isrjournals

International Journal of Advanced Research in Computer Science Engineering and Information Technology

Volume: 6, Issue: 3,Special Issue: 2 ,Apr,2023 ,ISSN_NO: 2321-3337

DETECTION OF HATE SPEECH CLASSIFICATION ON SOCIAL MEDIA USING MACHINE LEARNING

Bhavani.B[1], Jeevitha.A[2], Sevanthi.R[3], Antony Sibiya Varghese.V[4]

UG Scholar[1 2 3], Department of CSE, GRT Institute of Engineering And Technology, Tiruttani, India

Assistant Professor[4], Department of CSE, GRT Institute of Engineering And Technology, Tiruttani, India

bhavanibalan81@gmail.com, anandanjeevitha@gmail.com, sevanthir6@gmail.com, cbyablue@gmail.com

Abstract - The everyday lives of many people have been deeply affected by digital media. Hate speech is a narrative intended to distract or mislead the audience. Because of a number of factors, including the rise of online social networks in recent years, hate speech has become more common in the online world. Users of online social networks may easily be affected by this hateful speech. Hate speech has become a social problem, sometimes spreading more rapidly than the truth. Individuals are unable to identify all instances of hate speech. Thus, it is necessary to employ a machine learning algorithm to automatically detect hate speech. The development of machine learning models involves the utilization of algorithms that distinguish between speech that is considered hate speech, hurtful speech, or neither. Among the various algorithms, the Gradient Boosting Algorithm yields the highest level of accuracy. As a result, it has been chosen for use in this project's launch. The Kaggle dataset employed to classify hate speech comprises characteristics such as tweet count, hate speech, offensiveness, neutrality, classification, and tweet content.

Keywords : Machine learning, Online, Hate speech, Algorithm

1.INTRODUCTION

The rapid dissemination of hate speech on social media is a harmful form of communication that may occur due to prejudice or conflicts between groups on both national and international scales. Hate crimes are actions carried out against individuals based on their real or perceived affiliation with particular groups. Facebook categorizes hate speech as an attack on a person's honor, including aspects such as their ancestry, nationality, or race. Twitter's guidelines prevent users from using tweets to intimidate or mistreat others based on their gender, religious beliefs, race, or other traits. YouTube takes measures to censor content that encourages violence or animosity towards specific individuals or groups, as well as content that is restricted based on age, social status, or disabilities. Research into hate speech that may be linked to internet radicalization or criminal activities is often conducted.

2.RELATED WORK

A multi-domain hate speech corpus (MHC) of English tweets containing hate speech against religion, nationality, ethnicity, and gender, covering a range of topics including politics, terrorism, technology, natural disasters, and human/drug trafficking, was created using deep learning algorithms. Each occurrence in the dataset was classified as either containing hate speech or not. A

stacked-ensemble-based hate speech classifier (SEHC) was presented using the most recent state-of-the-art models to detect hate speech from Twitter data. The suggested approach can serve as a reliable starting point for further research. Recurrent neural frameworks, such as the Gated Recurrent Unit (GRU), were used for hate speech detection. A variety of techniques, such as Word2Vec embedding and RNN-GRU, were used to attain effective experimental outcomes for hatred speech detection. The first benchmark collection on hate speech that addresses various facets of the problem was developed. Several deep neural network (DNN) frameworks, such as GRU and Convolution Neural Network (CNN), were used for Twitter hate speech detection. The CNN model performed the best for identifying hate speech in Arabic tweets. Multiple machine learning algorithms and feature engineering approaches were compared to evaluate their performance on a publicly available dataset with three different classes. The suggested method uses sentimental and semantic features to classify tweets into hateful, offensive, and clean by automatically identifying hate speech patterns and the most prevalent unigrams. Crowdsourcing was used to categorize a portion of comments into three distinct groups, and a multi-class classifier was trained to differentiate between them using lexical techniques to locate possibly offensive terms.

3.PROPOSED SYSTEM

3.1. DATA PRE-PROCESSING

Pre-processing refers to the alterations we make to our data before feeding it to an algorithm. The aim of data pre-processing is to transform incomplete or raw data into a complete and structured set of data that can be analyzed. Raw data is usually gathered from various sources and requires some processing before it can be used for machine learning models to improve their accuracy. For instance, certain machine learning algorithms such as the Random Forest algorithm cannot handle null values, so the data needs to be organized and any missing values must be addressed. To make sure that a single dataset can work with multiple Deep Learning and machine learning algorithms, the data needs to be formatted correctly.

