# Study on Optimizing Feature Selection in Hate speech using Evolutionary Algorithms

Harsh Mittal[1], Kartikeya Singh Chauhan[2], Prashant Giridhar Shambharkar[3]

[1]Research Scholar, Department of Engineering Physics

[2]Research Scholar, Department of Electronics and Communications,

[3]Assistant Professor, Department of Computer Science and Engineering

[1,2,3]Delhi Technological University, Bawana Road, New Delhi, Delhi-110042

24mithar@gmail.com, ksc13dec@gmail.com,
prashant.shambharkar@dtu.ac.in

**Abstract.** Hate speech is an important problem while dealing with user-generated content on online social media platforms. The huge amount of data generated makes it nearly impossible to manually moderate hate speech content and take appropriate measures. In this paper, we utilize various optimisation algorithms to enhance the feature extraction and vectorization, of various techniques like TF-IDF, Word2Vec, and Bag of Words and appertain on the machine learning models for two-fold classification. We gauge and visualize the conclusion of the propounded methodology of the hate speech problem about Twitter tweets. We examine our suggested technique on three datasets; out of which two of the datasets were highly unbalanced and SMOTE was used for class balance. Our experiments indicate the random behavior of Particle Swarm Optimization and Genetic Algorithm, and the decrease in accuracy when applied individually to the experiments. The results also indicate that the accuracy can be achieved back by applying Particle Swarm Optimization and Genetic Algorithm parallels, countering their random behavior.

**Keywords:** Particle Swarm Optimization · Genetic Algorithm · Natural Language Processing · Machine Learning · Hate · Cyberbullying

## 1. Introduction

The invention of social media is a global revolution. It is an inexpensive mode of communication that attracts the masses in a very short period. Social media has the advantage of the global reach of access. But along with this social media brings with it an important challenge. Mass or global access to these platforms gives a chance for people to post content that is offensive or hateful towards another community or a group of people. Hate speech is described by many dictionaries as; speech addressed to the people, demonstrates hate or embolden acts of violence against an individual or coterie based on arguments such as caste creed, culture, gender, or sexual orientation. The laws of some nations define hate speech as speech, gesticulations, demeanor, or exhibits that encourage vehemence or detrimental influence against individuals or community based on their belonging in the group, or which belittles or scares and terrorizes a group of individuals based on their belonging in the group. In some nations, hate speech has not yet been added as a legal term whereas, in other countries, most of it is safeguards the citizens constitutionally. In other nations, the sufferer of hate speech may seek remedy under criminal, civil or both laws [1]. Hate speech in general on social media can cause a lot of harm to any victims who are targeted to fabricate a sense of exclusion amongst their communities, to toxify public opinion and coax other forms of radical and detestable behavior. Accordingly, there is a need for an automated mechanism to distinguish hate speech at an exhaustive scale, thereby enabling analyses of large textual datasets collected from social media [2–4]. We can battle the problem of Hate Speech and cyberbully by promoting awareness of the problem, sensitizing the people about the problem, and reporting hate ourselves when we find it online. The best way of fighting the problem is by educating society in general and creating an environment in a community where such deeds are considered inappropriate. Everyone has to play their part in attempting to make society think about the concomitant of comments they make, and this might deter them from making hateful or abusive comments in the first place [5]. Hate speech is widespread in social media. A glance through the comments section of any social media platform demonstrates how prevalent the penetrating problem is [6, 7]. Social media platforms provide an economical communication medium that gives everyone access to millions of users. Consequently, in these platforms, any kind of content can be published and accessed by anyone, representing a groundbreaking revolution in our society [8]. Safety and security in social media have become a concerning topic and thus research work in this field has grown substantially in the last decade. The most impactful aspect of this work is detecting and averting the use of various forms of hate speech [9]. Across all social media platforms, recreation and innovation are continuously transformed by humans. In this heed, social media companies play a major role, mainly the US and Chinese firms which have grown into universal giants [10]. The availability of a large elucidated troupe of social media content and the elaboration of powerful classification vantage points, such as the algorithms we have used in our research have contributed in an unparalleled way to contrivance the challenge of examining users' statements, comments and sentiment online social platforms across time [11].

