Automatic Hate Speech Detection using Ensemble Method and Natural Language Processing Techniques

1st N V Sahana
Computer Science and Engineering
Bangalore Institute of Technology
Bangalore, India
nvsahanarao@gmail.com

2nd Prerana R
Computer Science and Engineering
Bangalore Institute of Technology
Bangalore, India
preranaramesh1510@gmail.com

3rd Niharika S
Computer Science and Engineering
Bangalore Institute of Technology
Bangalore, India
niharikasathya23@gmail.com

4th Rakshitha S
Computer Science and Engineering
Bangalore Institute of Technology
Bangalore, India
rakshithasurendra30m@gmail.com

5th Bhanushree K J
Computer Science and Engineering
Bangalore Institute of Technology
Bangalore, India
bhanushreekj.bit@gmail.com

2023 International Conference on Network, Multimedia and Information Technology (NMITCON) | 979-8-3503-0082-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/NMITCON58196.2023.10276372

***Abstract*—The use of social media has exploded in recent years, and sharing information has numerous benefits for society. Hateful content has increased as a result of the increased usage of social media. Hate speech can exist in any sort of content designed to slander, dishonor, or incite hatred toward certain affecting various communities, organizations and individuals. It is critical to distinguish between hate and offensive texts to detect hate and offensive speech in any given text. Since no previous work has been done utilizing both English and Hinglish (Hindi- English code combined) data sets for multi class prediction, an ensemble model has been proposed to categorize any given input sentence into one of the three categories: hatred, offensive, or neither. Data collection, data pre-processing, feature extraction, and text categorization are significant processes needed for the proposed approach to detect hate speech. The data set collected for this model is a publicly available Twitter data set in English and Hindi-English code mix language to which data preprocessing is done. Extraction of n-grams as features is done using the term frequency-inverse document frequency (TFIDF) extraction method. Several single classification methods, such as Decision tree, SVC, Logistic Regression, Random Forest, and Naive Bayes are considered, evaluated and compared, combinations of various single classification models are done to form a better ensemble model with a combination of Random Forest and Support Vector Classifier. When tested against the testing dataset, the proposed ensemble model achieved an accuracy of 90.7.**

***Index Terms*—hate speech, n-gram, offensive speech, ensemble, TFIDF**

I. Introduction

In recent years, the widespread and increased use of social media catering to different categories of people has increased cyber hate and cyberbullies. Recent surveys suggest several examples of hate crimes, such as the US electing Donald Trump. The European Union Commission has adopted several actions, including legislation, to prevent the negative effects of hate and abusive speech. The Commission recently demanded that social media platforms sign an EU hate speech policy and remove any content that promotes hatred. It has become necessary to address this issue, and various works have been done to reduce the consequences of hate crimes.

Hate speech is directed at persons from specific social groups and is based on differences such as race, country of origin, and gender. Hate speech is linked to freedom of expression, individual rights, groups, and minorities, as well as ideas like dignity, freedom, and togetherness [9].

There is a rise in the code-mixing of languages on social media due to the ease and livability of regional material. When combined with the rich diversity of users on such platforms, social media is used for breeding negative thoughts. Given the diversity of regional languages, a conventional model will fail to recognize obscenities in content and discern its strength when applied to such mixes of languages. Because of the prevalence of such code-mixed information on social media in the politically heated and volatile nation, handling abusive and despised content in Hinglish requires special attention.

To address all these issues, an automatic hate speech detection system has been proposed which employs techniques of Natural Language Processing for feature extraction and an ensemble machine learning approach which is a model built by combining various single classification techniques that had been used to efficiently classify a given textual content into one of the three categories, i.e. either hate speech, offensive speech or neither.

The rest of the paper is organised as follows: Section II comprises a Literature Survey, Section III delves into the architecture of our proposed system, and Section IV, elaborates on the results and analysis of our model. Finally, Section V provides an overview of the conclusion.

II. Literature Survey

Tosev et al. [1] contributed to research pertaining to detecting hate speech on social media where diverse sparse and dense feature representations were inspected and used in combination with multi-level stacked ensemble learning using SVM, Logistic Regression and RF. The limitation is that the ensemble model gave a less accurate result. They suggested that pre-trained vector embedding techniques and other ML methods can be used in the future.

