

A Machine Learning Based Approach for Automated Identification of Abusive and Hateful Speech

Priyanka Goswami¹, Prof. Preetish Kshirsagar²

Abstract: Sentiment Analysis models are being used extensively as an important application of natural language processing. One such application is identification of hateful and radical speech on social media. With millions of posts and comments generated every day, relying solely on manual detection methods is impractical. Moreover, hate speech often evolves in subtle ways, making it difficult to track using predefined rules or simple keyword-based methods. This necessitates the development of automated systems capable of understanding and adapting to the nuanced and context-dependent nature of hate speech. The proposed approach combines TF-IDF and Neural Networks for identifying potential hate speech. The approach is based on the conjugate back propagation model, along with term and group frequencies as additional features. It has been shown that the proposed approach attains higher classification accuracy compared to existing work in the domain.

Keywords:- Natural Language Processing, Sentiment Analysis, Hate Speech, TF-IDF (Term Frequency–Inverse Document Frequency), Deep Learning, Classification Accuracy.

I. Introduction

Hate speech, defined as any communication that belittles or discriminates against individuals based on their race, religion, ethnicity, gender, sexual orientation, or other characteristics, presents a significant challenge for moderation [1]. In the age of social media, platforms like Twitter have become rich sources of real-time public opinion. Sentiment analysis of tweets enables businesses, governments, and researchers to gauge public mood, customer satisfaction, or political trends. However, due to the informal and concise nature of tweets, analyzing sentiment accurately requires robust natural language processing techniques

The scale of the problem is immense; millions of posts, comments, and messages are generated daily across

various platforms. Moreover, hate speech is often subtle and context-dependent, making it difficult for traditional keyword-based filters to detect effectively [2]. This complexity necessitates the use of advanced ML algorithms capable of understanding context, sentiment, and implicit meanings. Machine learning offers several advantages over traditional methods in identifying hate speech [3].

Firstly, ML models can be trained on vast datasets, enabling them to recognize patterns and nuances that human moderators might miss. These models can learn from context, differentiating between hate speech and benign content that might contain similar keywords. Secondly, ML systems operate at a scale and speed unattainable by human moderators, allowing for real-time monitoring and intervention. This rapid response is crucial in mitigating the spread of harmful content before it reaches a large audience. [4]



Fig.1 Occurrences of Hate Speech on Social Media

Figure 1 depicts the occurrence of potential hate speech. Moreover, hate speech is often subtle and context-dependent, making it difficult for traditional keyword-based filters to detect effectively. This complexity necessitates the use of advanced ML algorithms capable of understanding context, sentiment, and implicit meanings [5].

II. Existing Challenges

Despite its potential, implementing ML for hate speech detection is fraught with challenges. One major issue is the bias in training data, which can lead to models that unfairly target certain groups while overlooking others. Ensuring the diversity and representativeness of training datasets is essential to mitigate this risk. Additionally, the dynamic nature of language, including slang, memes, and coded language used to evade detection, requires continuous updating and retraining of ML models. Privacy concerns also arise, as the collection and analysis of user data must comply with regulations and respect individual rights [6].

While ML can significantly enhance the identification of hate speech, it is not a standalone solution. Ethical considerations, such as the risk of over-censorship and the potential for false positives, necessitate a hybrid approach where human moderators work alongside ML systems. Human oversight is crucial to review contentious cases and provide context that automated systems might lack. This collaborative approach helps balance the need for effective moderation with the protection of free speech and user rights [7].

III. Methodology

The proposed methodology employs the back propagation based neural networks for classification of potential hate speech. Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. Other advantages include [8]:

1. **Adaptive learning:** An ability to learn how to do tasks based on the data given for training or initial experience.
2. **Self-Organization:** An ANN can create its own organization or representation of the information it receives during learning time.
3. **Real Time Operation:** ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability [9].

