Proceedings of the International Conference on Sustainable Computing and Data Communication Systems (ICSCDS-2022)
IEEE Xplore Part Number: CFP22AZ5-ART; ISBN: 978-1-6654-7884-7

Machine Learning based Automatic Hate Speech Recognition System

*P. William
Department of Information Technology
Sanjivani College of Engineering,
Savitribai Phule Pune University,
Pune, India
anil.pawar1983@gmail.com

Ritik Gade
Department of Computer Engineering
Sanjivani College of Engineering,
Savitribai Phule Pune University,
Pune, India
ritikgade9@gmail.com

Rupesh Chaudhari
Department of Computer Engineering
Sanjivani College of Engineering,
Savitribai Phule Pune University,
Pune, India
rupeshchaudhari2151@gmail.com

Dr. A. B. Pawar
Department of Computer Engineering
Sanjivani College of Engineering,
Savitribai Phule Pune University,
Pune, India
anil.pawar1983@gmail.com

Dr. M. A. Jawale
Department of Information Technology
Sanjivani College of Engineering,
Savitribai Phule Pune University,
Pune, India
jawale.madhu@gmail.com

2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS) | 978-1-6654-7884-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICSCDS53736.2022.9760959

***Abstract*—Social media and knowledge sharing have had a positive impact on humanity. However, this has also led to a number of issues, such as the dissemination and dissemination of hate speech. This new problem of hate speech on social media has been addressed by recent studies that utilized a number of feature engineering techniques and machine learning algorithms. It's not clear if there is a study that compares different methods for generating features and machine learning algorithms in order to determine which one is better for a standard publicly available dataset. With the support vector machine technique, the testing findings showed that bigram features performed best with 79 percent overall accuracy when utilized with the bigram feature set. Detecting automated hate speech messages can be made easier with the findings of our investigation. It will also be used as a benchmark for future research into existing automatic text classification algorithms, based on the results of the various comparisons. The use of natural language processing to classify text and hate speech are all examples of machine learning**

Keywords – Hate speech, natural language processing and machine learning

I. INTRODUCTION

Detecting hate speech is challenging due to the wide range of meanings. As a result, certain content may be judged hateful by some but not by others. [1] defines hate speech as follows:

Racial, religious, and ethnic profiling are instances of content that advocates violence against individuals or groups.

Despite the different criteria, some recent study claims that automated hate speech identification works [2-4]. To identify hate speech, the suggested approaches combined feature engineering and machine learning algorithms. There's still work to be done comparing the effectiveness of various strategies for identifying hate speech. Feature engineering approaches and ML algorithms are not compared in existing research [5].

What follows here is organised as follows: The connected works are highlighted in Section II. Discussed in Section III are the methods. Experimentation, results, and conclusions are discussed in detail in Sections IV, V, and VI of this paper. Sec. VII concludes with discussion of limitations, future research and conclusions.

II. RELATED WORKS

On social media, there is a lot of hate speech. Researchers have previously used a text classification strategy that relies on the use of supervised machine learning (ML). It has been reported that a variety of feature representation strategies have been used by different researchers. These include dictionary [6], bag/words [7], N/Ngrams, TFIDF, and deep-learning [8] techniques, as well as others.

A dictionary-based technique was used by Peter Burnap et al. to identify cyber hatred on Twitter. To construct the numeric vectors from a predetermined vocabulary of nasty phrases, they used an N-gram feature engineering technique. An F-score of 67% was attained using an SVM classifier fed with the resulting numeric vector. Automatic detection of racism in Dutch social media was also carried out using a dictionary-based technique by Stéphan Tulkens and colleagues [9]. The distribution of words among three dictionaries was employed as a characteristic in this analysis. The SVM classifier used the generated features. In their experiments, the F-Score was 0.46 out of 100. Hate speech in online forums and blogs was classified by Njagi Dennis et al. [10] using an ML-based classifier. It was decided to use a dictionary-based approach to construct the master feature vector by the authors. An emphasis on hate speech led to the use of sentiment expressions and semantic and subjective elements. A rule-based classifier was then fed the masters feature vector. The scientists used a precision performance metric to test their classifier in the experiment and achieved a 73 percent accuracy rate.