3.2. DATAVALIDATION CLEANING PREPARING PROCESS

Data validation is an essential process that involves confirming and verifying the accuracy and quality of the collected data before it can be used for any purpose. This step ensures that the data is reliable and consistent. The process of data cleaning is crucial for identifying and removing errors and inconsistencies in the data, which can improve its value for analysis and decision-making. Therefore, it is essential to check the source data thoroughly before importing or processing it to ensure that it is error-free and reliable.

3.3. DATA VISUALIZATION

Visualizing data is an important set of methods for gaining a qualitative understanding. If you are exploring a dataset to discover patterns, errors, anomalies, and

Volume: 6, Issue: 3,Special Issue: 2 ,Apr,2023 ,ISSN_NO: 2321-3337

other characteristics, visualizations can be very useful. Unlike correlation and significance tests, data visualizations can use graphs and images to convey meaningful relationships, making the information more tangible and engaging for stakeholders.

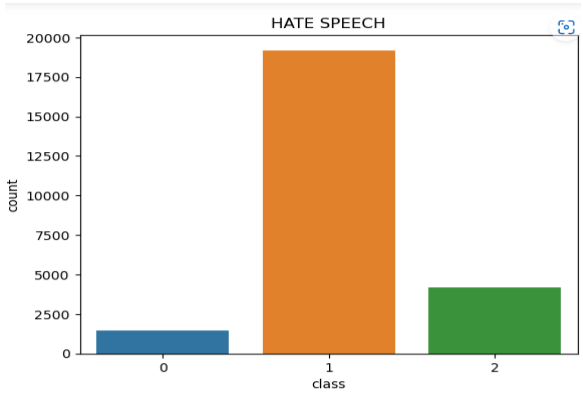

Fig 1. Data visualization graph

This graphic presents a bar chart data visualization that illustrates data. The horizontal axis shows the different categories, which are labeled as 0, 1, and 2, while the vertical axis displays the number of occurrences for each category.

3.4. ML MODEL DEVELOPMENT:

For the purpose of building a test harness, it is evident that scikit-learn in Python can be utilized. In order to effectively compare the performance of different machine learning algorithms, a reliable test harness is necessary. This test harness can serve as a reference point for your particular machine learning problems, and can be used for additional and alternative means of comparison. Each algorithm will have different efficiency characteristics, and the potential accuracy of each model on unseen data can be evaluated using techniques such as cross-validation. From the set of models you have created using these approximations, you must select one or two of the best. It is a good idea to visualize new information using various techniques to view data from different perspectives. Model selection follows the same principle. Before choosing one or two algorithms for finalization, you should evaluate the anticipated accuracy of your machine learning algorithms in a variety of ways. One method to do this is by utilizing various visualization techniques to display the average accuracy, variance, and other properties of the range of model accuracies.

3.4.1. RANDOM FOREST ALGORITHM

The popular guided learning approach employs a well-known machine learning algorithm known as Random Forest, which can be applied to solve both classification and regression problems. The algorithm uses the principle of ensemble learning, which involves combining various classifiers to handle complex problems and enhance model performance. Random Forest acts as a classifier that combines multiple decision trees, each applied to different segments of the dataset, to improve the predictive ability of the dataset. As its name implies, the algorithm constructs a forest of trees, and each tree predicts a target variable based on the values of input features. The final output of the Random Forest algorithm is the average of the predictions from all trees.

3.4.2. NAÏVE BAYES ALGORITHM

The Naive Bayes algorithm is a basic technique that utilizes probabilities for each

Volume: 6, Issue: 3,Special Issue: 2 ,Apr,2023 ,ISSN_NO: 2321-3337

characteristic that belongs to every class in order to make predictions. It is a supervised learning technique used to model a probabilistic forecasting problem. The naive bayes approach simplifies the calculation of probabilities by assuming that the probable result of every feature being linked to a known group outcome is separate from that of the other features. Despite being a powerful assumption, it leads to a quick and efficient technique. Naive Bayes is a statistical categorization method based on Bayes Theorem and is one of the simplest supervised learning techniques available. The naive Bayes classification algorithm is highly effective, reliable, and fast. Naive Bayes classifiers work quickly and accurately on large datasets.