979-8-3503-0082-6/23/$31.00 ©2023 IEEE

M K Aljero et al. [2] proposed a novel hybrid mutation technique called the Genetic Programming (GP) model that was built on Darwinian principles to classify hate speech into hate or non-hate speech on social media. The proposed model treats text classification as a binary classification problem and has not yet been adapted to the multi-classification problem.

Shreelakshmi K et al. [3] proposed a machine learning model with fastText characteristics that provide superior feature representation with (SVM)-Radial Basis Function (RBF) classifiers for the purpose of detecting hate speech in Hindi-English code-mixed social media texts. Only binary classifications of texts as either ”hate” or ”non-hate” were present in the data collection under consideration.

Zafer Al-Makhadmeh et al. [4] proposed a hybrid NLP and ML technique to anticipate hate speech from social media networks. A powerful natural language processing optimization ensemble deep learning approach is used to test the acquired dataset (KNLPEDNN). Hate speech detection is classified using an English rather than a multilingual dataset.

Ibrahim et al. [5] presented a multi-label text classification for abusive language and hate speech detection using machine learning approaches with Naive Bayes (NB), Random Forest Decision Tree (RFDT) and Support Vector Machine (SVM) classifiers. Error analysis shows that an imbalanced dataset likely causes many false-negative errors.

Aditya Gayadhani et al. [6]. Proposed a model that uses n-gram features, weighted according to their term frequency-inverse document frequency (TFIDF) values to identify hate speech and abusive language on Twitter. Because other examples of inflammatory terms that did not contain offensive words were not taken into account and discovered that 4.8% of offensive tweets were inadvertently labelled as disliked.

Fauzi et al. [7] proposed identifying hate Speech in Indonesian Twitter data using two hard and soft voting ensemble methods and compared results with five stand-alone classifiers. Bag of Words and ensembles of feature sets were applied to reduce the risk of selecting a poor classifier, but the results were not significantly improved.

N. A. Setyadi et al. [8] proposed a model to classify offensive speech elements in a text using an Artificial Neural Network method optimized with a Backpropagation algorithm. Here only a Boolean classification of tweets to hate or non-hate was implemented.

K. K. Kiilu et al. [9] proposed an approach for detecting and classifying hateful speech that uses content created by self-identifying hateful communities on Twitter. The issue addressed is a limitation of the Twitter API for commercial research with limited approval.

H. Watanabe et al. [10] proposed a system that automatically recognizes malicious expression patterns and uses emotive criteria. It classifies tweets as malicious, offensive, and clean using the most popular unigrams. Words highly relevant to hatred are similar to those commonly used to insult, demean, or offend. Therefore, the binary classification of hate speech into two classes, ”hate” and ”offensive”, the accuracy of the tweet function certified as ”unigram” will reduce.

Sharma Sanjana et al. [11] 2018 describes how an ontological classification of hate speech is constructed based on the level of malicious intent and used to annotate Twitter data suitably using Naïve Bayes, SVM, and random forest classification. Class-based tagging of tweets makes the current task very one-dimensional.

Rakshita Jain et al. [12] 2021 Using NLP approaches for the two separate languages of English and Spanish on the two datasets supplied by PAN @CLEF 2021, have attempted to complete the aforementioned assignment. Three deep learning models and four machine learning classifiers—multinomial naive Bayes, K-Nearest Neighbours (KNN) classifier, logistic regression, and linear SVM—are included in this work.

Jitendra Singh Malik et al. [13] 2022 proposed a combination of CNN, Bi-LSTM, and MLP along with different embedding techniques, such as Term Frequency - Inverse Document Frequency (TF-IDF), Glove (Global Vector), and transformers- based embedding (e.g. BERT, ELECTRA, AlBERT, etc.), to give a detailed comparison in order to determine the most effective method for performing hate speech detection.

P. Preethy Jemima et al. [14] proposed an approach for hate speech detection using machine learning in which the category variable was changed from the entire data set using one-hot encoding, making it easier to analyze. The R squared values for the training and testing datasets are computed, and a KNN classifier model with 10 neighbor classes and the Euclidean distance between them was also employed.

Fetahi et al. [15] 2023 proposed an approach which helps in the training of an AI model through the investigation of latest datasets across different languages. After all the research and studies on different hate speech algorithms, It uses the feature selection, which in term helps to analyze most utilized and impactful features in the domain.