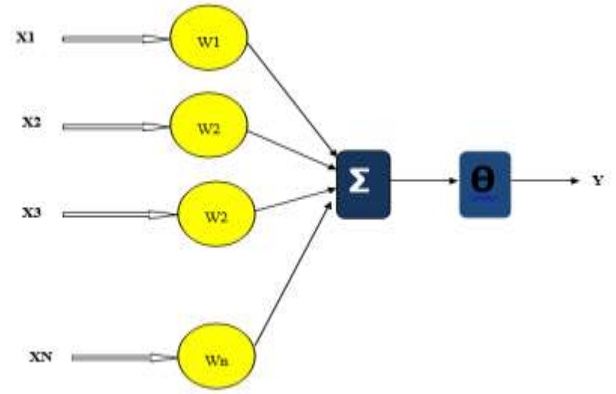


Fig.2 Mathematical Model of Neural Network

The output of the neural network is given by:

$$Y = f \sum_{i=1}^n X_i W_i + \Theta \quad (1)$$

Where,

X_i represents the signals arriving through various paths,

W_i represents the weight corresponding to the various paths and

Θ is the bias. It can be seen that various signals traversing different paths have been assigned names X and each path has been assigned a weight W . The signal traversing a particular path gets multiplied by a corresponding weight W and finally the overall summation of the signals multiplied by the corresponding path weights reaches the neuron which reacts to it according to the bias Θ . Finally its the bias that decides the activation function that is responsible for the decision taken upon by the neural network. The activation function φ is used to decide upon the final output. The learning capability of the ANN structure is based on the temporal learning capability governed by the relation [10]:

$$w(i) = f(i, e) \quad (2)$$

Here,

$w(i)$ represents the instantaneous weights

i is the iteration

e is the prediction error

The weight changes dynamically and is given by:

$$W_k \xrightarrow{e, i} W_{k+1} \quad (3)$$

Here,

W_k is the weight of the current iteration.

W_{k+1} is the weight of the subsequent iteration.

(i) Regression Learning Model

Regression learning has found several applications in supervised learning algorithms where the regression analysis among dependent and independent variables

is eeded [11]. Different regression models differ based on the the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used. Regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a relationship between x (input) and y(output). Mathematically [12],

$$y = \theta_1 + \theta_2 x \quad (4)$$

Here,

x representst the state vector of inut variables

y rpresenst the state vector of output variable or variables.

θ_1 and θ_2 are the co-efficients which try to fit the regression learning models output vector to the input vector.

By achieving the best-fit regression line, the model aims to predict y value such that the error difference between predicted value and true value is minimum. So, it is very important to update the θ_1 and θ_2 values, to reach the best value that minimize the error between predicted y value (pred) and true y value (y). The cost function J is mathematically defined as [13]:

$$J = \frac{1}{n} \sum_{i=1}^n (\text{pred}_i - y_i)^2 \quad (5)$$

Here,

n is the number of samples

y is the target

pred is the actual output.

(ii) Proposed Algorithm

The training algorithm for the proposed work is the Conjugate Gradient Back-propagation with Fletcher-Reeves Restarts. This algorithm is based on the concept of feeding back the errors to the neural network i.e. back propagation [14]. The salient feature of the algorithm is its relatively low time complexity and accuracy. The reason for the mentioned phenomena is the fact that the algorithm searches for the direction for the steepest direction right from the first iteration. Mathematically,

$$p_0 = -g_0 \quad (6)$$

Here,

p_0 is the negative of the gradient vector g_0

For the k^{th} iteration [15]:

$$p_k = -g_k + \theta_k p_{k-1} \quad (7)$$

It is worth noting that the in addition to the weights, the search vector also keeps updating with the iterations.

The term θ_k is calculated as:

$$\theta_k = \frac{g_k g_k^T}{g_{k-1} g_{k-1}^T} \quad (8)$$

The overall training rule for the algorithm can be mathematically expressed as [16]:

$$w_{k+1} = w_k + \beta_k p_k \quad (9)$$

The Conjugate Gradient Back-propagation algorithm, particularly with Fletcher-Reeves restarts, is a powerful optimization method used in machine learning and neural network training [17]. Combining the efficiency of the conjugate gradient method with restart mechanisms provides numerous advantages over traditional back-propagation and other gradient-based optimization techniques. These advantages contribute to improved convergence rates, computational efficiency, and robustness, making it a preferred choice in many machine learning tasks [18].