3.4.3. GRADIENT BOOSTING

Ensemble modeling methods have gained popularity in recent years, and one such method is boosting, which combines multiple weak models to create a strong classifier. The process of boosting begins with the creation of a primary model using readily accessible training data sets. After locating errors in the base model, a secondary model is constructed, and the procedure continues by adding more models until we have a complete collection of training data that can accurately predict. Gradient Boosting Machines (GBM) is a popular algorithm used to boost weak learners into strong ones in the field of machine learning. This article on "GBM in Machine Learning" will cover topics such as boosting algorithms, gradient machine learning algorithms, the history of GBM, and different GBM terminologies. Before diving into GBM, it's essential to understand the boosting concept and different boosting algorithms used in machine learning.

3.5. DEPLOYMENT USING FLASK:

The model that is implemented utilizes Gradient Boosting, which is considered to be the most accurate algorithm among the three algorithms available. The deployment of this model is carried out using the Flask micro web framework.

4. ACCURACY COMPARISON

After comparing the accuracy of three different algorithms, it was determined that the Gradient Boosting algorithm outperformed the other two. Therefore, the implementation of the project utilized the Gradient Boosting algorithm to achieve the best accuracy possible.

4.1. RANDOM FOREST ALGORITHM

```
from sklearn.metrics import accuracy_score
print('Accuracy of Random forest Classifier is ',accuracy_score(y_test,predict)*100)

Accuracy of Random forest Classifier is  80.0
```

Fig 2. Screenshot of accuracy of Random forest algorithm

The image above displays the accuracy achieved in the model training process when using the Random Forest algorithm. It is found that 80.0% of the classifiers produced by the random forest model are accurate.

4.2.NAÏVE BAYES ALGORITHM

```
from sklearn.metrics import accuracy_score
print('Accuracy of Multinomial Naive Bayes is ',accuracy_score(y_test,predict)*100)

Accuracy of Multinomial Naive Bayes is  80.0
```

Fig 3. Screenshot of accuracy of Naïve Bayes Algorithm

isrjournals

International Journal of Advanced Research in Computer Science Engineering and Information Technology

Volume: 6, Issue: 3,Special Issue: 2 ,Apr,2023 ,ISSN_NO: 2321-3337

The above image demonstrates the accuracy obtained during model training using the Naive Bayes algorithm. The accuracy of the multinomial Naive Bayes model was determined to be 80.0%.

4.3. GRADIENT BOOSTING

```
from sklearn.metrics import accuracy_score
print('Accuracy of Gradient Boosting Classifier is ',accuracy_score(y_test,predict)*100

Accuracy of Gradient Boosting Classifier is  86.04651162790698
```

Fig 4. Screenshot of accuracy of Gradient Boosting

The accuracy achieved by using the Gradient Boosting algorithm to train the model is shown in the above image. The accuracy attained is 86.04%, which is the accuracy of the gradient-boosting classifier.

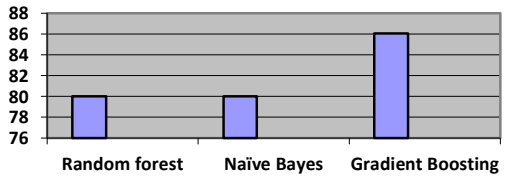

Fig 5.Accuracy Comparison

The graph above displays a comparison of three recommended working algorithms. Based on the data presented, the "Gradient Boosting Algorithm" delivers the highest accuracy when compared to the other two algorithms. Therefore, the Gradient Boosting Algorithm is selected and implemented in the project deployment for its superior accuracy.

Comparison of accuracies in proposed and related work:

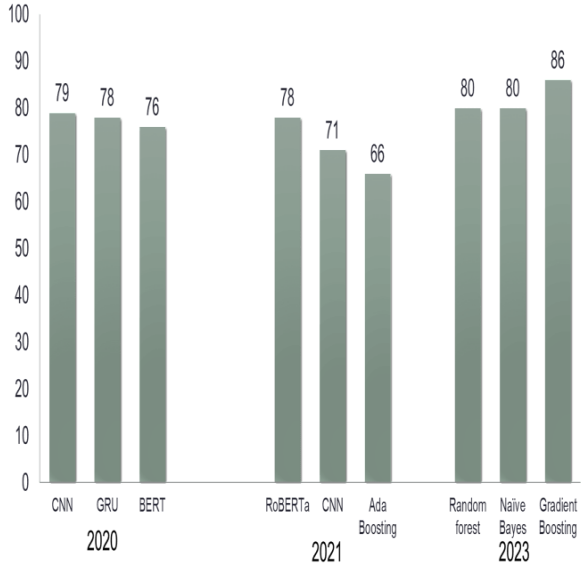

Fig 6. Comparison of accuracies in proposed and related work

According to the performance analysis presented above, the Gradient Boosting Algorithm produces the greatest accuracy. The accuracy of algorithms like CNN, GRU, and BERT in study papers from 2020 was 79%, 78%, and 76%, respectively. The accuracy of algorithms like RoBERTa, CNN, and Ada Boosting in study papers from 2021 was 78%, 71%, and 66%, respectively. This leads to the conclusion that the proposed work has produced better accuracy results than its related work.

5. CLASSIFICATION REPORT

A classification report is a performance assessment measure used in machine learning to present the accuracy, recall, F1 Score, and support of the trained classification model. The following are the classification reports for the three methods:

isrjournals

International Journal of Advanced Research in Computer Science Engineering and Information Technology

Volume: 6, Issue: 3,Special Issue: 2 ,Apr,2023 ,ISSN_NO: 2321-3337

5.1. RANDOM FOREST ALGORITHM

```
from sklearn.metrics import classification_report
print('Classification report of Random forest Classifier\n\n',classification_report(y_test,pre
```

```
Classification report of Random forest Classifier

                      precision    recall  f1-score   support

         Hate_Speech       0.71      0.62      0.67        16
No_Hate_and_Offensive       0.81      0.88      0.85        25
   Offensive_Language       0.84      0.84      0.84        19

            accuracy                           0.80        60
           macro avg       0.79      0.78      0.78        60
        weighted avg       0.80      0.80      0.80        60
```

Fig 7. Screenshot of classification report of Random forest algorithm

The figure presented above illustrates the values of precision, recall, f1-score, and support for the hate speech, no hate, and offensive language classes, obtained from the model trained using the Random Forest method.

5. 2. NAÏVE BAYES ALGORITHM

```
from sklearn.metrics import classification_report
print('Classification report of Multinomial Naive Bayes\n\n',classification_report(y_test,predict
```

```
Classification report of Multinomial Naive Bayes

                      precision    recall  f1-score   support

         Hate_Speech       0.72      0.76      0.74        41
No_Hate_and_Offensive       0.94      0.76      0.84        41
   Offensive_Language       0.77      0.89      0.83        38

            accuracy                           0.80       120
           macro avg       0.81      0.80      0.80       120
        weighted avg       0.81      0.80      0.80       120
```

Fig 8. Screenshot of classification report of Naïve Bayes algorithm

The figure above presents the accuracy, recall, f1-score, and support values for the Hate_Speech, No_Hate_and_Offensive, and Offensive Language categories, which were obtained by training the model with the Naive Bayes algorithm.

5.3. GRADIENT BOOSTING

```
from sklearn.metrics import classification_report
print('Classification report of Gradient Boosting Classifier\n\n',classification_report(y_test,predict))
```

```
Classification report of Gradient Boosting Classifier

                      precision    recall  f1-score   support

         Hate_Speech       0.73      0.80      0.76        10
No_Hate_and_Offensive       0.89      1.00      0.94        17
   Offensive_Language       0.92      0.75      0.83        16

            accuracy                           0.86        43
           macro avg       0.85      0.85      0.84        43
        weighted avg       0.87      0.86      0.86        43
```

Fig 9. Screenshot of classification report of Gradient Boosting algorithm

The above figure shows the precision, recall, f1-score, and support values for hate speech, no hate and offensive language that were obtained by training the model with the Gradient Boosting Algorithm.