III. PROPOSED SYSTEM

The proposed model's architecture, aimed at categorizing Textual or Twitter content into three categories, namely hate, offensive, or neither, is illustrated in Fig. 1. This architecture encompasses a sequence of seven fundamental stages: data collection, data preprocessing, feature extraction, data partitioning, model training, model validation, and results analysis. Subsequent sections will delve into an exhaustive discussion of these crucial stages.

These seven phases delineate the operational workflow of the proposed system, which seeks to classify text data into the designated categories. The efficacy of this system hinges on several factors, including the quality of the acquired dataset, the suitability of the selected features, and the performance exhibited by the machine learning model during its training and validation processes. It's worth noting that fine-tuning and iterative development may be imperative to attain optimal results.

A. Dataset Collection

The English dataset collected for this research is a publicly available dataset from Kaggle. Hindi-English code mixed dataset is collected from GitHub. The dataset comprises data belonging to all three classes required.

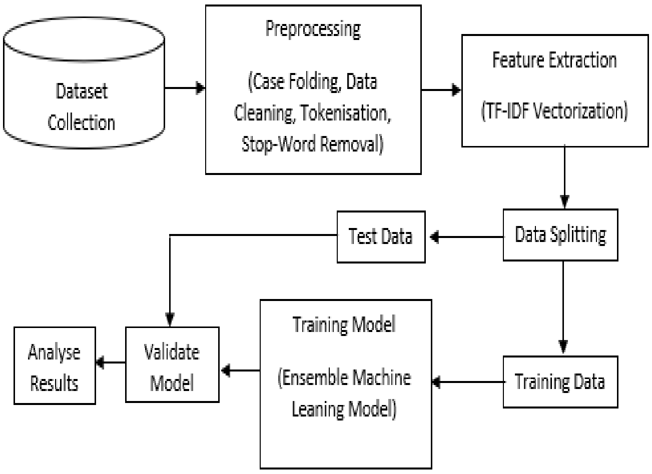
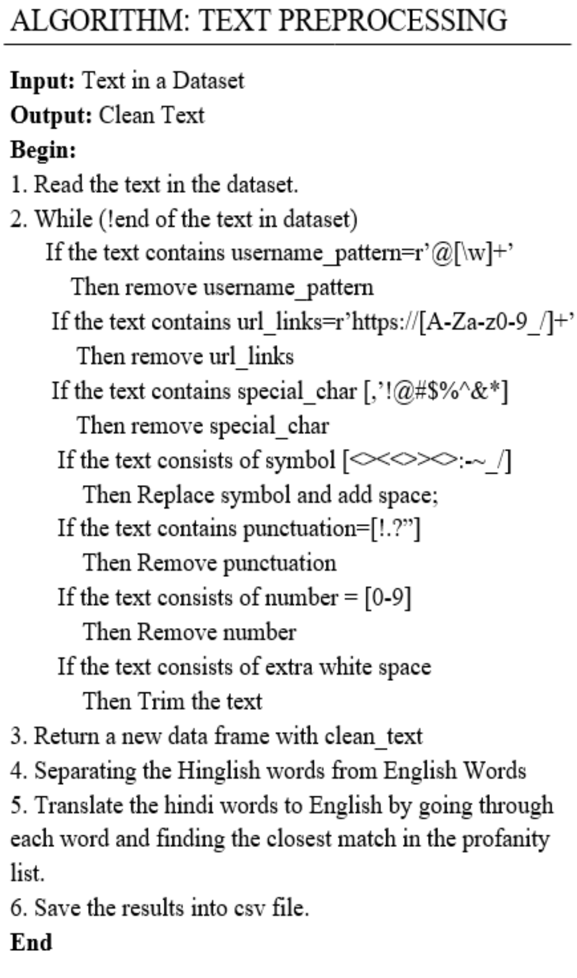


Fig. 1. The architecture of the proposed model.

B. Data Preprocessing

The data pre-processing techniques employed include case- folding, data cleaning, tokenisation and stop-word removal. This data pre-processing allows converting the collected dataset into a suitable format for further processing.