The rest of the paper is organized as follows: Section 2 gives a brief overview of similar research done in the field of hate speech detection and classification. Section 3 gives a detailed explanation of the methodology used, dataset collection and how Particle Swarm Optimization and Genetic Algorithm were implemented on these datasets. Section 4 gives a tabular representation of the result of deploying the aforementioned algorithms on the datasets. A graphical description of the same has also been given. Section 5 gives the limitations of our experimental work. Finally, section 6 gives the conclusion and further scope for work in this field.

## 2. Related Work

A review of analogous work was done to collect, analyses and create a solution for hate speech on social media. Joshua Uyheng et al. made a dynamic lattice frame to identify hate coterie, emphasizing Twitter tweets kindred to the COVID epidemic in the United States of America and the Philippines. The data was collected using Twitter's Rest API. In the end, the dataset contained 15 million Twitter tweets, containing 1 million users from the Philippines, and 12 Million conversations containing 1.6 Million users. The Random Forest Classifier was exercised to enforce decipherability and scalability. The Net mapper software, selected from each conversation multiple lexical considerations of the use of pronouns, derogatory words, exclusive terms, absolutist words, and specific names among others. Pair-wise products of every linguistic measure were used as additional features to extract the ways they occur simultaneously within the Twitter tweets. A study by Araque et al [12] presented two novel feature extraction algorithms, which use previous Sentic computing resources, affective Space and Sentic Net. It also gave a machine learning frame using an ensemble of two different features to enhance the overall classification performance. Classifiers like SVM (Support Vector Machine), LR (Logistic Regression), deep learning-based CNN (Convolutional Neural Network), and BERT (Bidirectional Encoder Representations from Transformers) pre-trained models were proposed by Google. The weighted F1 Score was 0.64 and 0.62 on Trac Facebook English and Hindi datasets, while on Twitter English and Twitter Hindi, the weighted F1 score was 0.58 and 0.50 respectively. The dataset was collected in code-Hindi and English languages from Facebook and Twitter user comments of celebrity posts. Modha et al [13] suggested BERT model pre-trained with fast Text word embedding, the model marginally outperformed the BOW and TF-IDF (Term Frequency Invert Document Frequency) feature extraction techniques. Safa Alsafari et al [14] used feature extraction techniques like random embedding where every word is converted into a feature vector having a dimension of 300. The vectors are arbitrarily initialized and their training is a part of the model training. The study in [14] also includes a pre-trained word embedding, Fast Text, which is trained on Arabic Wikipedia corpus by Facebook. Kristian Miok et al. [15] used the BERT model along with Monte Carlo Dropout Method and provided a reliable estimation in the transformer network. Tomer Wullach et al. [16] used TF-IDF as word embedding and they also used GANs (Generative Adversarial Network) to scale up the dataset. Joni Salminen et al. [17] proposed an online hate detection by using multi-platform data, where 197566 comments from YouTube, Wikipedia, Reddit and Twitter was taken, with 80% of the data docketed as not hate and the left 20% of the comments were docketed as hateful. Then experiments with several classification algorithms like SVM (Support Vector Machines), Logistic Regression, XGBoost, Naive Bayes and neural networks were performed. The accentuate delineation has been done using BOW, TF-IDF, Word2Vec, Bert and their combinations. All the models surpass the keyword-based baseline classifier, XGBoost, plying all foregrounds with an F1 score of 0.92. BERT features were the most impactful. Here, multi-platform training of data has been done, as the uni-platform emphasis is taxing because there is no guarantor that the models developed discern well across other platforms. This helps to overcome the problem of reinventing the wheel. The model performs decently across numerous platforms, using BERT (Bidirectional Encoder Representation from Transformer).