One of the primary advantages of the Conjugate Gradient Back-propagation with Fletcher-Reeves restarts is its ability to achieve faster convergence compared to standard gradient descentm[19] The conjugate gradient method uses information about previous search directions to construct conjugate directions, ensuring that successive iterations do not undo previous progress. This efficiency often results in fewer iterations to reach the optimal solution, which is particularly valuable for training deep neural networks with a large number of parameters [20]. The additional metric computed in this appraoch is the term and inverse-term frequency [21]. The term frequency measures how frequently a term t appears in a tweet d. The basic formula is [22]:

$$TF(t, d) = \frac{f_{t,d}}{\sum f_{k,d}} \quad (10)$$

Where:

$f_{t,d}$: Frequency of term t in tweet d

$\sum f_{k,d}$: Total number of terms in tweet d

This appraoch not only computes the term frequency but also the group frequency as the additional metric.

IV. Evaluation Parameters

Since errors can be both negative and positive in polarity, therefore its immaterial to consider errors with

signs which may lead to cancellation and hence inaccurate evaluation of errors. Therefore we consider mean square error and mean absolute percentage errors for evaluation. The system accuracy can be evaluated in terms of the mean square error which is mathematically defined as:

$$mse = \frac{1}{n} \sum_{i=1}^N (X - X')^2 \quad (11)$$

Here,

X is the predicted value and

X' is the actual value and n is the number of samples.

Further, the classification accuracy is computed as:

$$Ac = \frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

Here,

TN denotes true positive

TN denotes true negative

FP denotes false positive

FN denotes false negative

The results are presented in the next section.

IV. Results:

The simulations have been carried out on MATLAB on an PC with i5 processor and 16 GB of RAM. The major libraries used for the simulations are NNet and Deep Learning Toolboxes (Libraries).

The data has been collected from Kaggle.

Data Normalization: Canonization (normalization) of the text is the process of bringing to a single format, convenient for further processing. When working with large amount of information, it is necessary to exclude from the document all non-informative parts of speech (prepositions, particles, conjunctions, etc.). Subsequently both term and group frequencies are computed and fed to the neural network model for training. The results obtained are presented next:

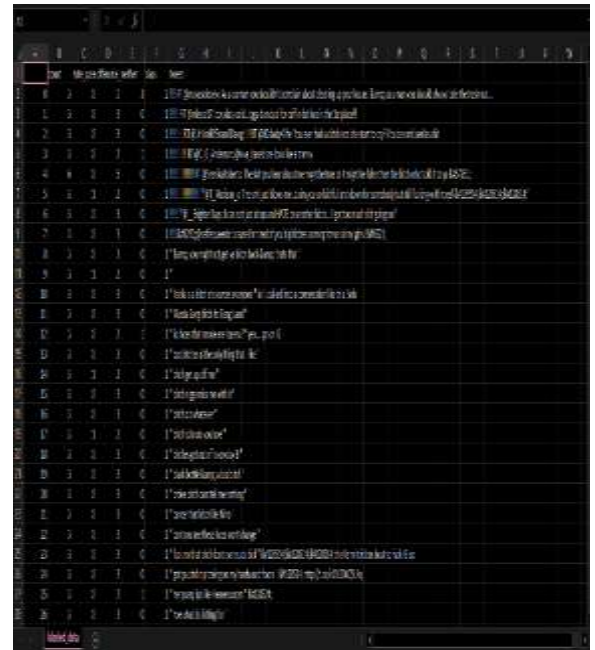


Fig.3 Raw Data

Figure 3 renders a sample screenshot of the raw data, which is imported as strings to the MATLAB workspace.