Fig. 2 displays the algorithm proposed for text pre-processing.



```
ALGORITHM: TEXT PREPROCESSING

Input: Text in a Dataset
Output: Clean Text
Begin:
1. Read the text in the dataset.
2. While (!end of the text in dataset)
     If the text contains username_pattern=r'@[\w]+'
        Then remove username_pattern
     If the text contains url_links=r'https://[A-Za-z0-9_/]+'
        Then remove url_links
     If the text contains special_char [,'!@#$%^&*]
        Then remove special_char
     If the text consists of symbol [<><<>><>:-~_/]
        Then Replace symbol and add space;
     If the text contains punctuation=[!.?"]
        Then Remove punctuation
     If the text consists of number = [0-9]
        Then Remove number
     If the text consists of extra white space
        Then Trim the text
3. Return a new data frame with clean_text
4. Separating the Hinglish words from English Words
5. Translate the hindi words to English by going through
each word and finding the closest match in the profanity
list.
6. Save the results into csv file.
End
```

Fig. 2. Algorithm for Text Pre-Processing

C. Feature Extraction

The machine learning algorithms could not work with the classification criteria if the raw textual content is provided. To understand the classification criteria, these algorithms require numerical features. Feature extraction is thus a vital step in text classification. This phase is required to identify the most important key features in the raw text data and present them in a numerical format. In this work, Term Frequency-Inverse Document Frequency (TF-IDF) features are extracted from raw data using an n-gram and a TF-IDF vectorizer. The TF-IDF vectorization, as the name suggests, multiplies the Term Frequency (TF) and Inverse Document Frequency (IDF) of a word to determine its score.

$$TF\,(t,d) = n/N \tag{1}$$

Where, n = *Number of Times t appears in d* and N = *Total number of terms in d*

$$IDF(t) = log\,\frac{N}{1+df}$$

$$TF - IDF(t,d) = TF\,(t,d) * IDF(t) \tag{2}$$

D. Data Splitting

Fig. 3 displays the dataset's general distribution and data set after splitting (i.e., Training set and Test set). To divide the pre- processed data, 80 per cent for training data and 20 per cent for test data, using the 80-20 rule. To train the classification model, training data is necessary. The proposed classifying model is evaluated using the testing data.

E. Training Model

The previously employed ensemble technique for hate speech detection suffers a significantly low accuracy. Hence, this study proposes an ensemble model by considering various single classification techniques such as KNN, Random-Forest Classifiers, RVM, SVM, etc., combining them to form the ensemble model. Then the training dataset is used to train the ensemble model built of the best single classifiers.

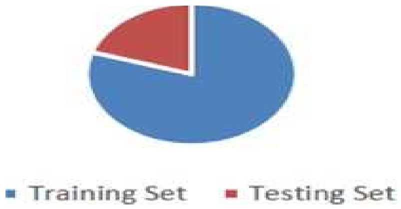


Fig. 3. Twitter Dataset Splitting

F. Validating Model

The model was validated by analyzing the accuracy obtained when testing data was provided to the ensemble model.

IV. RESULTS AND ANALYSIS

The results and analysis of the study on automatic hate speech detection is presented below, focusing on the evaluation of single classifiers and the development of an ensemble model.

Data Splitting and Train-Test Split: To evaluate the performance of the hate speech detection model, a common practice in machine learning, known as the train-test split, was adopted. The Scikit Learn library's Train Test Split module was utilized to perform an 80:20 train-test split. This means that 80% of the dataset was used for training the model, while the remaining 20% was reserved for testing its performance.

Evaluation of Single Classifiers: The assessment of several single classification algorithms commenced, selected based on recent studies that demonstrated success in classifying textual content. These algorithms were trained and evaluated using the dataset. The table 1 below provides a summary of the accuracy achieved by each of these classifiers:

TABLE. I. ACCURACY ACHIEVED WITH VARIOUS SINGLE CLASSIFIERS

Classifier Name	Accuracy
XGBoost Classifier	0.87
Random Forest Classifier	0.88
Naive Bayes Classifier	0.86
Logistic Regression	0.86
Decision Tree Classifier	0.84
Support Vector Classifier	0.88

The graphical representation depicted in Fig. 4 illustrates a comparative analysis of the results obtained through the utilization of each individual classifier employed in our model.

