutional Neural Networ*k*," *International Journal of Recent Engineering Science*, vol. 5, no. 3, pp. 1-7, 2018. *Crossref*, https://doi.org/10.14445/23497157/IJRES-V5I3P101
[44] Mike King, "Types of Profanity Filters for Online Safety," 2013. [Online]. Available: https://cleanspeak.com/blog/2013/03/28/types-of-profanity-filters-for-online-safety
[45] Ng Wai Foong, "Profanity Filtering in Speech," 2022. [Online]. Available: https://levelup.gitconnected.com/profanity-filtering-in-speech-41ae4fd6cccf
[46] Wikidocs, "Introduction to natural language processing using deep learning.", 2020. [Online]. Available: https://wikidocs.net/33520.