As a result, though, the use of both dictionary-based and machine learning methodologies resulted in an excellent outcome Such a technique, however, has a fundamental drawback: It relies on the vast corpus to hunt for domain words. There is a way around this problem that is quite similar to the dictionary-based technique, but uses training data instead of predetermined dictionaries to obtain the word characteristics.

The supervised ML technique was utilised by Edel Greevy et al. [11] to classify the racist material. Using a bigram feature extraction technique, the authors were able to turn the raw text into numerical vectors. BOW feature representation was employed in conjunction with bigram features by the authors in their study. The SVM was employed.

classifier to process the outcomes of the experiment. They achieved an accuracy rate of 87 percent in their findings. Using an ML-based technique, Irene Kwok and her colleagues were able to automatically detect racism against black people on Twitter. They used the BOW-based technique to produce the numeric vectors in their research. The Nave Bayes classifier was fed the resulting numerical vector by the authors. Their research yielded an accuracy rate of up to 76%. When it comes to classifying hate speech on Twitter, one group to look at is Sanjana Sharma and colleagues [12]. They used BOW characteristics in their research. This numeric vector was input to the Nave Bayes classifier by the authors of this paper. In their experiments, they found a maximum accuracy of 73%.

BOW, on the other hand, performed better in social network text classification than its competitors. On the other hand, the word order is ignored, which leads to erroneous classifications because different terms are employed in various situations. Researchers have developed an N-grams-based strategy to circumvent this constraint. The produced numeric vector was fed into the LR classifier, and an overall F-score of 73% was obtained. The ML-based strategy was utilised by Chikashi Nobata et al. to identify offensive words in online user content. The SVM classifier was fed the features by the authors. The overall F-score for the classifier was 77 percent. Using an ML-based technique, Shervin Malmasi and colleagues [13] were able to identify hate speech on social media. The authors provided the SVM classifier the derived numeric characteristics. The authors estimated an accuracy of up to 78%.

Until recently, only few researchers used ML to detect hate speech on the internet on autopilot. Sensitive themes, for example, have been categorised by Karthik Dinakar and his colleagues [14]. The numeric feature vectors were generated using the TFIDF feature representation methodology and the unigram method. In their research, they made use of TFIDF to express trigram characteristics. They employed the Naive Bayes classifier for classification. The Nave Bayes classifier performed best in their experiments, with a 68 percent accuracy rate.

There are two key drawbacks to the N-gram technique, which outperforms the BOW approach. Increasing the N value slows processing because the linked words may be far apart in the text [15].

Authors have recently used NLP algorithms based on deep learning to classify hate speech. Also included were the many forms of offensive language and the people it was directed at. Deep learning models with various feature sets were used by the authors to get the greatest F1 score of 0.74.

Natural language processing (NLP) was used to perform a survey on hate speech identification in 2017. A variety of feature engineering strategies for supervised categorization of hate speech messages were covered in depth by the authors of the paper. The survey's main shortcoming is that no experimental findings were provided for the approaches suggested [16 -17].

Hate speech identification in languages including German, Dutch, and English has been the subject of previous investigations by scholars from around the world. A comparative examination of multiple characteristics and machine learning techniques on the standard dataset, however, has not been conducted to our knowledge, which could serve as a benchmark for future academics working on hate speech recognition. this paper.

III. METHODOLOGY

To classify tweets into "hate speech, offensive but not hate speech, and neither hate speech nor offensive speech," we've created this section to clarify our suggested system. Figure 1 depicts the entire research procedure. The study approach illustrated in this figure includes six important steps: data collecting, data pre-processing, feature engineering, data splitting, classification model creation, and classification model assessment. We'll go through each step in great detail in the next sections.