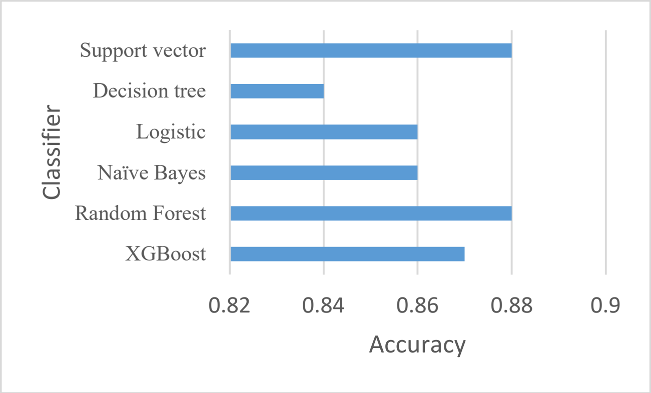


Fig. 4. Graphical Representation of Single Classifiers Vs Accuracy

The accuracy scores in the Table. 1 indicate how well each classifier performed in correctly categorizing text data into hate speech, offensive speech, or neither category. The Random Forest Classifier and Support Vector Classifier achieved the highest accuracy, both scoring 88%.

Ensemble Model Development: Ensemble methods, known for their ability to enhance performance, were explored in the creation of ensemble models using combinations of the aforementioned single classifiers. The objective was to construct a more robust and accurate model by leveraging the strengths of multiple classifiers.

Table. 2 below presents the results achieved with various ensemble models:

TABLE. II. ACCURACY ACHIEVED WITH VARIOUS ENSEMBLE MODELS

Ensemble Model	Accuracy
E (Random Forest, Support Vector, Logistic Regression)	0.89
E (Random Forest, Support Vector, Naive Bayes)	0.88
E (Random Forest, Support Vector)	0.97

The below Fig. 5 portrays a visual representation showcasing a comparative analysis of the outcomes achieved by employing different combinations of ensemble models within our system.

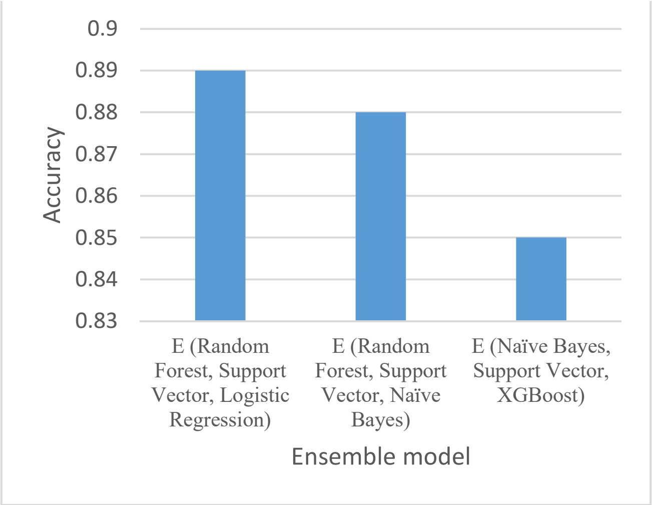


Fig. 5. Graphical Representation of Accuracy Vs Ensemble Model

Among the ensemble models considered, the combination of Random Forest and Support Vector Classifier demonstrated the highest accuracy, achieving an impressive accuracy rate of 97%. This combination significantly outperformed the single classifiers, as indicated in Table 1.

Web Application Development: To make the hate speech detection model accessible and practical, a web application was developed. This application takes textual content as input and utilizes the ensemble model to classify the text into one of three categories: Hate, Offensive, or Neither. This user-friendly tool can be used to detect and address hate speech and offensive content in various online platforms, contributing to a safer and more inclusive digital environment.

V. CONCLUSION

During the literature survey, it has been observed that till now no work has been carried out where both English and Hinglish datasets were used for the multiclass prediction of Sentences. Hence an ensemble model has been proposed that is built with a combination of Random Forest and Support Vector Classifier, which efficiently classifies English and Hinglish (Hindi-English code-mix) textual content given by the user into any of the three classes, i.e. Hate, Offensive or Neither using NLP techniques, i.e. n-feature Extraction which employs TF-IDF Vectorizer achieving a greater accuracy of 90.7% outperforming other ensemble models and single classifiers that were performed during this research, which can be seen by contrasting the outcomes displayed in Table. 1 and Table. 2.