6. CONFUSION MATRIX

6.1. RANDOM FOREST ALGORITHM

```
from sklearn.metrics import confusion_matrix
print('confusion matrix of Random forest Classifier\n\n',confusion_matrix(y_test,predict))
```

```
confusion matrix of Random forest Classifier

 [[2 0 0]
 [0 4 0]
 [2 0 2]]
```

Fig 10. Screenshot of confusion matrix of Random forest algorithm

The number of true positive values for hate speech is 2. The false negative for hate speech is 0 since none were missed by the model. The false positive for hate speech is 2 since 2 instances were classified as hate speech by the model but were actually not. The true negative value for hate speech is 6, meaning that 6 instances were correctly classified as not being hate speech by the model.

isrjournals

International Journal of Advanced Research in Computer Science Engineering and Information Technology

Volume: 6, Issue: 3,Special Issue: 2 ,Apr,2023 ,ISSN_NO: 2321-3337

6.2. NAÏVE BAYES ALGORITHM

```
from sklearn.metrics import confusion_matrix
print('confusion matrix of Multinomial Naive Bayes\n\n',confusion_matrix(y_test,predic

confusion matrix of Multinomial Naive Bayes

 [[31  2  8]
 [ 8 31  2]
 [ 4  0 34]]
```

Fig 11. Screenshot of classification report of Naïve Bayes algorithm

When assessing the accuracy of hate speech detection, various metrics are considered. Specifically, the True Positive value is 31, which indicates that 31 instances of hate speech were correctly identified. The False Negative value is 10, meaning that 10 instances of hate speech were not detected, consisting of 2 missed instances and 8 instances that were incorrectly classified as non-hate speech. Additionally, the False Positive value is 12, representing the number of non-hate speech instances that were mistakenly classified as hate speech. This value arises from 8 instances of non-hate speech being wrongly classified as hate speech and 4 true non-hate speech instances being classified as such. Finally, the True Negative value is 67, accounting for correctly identified non-hate speech instances (34) as well as missed hate speech instances (2) and true non-hate speech instances (31).

6.3. GRADIENT BOOSTING ALGORITHM

```
from sklearn.metrics import confusion_matrix
print('confusion matrix of Gradient Boosting Classifier\n\n',confusion_matrix(y_test,

confusion matrix of Gradient Boosting Classifier

 [[ 8  1  1]
 [ 0 17  0]
 [ 3  1 12]]
```

Fig 12. Screenshot of confusion matrix of Gradient Boosting algorithm

To assess the performance of hate speech detection, various metrics are used. Concerning hate speech, the True Positive value is 8, meaning that 8 instances of hate speech were correctly identified. Conversely, the False Negative value for hate speech is 2, indicating that two instances of hate speech were missed, including one instance that was misclassified as non-hate speech. Moreover, the False Positive value for hate speech is 3, implying that 3 instances of non-hate speech were erroneously identified as hate speech. This value is due to 3 instances of non-hate speech being incorrectly classified as hate speech, and none of the actual non-hate speech instances were correctly classified as such. Lastly, the True Negative value for hate speech is 30, which comprises accurately classified non-hate speech instances (12) and missed hate speech instances (1) as well as true non-hate speech instances (17).

7.RESULTS AND DISCUSSION

Fig 13 Represents the user interface where users enter comments for hate speech, offensive language, no hatred, and offensive classification.

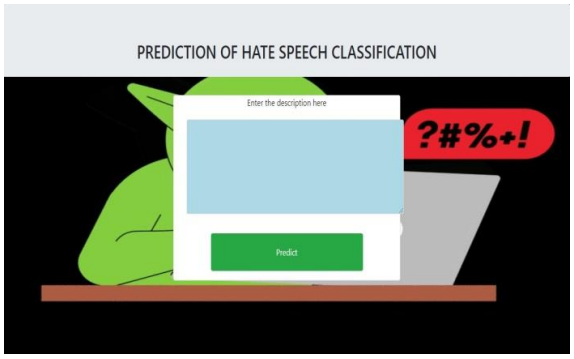

Fig. 13 Screenshot of output screen to classify the comments

In this screenshot the home page is displayed with the entered text and organized remarks.

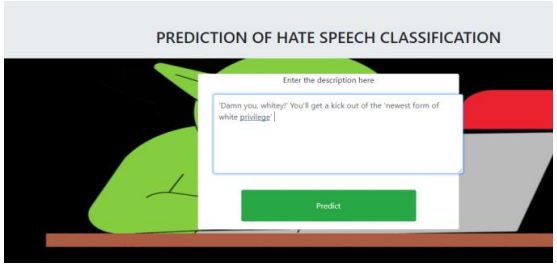

Fig 14. Screenshot of entering the text for classification

The image depicts the process of entering text into the designated textbox for comments. Once the text has been inputted, the user must click on the "Predict" button to initiate the process of analyzing the comment and revealing its classification.