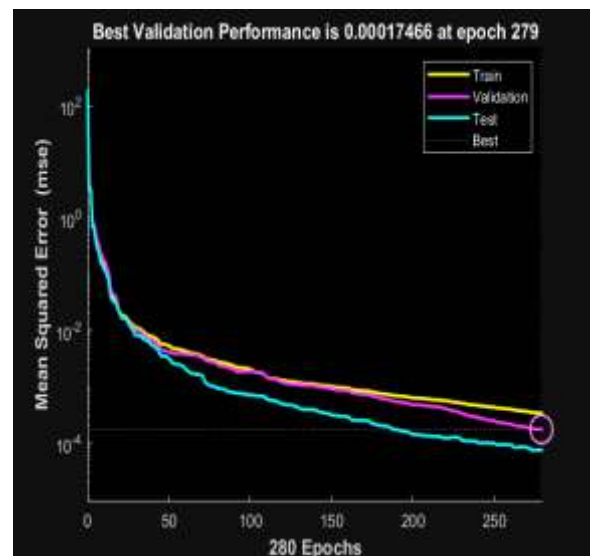


Fig.4 Variation of MSE

The variation of the mean squared error as a function of the number of epochs is shown in the above figure. It can be seen that the MSE after 279 iterations, after which the training is stopped.

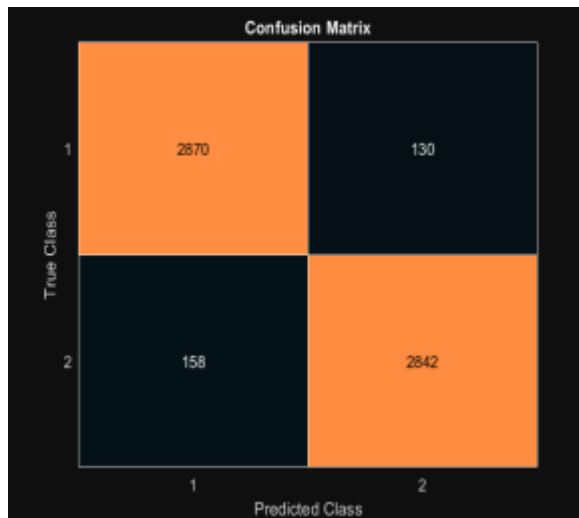


Fig.5 Confusion Matrix

Figure 5 depicts the confusion matrix for the proposed work. It can be observed that the proposed work attains a classification accuracy of around 95.2% which is significantly higher than that of previous work Bigoulaeva, et al. [23] which is around 78%.

VI. Conclusion:

Hate speech detection is a critical task in natural language processing (NLP) due to its implications for maintaining a safe online environment. Deep learning models have been widely employed for this task, leveraging their ability to learn complex patterns in textual data. Optimization techniques play a crucial role in enhancing the performance of these models. Among these techniques, the Conjugate Gradient Back-propagation algorithm with Fletcher-Reeves restarts stands out for its efficiency and robustness. The proposed approach combines both term and group frequencies as augmenting features with the data set to train a deep neural network model. The accuracy of the proposed model is shown to be significantly higher compared to the existing work in the domain.

References

1. J. Langham and K. Gosha, "The classification of aggressive dialogue in social media platforms," in Proc. ACM SIGMIS Conf. Comput. People Res., Jun. 2018, pp. 60–63.
2. P. Fortuna and S. Nunes, "A survey on automatic detection of hate speech in text," ACM Comput. Surv., vol. 51, no. 4, pp. 1–30, 2018.
3. W. Dorris, R. Hu, N. Vishwamitra, F. Luo, and M. Costello, "Towards automatic detection and explanation of hate speech and offensive language," in Proc. 6th Int. Workshop Secur. Privacy Anal., Mar. 2020, pp. 23–29.
4. Alrehili, "Automatic hate speech detection on social media: A brief survey" in Proc. IEEE/ACS 16th

Int. Conf. Comput. Syst. Appl. (AICCSA), Nov. 2019, pp. 1–6.

5. S. Modi, "AHTDT—Automatic hate text detection techniques in social media" in Proc. Int. Conf. Circuits Syst. Digit. Enterprise Technol. (ICCSDET), Dec. 2018, pp. 1–3.