The proposed ensemble model of the experiment combining Random Forest and Support Vector Classifier offers advantages such as leveraging strengths of each classifier, leading to enhanced performance in capturing complex patterns and variations in the text. Moreover, the model's ability to handle multiclass prediction effectively is valuable for real-world applications where text content can belong to multiple distinct categories and Robustness, making it a promising tool for text classification tasks.

However, the limitations related to data size, diversity, and generalization to other languages should be considered in future research to further improve the model's effectiveness and applicability in real-world scenarios.

REFERENCES

[1] Tosev, Darko, and Sonja Gievska. ”Multi-level stacked ensemble learn- ing for identifying hate speech spreaders on Twitter.” (2021).

[2] M. K. A. Aljero and N. Dimililer, ”Genetic Programming Approach to Detect Hate Speech in Social Media,” in IEEE Access, vol. 9, pp. 115115-115125, 2021, doi: 10.1109/ACCESS.2021.3104535.

[3] K. Sreelakshmi, B. Premjith, K.P. Soman, “Detection of Hate Speech Text in Hindi-English code mixed Data”, Procedia Computer Science, Vol. No. 171, Page No. 737-744, 2020.

[4] Al-Makhadmeh, Zafer, and Amr Tolba. ”Automatic hate speech detec- tion using killer natural language processing optimizing ensemble deep learning approach.” Computing 102, no. 2 (2020): 501-522.

[5] Ibrohim, Muhammad Okky, and Indra Budi. ”Multi-label hate speech and abusive language detection in Indonesian twitter.” In Proceedings of the Third Workshop on Abusive Language Online, pp. 46-57. 2019.

[6] Aditya Gayadhani, Vikrant Doma, Shrikant Kndre, Laxmi Bhagwat, “Detecting Hate Speech and Offensive Language on Twitter using Machine Learning: An N-gram and TF IDF based Approach”, IEEE International Advance Computing Conference(2018), 2018.

[7] Fauzi, M. Ali, and Anny Yuniarti. ”Ensemble method for Indonesian twitter hate speech detection.” Indonesian Journal of Electrical Engineering and Computer Science 11.1 (2018): 294-299.

[8] Fauzi, M. Ali, and Anny Yuniarti. ”Ensemble method for Indonesian twitter hate speech detection.” Indonesian Journal of Electrical Engineering and Computer Science 11.1 (2018): 294-299.

[9] N. A. Setyadi, M. Nasrun and C. Setianingsih, ”Text Analysis For Hate Speech Detection Using Backpropagation Neural Network,” 2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), 2018, pp. 159-165, doi: 10.1109/IC- CEREC.2018.8712109.

[10] K.K. Kiilu, Kelvin & Okeyo, George & Rimiru, Richard & Ogada, Kennedy. (2018). “Using Na¨ıve Bayes Algorithm in detection of Hate Tweets. International Journal of Scientific and Research Publications” (IJSRP). 8. 10.29322/IJSRP.8.3.2018.p7517.

[11] H. Watanabe, M. Bouazizi and T. Ohtsuki, ”Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection,” in IEEE Access, vol. 6, pp. 13825- 13835, 2018, doi: 10.1109/ACCESS.2018.2806394.

[12] Sharma Sanjana and Agarwal, Saksham and Srivatsava, Manish, “Degree based Classification of Harmful Speech using Twitter data”, Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), Page No. 106-112.

[13] Rakshita Jain1, Devanshi Goel1, Prashant Sahu1, Abhinav Kumar2 and Jyoti Prakash Singh,”Profiling Hate Speech Spreaders on Twitter”,CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021.

[14] Malik, Jitendra Singh, Guansong Pang, and Anton van den Hengel, ”Deep learning for hate speech detection: a comparative study”, arXiv preprint arXiv:2202.09517,2021.

[15] P. Preethy Jemima, Bishop Raj Majumder, Bibek Kumar Ghosh,Farazul Hoda, ”Hate Speech Detection using Machine Learning”,Proceedings of the Seventh International Conference on Communication and Electronics Systems (ICCES 2022) IEEE Xplore Part Number: CFP22AWO-ART;ISBN: 978-1-6654-9634-6

[16] E Fetahi, M Hamiti, A Susuri, V Shehu, A Besimi,”Automatic Hate Speech Detection using Natural Language Processing: A state-of-the-art literature review”, 2023 12th Mediterranean Conference on Embedded Computing (MECO). IEEE, 2023.