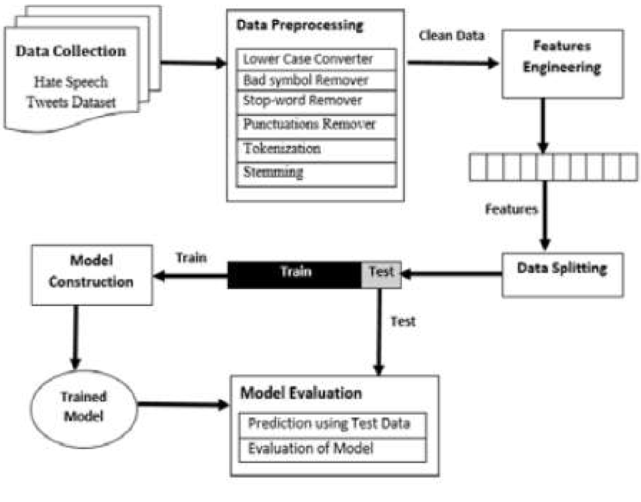


Fig. 1. Overview of the System

A. *Obtaining Information*

We used a publicly available collection of hate speech tweets for this investigation. CrowdFlower compiled and annotated this dataset [18-20]. As a result, the dataset has three unique groups of tweets that are categorised as either offensive or hate speech. There are 14509 tweets in this dataset. 16 percent of the tweets fall under this category. Furthermore, 50% of tweets are not inflammatory, while the remaining 33% are offensive but do not fall under the hate speech category. This distribution is depicted in Fig. 2 as well.

B. Pre-processing of the text

Pre-processing the text before classifying it improves classification outcomes, according to a number of studies. As a result, in our dataset, we used a variety of pre-processing-techniques to remove erroneous and irrelevant information from the tweets. We converted the tweets to lowercase during pre-processing [21-23].

C. Designing and Creating Feature-Loaded Software

raw text cannot be used by machine learning (ML) algorithms to decipher classifiers. The categorization principles of these algorithms can only be understood numerically. As a result, feature engineering is a critical stage in text classification.

D. Diffusion of Data

There is a whole dataset's class distribution, once data splitting is completed (i.e., Training set and Test set). We divided the pre-processed data in half using an 80-20 ratio to ensure equal distribution (i.e., 80 percent for Training Data and 20 percent for Test Data). Training data is utilised to train the classification model [24-25]. A second evaluation of the classification model is carried out using the test data. Models for Machine Learning, E.

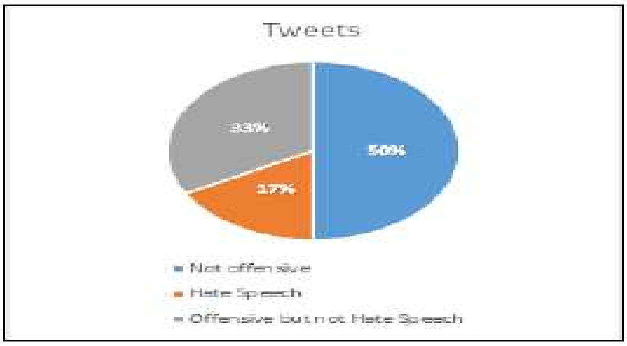


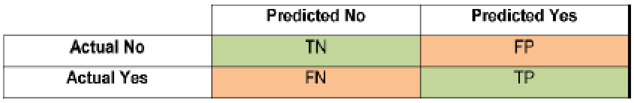
Fig. 2. Data Splitting

TABLE 1: Data Splitting Details

	Class	Total Instances	Training instances	Testing instances
0	Hate Speech	2399	1909	490
1	Not offensive	7274	5815	1459
2	Offensive but not Hate Speech	4836	3883	953
	Total	**14509**	**1607**	**2902**

E. Experimentation with Classifiers

Predicting unlabelled text's class using a test set is a stage in which the classifier we've built makes predictions about the text's true class (i.e. True negatives (TN), false positives (FP), false negatives (FN), and true positives (TP) are used to evaluate the classifier's performance (TP). Figure 3 shows a confusion matrix with these four integers [26-28]. The classifier's performance is evaluated using a variety of parameters. The following are some of the most used performance metrics in text categorization.