## 3. Methodology

In our analysis, we sunder our experimentation into the subsequent parts: Dataset Collection, Pre-processing, Optimization and classification.

### (a) Data set Collection

The data was collected from various public-access hate speech datasets, which are fundraised and annotated by multiple individuals, which are:

- **New Combined Dataset**, which is formed by merging T.Davidson [18] and Personal Attack dataset, makes it a highly balanced and reliable dataset [19]. The New Combined dataset has a binary output and classifies the tweets as either hate or not hate. It had 37521 hate and 45045 not hate entries.
- **TweetBLM** [20], is an open-source dataset, which contains tweets relating to Black Lives Matter. It had 3048 hate and 6081 not hate entries.
- **Kaggle Dataset** [21], is an open-source dataset, available on Kaggle, which contains Twitter tweets, and has binary classification. It had 2242 hate and 29720 not hate entries.

### (b) Pre-processing

Our data consisted of raw tweets, and they had to be pre-processed so that they could be fed to the machine learning model for classification. The ensuing task was taken to pre-process the data: Removal of URLs, Emoji, Twitter Handles, Hyperlinks, Hashtags, stop words, Numbers, Converting the tweets into Lower Case, Lemmatization of words.

### (c) Optimization

For optimization of the feature selection process, two algorithms were deployed, Particle Swarm Optimization and Genetic Algorithm. We also tried out the combination of the two algorithms when executing one after the other.

Following is the description of the two algorithms which we have used to optimize the feature selection process, namely Particle Swarm Optimization and Genetic Algorithm.

1. **PSO (Particle Swarm Optimization)**

PSO is a meta-heuristic optimization algorithm. The foundation of this powerful optimization algorithm lies in artificial life and swarming theory. Its confederation can be traced to genetic algorithms and stochastic processes such as evolutionary programming. The algorithm is computationally inexpensive and involves the use of simple mathematical operators [22]. It looks for its inspiration from the swarm behaviour observed in nature such as a swarm of birds or a shoal of fish. The algorithm seeks to simulate and efficiently implement the aforementioned behaviour of these processes in nature. A bird's observation range is limited. But a swarm of birds enables them to be aware of a larger area. This is an example of communal behaviour and information transmission which the PSO aims to implement [23]. A velocity vector is delineated to each particle in the swarm to update its current position. These particles have memory and they have been given the ability to update their memory on their own decision. These particles then update their position based upon the collective behaviour of the population that is the swarm. The nature of this process is stochastic. The memory of each particle and the knowledge gained by the swarm is used in the next iterations.

$$x^i{}_{k+1} = x^i{}_k + v^i{}_{k+1}\Delta t \qquad (1)$$

Here, $x^i{}_{k+1}$ denotes the position of particle $i$ at iteration $k+1$. $v^i{}_{k+1}$ is the corresponding velocity vector of a particle. $\Delta$t is the unit time step function used in basic PSO A common scheme for updating the velocity vector of each particle introduced by Shi and Eberhart, [26] is

$$V^i{}_{k+1} = \omega v^i{}_{k+1} + c_1 r_1\,(p^i\text{-}x^i{}_k)/\,\Delta\, t + c_2 r_2\,(p^g{}_k\text{-}x^i{}_k)/\,\Delta\, t \quad (2)$$

Here $r^1$ and $r^2$ are random independent numbers between 0 and 1. $P^i$ is the best position found by particle $i$ so far, whereas $p^g{}_k$ is the best position in the swarm at time $k$. $\Delta$ $t$is the unit time step function. $\omega$ denotes the inertia of each particle. This variable is responsible for controlling the exploration properties of the PSO algorithm. Large values of $\omega$ mean that the algorithm will show a more global behaviour whereas a small value of $\omega$ means that the algorithm's behaviour will be more local. Parameters $c^1$ indicates how much confidence a particle has in itself, whereas $c^2$ denotes how much confidence the particle has in the swarm.
Initialization consists of creating a swarm by choosing its size and a random distribution of the particles throughout the design space. Each particle is then given a random velocity. Each iteration would then update the position of each particle and a distinction would be made between a particle's memory and the memory of the best particle in the neighbourhood.