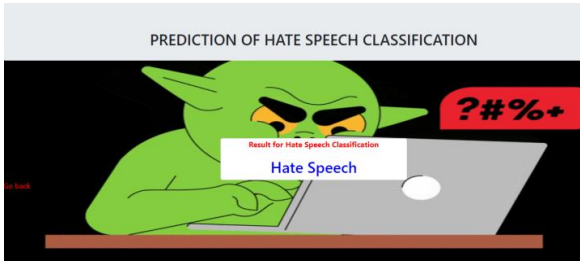

Fig 15. Screenshot of the prediction of Hate Speech

Upon clicking the "Predict" button following the entry of text, this image displays the resulting categorization and presentation of the comment on the webpage. The comment displayed in the figure is classified as hate speech.

8. CONCLUSION

In the era of digital technology, the internet has provided a platform for anyone to publish content, making it challenging to control hate speech. Consequently, there is a high risk of individuals being misled by such content. However, machine learning offers a solution to this complex problem as it takes less time to analyze large amounts of data. By leveraging historical data, machine learning algorithms can accurately identify hate speech and detect patterns, which can be used to refine the model's parameters for increased accuracy. Subsequently, the model can be utilized as a categorization tool by comparing various algorithms. Among the algorithms tested, Gradient Boosting was found to provide the highest accuracy, and thus it was used in the project implementation.

9.REFERTENCES

[1] Thomas Davidson,Dana Warmsle, Michael Macy,Ingmar Weber," Automated Hate Speech Detection and the Problem of Offensive Language" in Proceedings of the international AAAI conference on web and social media 11 (1), 512-515, 2017

[2] Hajime Watanabe, Mondher Bouazizi, Tomoaki Ohtsuki ," Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection"in IEEE access 6, 13825-13835, 2018

[3] Georgios Rizos, Konstantin Hemker, Bjorn Schuller," Augment to Prevent: Short-Text Data Augmentation in Deep Learning for Hate-Speech Classification" in the Proceedings of the 28th ACM international

conference on information and knowledge management, 991-1000, 2019

[4] Amrutha, B R and Bindu, K R, "Detecting Hate Speech in Tweets Using Different Deep Neural Network Architectures in 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 923-926, 2019

[5] Sindhu Abro1 , Sarang Shaikh2 , Zafar Ali ,**"** Automatic Hate Speech Detection using Machine Learning: A Comparative Study"in International Journal of Advanced Computer Science and Applications 11 (8), 2020

[6] Raghad Alshalan and Hend Al-Khalifa," A Deep Learning Approach for Automatic Hate Speech Detection in the Saudi Twittersphere"in Applied Sciences 10 (23), 8614, 2020

[7]Zewdie Mossie, Jenq-Haur Wang,"Vulnerable community identification using hate speech detection on social media" in Information Processing & Management 57 (3), 102087, 2020

[8] Ching Seh Wu, Unnathi Bhandary," Detection of Hate Speech in Videos Using Machine Learning" in 2020 International Conference on Computational Science and Computational Intelligence (CSCI), 585-590, 2020

[9] Ching Seh Wu, Unnathi Bhandary," Detection of Hate Speech in Videos Using Machine Learning" in 2020 International Conference on Computational Science and Computational Intelligence (CSCI), 585-590, 2020

[10] Varsha Pathak , Manish Joshi , Prasad A. Joshi , Monica Mundada , Tanmay Joshi," Using Machine Learning for Detection of Hate Speech and Offensive Code-Mixed Social Media text " in arXiv preprint arXiv:2102.09866, 2021

[11] Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, Animesh Mukherjee," HateXplain "A Benchmark Dataset for Explainable Hate Speech Detection" in Proceedings of the AAAI Conference on Artificial Intelligence 35 (17), 14867-14875, 2021

[12] Soumitra Ghosh , Asif Ekbal , Pushpak Bhattacharyya, Tista Saha, Alka Kumar, and Shikha Srivastava,"SEHC: A Benchmark Setup to Identify Online Hate Speech in English" in IEEE Transactions on Computational Social Systems, 2022

Volume: 6, Issue: 3,Special Issue: 2 ,Apr,2023 ,ISSN_NO: 2321-3337