	Predicted No	Predicted Yes
Actual No	TN	FP
Actual Yes	FN	TP

Fig. 3. Confusion Matrix

F. Experimental Settings

Specifically, we employed n-gram (bigram) with TFIDF, Word2vec and Doc2vec as features. In the end, there are three alternative ways to depict a feature's master form [29]. In addition, the three master feature vectors were subjected to an array of eight distinct machine learning methods. As a result, a total of 24 analyses were conducted to evaluate the performance of categorization models [30-31].

IV. RESULTS AND DISCUSSIONS

This section summarises the 24 analyses' findings. The bolded numbers represent the maximum and minimum. In the lab, multiple feature representation and classification techniques were tested. Bigram TFIDF features outranked Word2vec and Doc2vec. There was some overlap between bigram and Doc2vec. SVM outperformed the other seven text classifiers. SVM beat AdaBoost and RF classifiers in accuracy but not speed.

Confusion matrices of the best research SVM classifier bigram and TFIDF confusion matrix Only 155 out of 490 "hate speech" tweets were successfully identified. But 335 were mislabelled. Of the 335 incidents, 54 were deemed harmless, while 281 were deemed offensive but not hateful. There are 1459 instances, however only 1427 tweets are abusive. These were mislabelled as hate speech in five cases and objectionable language in 27 situations. There were 953 problematic words out of 2902, but none were hated speech. 698 tweets were judged as offensive but not as hate speech by SVM. 122 and 133 were mislabelled as hate words over 120 times.

Adaboost's confusion matrix employs bigram and TFIDF. Adaboost's classifier beats SVM using bigram and TFIDF. However, the Adaboost did not excel in hate speech.

V. CONCLUSION

The methods used in this study to detect hate speech texts are automated text categorization systems. Experiment results revealed that when the bigram features were expressed using TFIDF, they performed significantly better than when the features were generated using the word2Vec and Doc2Vec techniques. The KNN algorithm had the lowest performance. The findings of this research study will be of practical significance. It has scientific significance because it provides experimental data in the form of more than one scientific measurement that may be used for automatic text categorization. First and foremost, the approaches based on lexicons will be investigated and evaluated in comparison to other existing state-of-the-art outputs. Second, a greater number of data examples will be acquired, which will be used to more effectively learn the categorization rules.

REFERENCES

[1] Kumar R, Ojha AK, Malmasi S, Zampieri M. Benchmarking Aggression Identification in Social Media. In: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018). ACL; 2018. p. 1–11.

[2] CodaLab—Competition;. Available from: https://competitions.codalab.org/competitions/19935.

[3] Detecting Insults in Social Commentary;. Available from: https://kaggle.com/c/detecting-insults-insocial- commentary.

[4] Neuman Y, Assaf D, Cohen Y, Last M, Argamon S, Howard N, et al. Metaphor Identification in Large Texts Corpora. PLoS ONE. 2013; 8(4). https://doi.org/10.1371/journal.pone.0062343

[5] P. William and A. Badholia, "Analysis of Personality Traits from Text Based Answers using HEXACO Model," 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), 2021, pp. 1-10, doi: 10.1109/ICSES52305.2021.9633794.

[6] P William, Dr. Abhishek Badholia 2021. Assessment of Personality from Interview Answers using Machine Learning Approach. International Journal of Advanced Science and Technology. 29, 08 (Jul. 2021), 6301-6312.

[7] P William, Dr. Abhishek Badholia (2020) Evaluating Efficacy of Classification Algorithms on Personality Prediction Dataset. Elementary Education Online, 19 (4), 3400-3413. doi:10.17051/ilkonline.2020.04.764728

[8] P. William, Dr. Abhishek Badholia."A Review on Prediction of Personality Traits Considering Interview Answers with Personality Models", Volume 9, Issue V, International Journal for Research in Applied Science and Engineering Technology (IJRASET) Page No: 1611-1616, ISSN : 2321-9653.