6. F. E. Ayo, O. Folurunso, F. T. Ibharalu, and I. A. Osinuga, "Machine learning techniques for hate speech classification of Twitter data: State of the-art, future challenges and research directions" Comput. Sci. Rev., vol. 38, Nov. 2020, Art. no. 100311.

7. Schmidt and M. Wiegand, "A survey on hate speech detection using natural language processing" in Proc. 5th Int. Workshop Natural Lang. Process. Social Media, 2017, pp. 1–10

8. E. Shushkevich and J. Cardiff, "Automatic misogyny detection in social media: A survey," Computación Y Sistemas, vol. 23, no. 4, pp. 1159–1164, Dec. 2019.

9. F. Poletto, V. Basile, M. Sanguinetti, C. Bosco, and V. Patti, "Resources and benchmark corpora for hate speech detection: A systematic review," Lang. Resour. Eval., vol. 55, pp. 477–523, Jun. 2020.

10. R. Magu, K. Joshi, and J. Luo, "Detecting the hate code on social media," in Proc. Int. AAAI Conf. Web Social Media, 2017, vol. 11, no. 1, pp. 608–611.

11. I Bigoulaeva, V Hangya, I Gurevych, A Fraser, "Label modification and bootstrapping for zero-shot cross-lingual hate speech detection", Language Resources and Evaluation, Springer 2023, Art.no.1198.

12. S Akuma, T Lubem, IT Adom, "Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets", International Journal of Information Technology, Springer 2022, vol.14, pp. pp.3629–3635

13. S. Khan et al., "HCovBi-Caps: Hate Speech Detection Using Convolutional and Bi-Directional Gated Recurrent Unit With Capsule Network," in IEEE Access, 2022, vol. 10, pp. 7881–7894

14. C. -C. Wang, M. -Y. Day and C. -L. Wu, "Political Hate Speech Detection and Lexicon Building: A Study in Taiwan," in IEEE Access, 2022, vol. 10, pp. 44337–44346

15. N. S. Mullah and W. M. N. W. Zainon, "Advances in Machine Learning Algorithms for Hate Speech Detection in Social Media: A Review," in IEEE Access, 2021, vol. 9, pp. 88364–88376.

16. F. T. Boishakhi, P. C. Shill and M. G. R. Alam, "Multi-modal Hate Speech Detection using Machine Learning," 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 2021, pp. 4496–4499.

17. M. Z. Ali, Ehsan-Ul-Haq, S. Rauf, K. Javed and S. Hussain, "Improving Hate Speech Detection of Urdu Tweets Using Sentiment Analysis," in IEEE Access, 2021, vol. 9, pp. 84296–84305.

18. P Charitidis, S Doropoulos, S Vologiannidis, "Towards countering hate speech against journalists on

social media”, Online Social Networks and Media, Elsevier 2020, vol.17, 100071.

19. P. K. Roy, A. K. Tripathy, T. K. Das and X. -Z. Gao, "A Framework for Hate Speech Detection Using Deep Convolutional Neural Network," in IEEE Access, 2020, vol. 8, pp. 204951-204962.

20. Y. Jusman, M. A. Nur'Aini and S. Puspita, "Classification of Dental Caries Level Using Conjugate Gradient Backpropagation Models," 2023 International Seminar on Application for Technology of Information and Communication (iSemantic), Semarang, Indonesia, 2023, pp. 204-208.

21. N. S. Mohd Nafis and S. Awang, "An Enhanced Hybrid Feature Selection Technique Using Term Frequency-Inverse Document Frequency and Support Vector Machine-Recursive Feature Elimination for Sentiment Classification," in IEEE Access, vol. 9, pp. 52177-52192, 2021

22. H. Liu, X. Chen and X. Liu, "A Study of the Application of Weight Distributing Method Combining Sentiment Dictionary and TF-IDF for Text Sentiment Analysis," in IEEE Access, vol. 10, pp. 32280-32289, 2022.

23. I. Bigoulaeva, V. Hangya, I. Gurevych, and A. Fraser, "Label Modification and Bootstrapping for Zero-Shot Cross-Lingual Hate Speech Detection, Language Resources and Evaluation, Springer 2023, vol.57, pp.1515-1546.