[9] P. William, P. Kumar, G. S. Chhabra and K. Vengatesan, "Task Allocation in Distributed Agile Software Development using Machine Learning Approach," 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), 2021, pp. 168-172, doi: 10.1109/CENTCON52345.2021.9688114.

[10] P William and Vaishali Sanjay Patil. “Architectural Challenges of Cloud Computing and Its Security Issues with Solutions" International Journal for Scientific Research and Development 4.8 (2016): 265-268

[11] Salton G, Yang CS, Wong A. A Vector-Space Model for Automatic Indexing. Communications of the ACM. 1975; 18(11):613–620. https://doi.org/10.1145/361219.361220

[12] Grossman DA, Frieder O. Information Retrieval: Algorithms and Heuristics. Berlin, Heidelberg: Springer-Verlag; 2004.

[13] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed Representations of Words and Phrases and their Compositionality. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ, editors. Advances in Neural Information Processing Systems 26. Curran Associates, Inc.; 2013. p. 3111–3119.

[14] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:181004805 [cs]. 2018;.

[15] Yang Z, Chen W, Wang F, Xu B. Unsupervised Neural Machine Translation with Weight Sharing. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics; 2018. p. 46–55. Available from: http://aclweb. org/anthology/P18-1005.

[16] Kuncoro A, Dyer C, Hale J, Yogatama D, Clark S, Blunsom P. LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modeling Structure Makes Them Better. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics; 2018. p. 1426–1436. Available from: http://aclweb.org/ anthology/P18-1132.

[17] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. JMLR. 2011; 12:2825–2830.

[18] Kim Y. Convolutional Neural Networks for Sentence Classification. In: EMNLP; 2014.

[19] Hagen M, Potthast M, Bu¨chner M, Stein B. Webis: An Ensemble for Twitter Sentiment Detection. In: SemEval@NAACL-HLT; 2015.

[20] Joulin A, Grave E, Bojanowski P, Mikolov T. Bag of Tricks for Efficient Text Classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. ACL; 2017. p. 427–431.

[21] Zhang Z, Robinson D, Tepper J. Detecting hate speech on twitter using a convolution-gru based deep neural network. In: European Semantic Web Conference. Springer; 2018. p. 745–760.

[22] Zhao J, Xie X, Xu X, Sun S. Multi-view learning overview: Recent progress and new challenges. Information Fusion. 2017.

[23] Greevy, E. and A.F. Smeaton. Classifying racist texts using a support vector machine. in Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. 2004. ACM.

[24] Kwok, I. and Y. Wang. Locate the hate: Detecting tweets against blacks. in Twenty-seventh AAAI conference on artificial intelligence. 2013.

[25] Sharma, S., S. Agrawal, and M. Shrivastava, Degree based classification of harmful speech using twitter data. arXiv preprint arXiv:1806.04197, 2018.

[26] Malmasi, S. and M. Zampieri, Detecting hate speech in social media. arXiv preprint arXiv:1712.06427, 2017.

[27] Nobata, C., et al. Abusive language detection in online user content. in Proceedings of the 25th international conference on world wide web. 2016. International World Wide Web Conferences Steering Committee.

[28] Waseem, Z. and D. Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. in Proceedings of the NAACL student research workshop. 2016.

[29] Dinakar, K., R. Reichart, and H. Lieberman. Modeling the detection of textual cyberbullying. in fifth international AAAI conference on weblogs and social media. 2011.

[30] Pandian, A. Pasumpon. “Performance Evaluation and Comparison using Deep Learning Techniques in Sentiment Analysis.” Journal of Soft Computing Paradigm (JSCP) 3, no. 02 (2021): 123-134.

[31] Kottursamy, Kottilingam. “A review on finding efficient approach to detect customer emotion analysis using deep learning analysis.” Journal of Trends in Computer Science and Smart Technology 3, no. 2 (2021): 95-113.