$$x^i{}_o = x_{min} + r_3(x_{max}\text{-}x_{min}) \qquad (3)$$

$$v_o{}^i = [x_{min} + r_4(x_{max}\text{-}x_{min})]/\,\Delta\, t \qquad (4)$$

In equations (3) and (4), $v_o{}^i$ is the initial velocity vector of a particle; $x^i{}_o$ is the initial position vector of the particle. $r_3$ and $r_4$ are independent random numbers between 0 and 1, $x_{min}$ is the vector of lower bounds and $x_{max}$ is the vector of upper bounds [24]. The basic PSO algorithm was designed for unconstrained problems only. When it is used for optimization on the training set, it performs miserably. It has its demerits in the form that it may converge prematurely and its random behaviour is not suited when implemented on the training set.

## 2. GA (Genetic Algorithm)

GA is a heuristic adaptive search algorithm that belongs to a large category of evolutionary algorithms. It has been inspired by the theories revolving around evolution, natural selection and genetics. "Survival of the fittest" also known as the natural selection which has already been mentioned above is the main ideology behind this algorithm. These algorithms are used to obtain accurate solutions for search and optimization problems through random search backed by previous data experience for better efficiency and performance in solution space. Genetic Algorithms are generally regarded as function optimizers.

- The implementation of the genetic algorithm starts with a population of individuals analogous to a chromosome (represented as an integer/float/ string) which represent a point in the search space and a possible solution. These chromosomes are made up of several variable components known as genes.
- The selection of individuals (parents) is done for mating by the application of crossover and mutation operators on them. This generates new offspring (chromosomes).
- The problem function after a proper assessment is then given generative scope in such a way that the chromosomes which give a superior result are given more chances to "reproduce" as opposed to chromosomes representing inferior solutions.
- This is done by allocating each individual a fitness score. A higher score would imply a better solution and hence a higher probability of "reproduction".
- Since the population is static, the search space will be replaced by newer generations of individuals by causing the death of older individuals.
- This will consequently create new generations which represent a better partial solution set. Each new generation would have better partial solutions than the previous one.
- The aim is to converge the algorithm to a set of optimal solutions for the problem function. This happens when there is no significant difference in the solution set between multiple generations.

The advantages of the Genetic algorithm will include its robustness as they do not break on a slight change in input. Moreover, they also optimize over a large space state. Constrained optimization problems can be troublesome for the genetic algorithm as crossover and mutation operators might give infeasible solutions. Many mechanisms can be employed to overcome this problem but the discussion of those methods is beyond the scope of our research [25].

### 3. Particle Swarm Optimization + Genetic Algorithm

Lastly, Particle Swarm Optimization and Genetic Algorithm were deployed in series, to counter the randomness, which was earlier supported by the convex loss function. Firstly, Particle Swarm Optimization was used and the modified training set was then moved down to the Genetic Algorithm, which modified the training set again, which were finally passed to various classifiers for binary prediction.

### 4. Classification

For classification, various machine learning techniques like Logistic Regression, Random Forest classifiers and Naive Bayes were used, to produce a binary prediction of hate and not hate.

## 4. Implementation and Results

We have performed various experiments on baseline classifiers and experimented with different feature engineering techniques on the given datasets.

- In step 1, we deployed machine learning models, Logistic Regression, Random Forest Classifier, Naive Bayes, were deployed on feature extraction techniques of TF-IDF, BOW, Word2Vec, on three different datasets, New Combined Datasets, Tweet BLM and Kaggle Dataset.

- In step 2, we deployed machine learning models, Logistic Regression, Random Forest Classifier, Naive Bayes, on feature extraction techniques of TF-IDF, BOW, Word2Vec, which were optimized with Particle Swarm Optimization, on three different datasets: New Combined Datasets, Tweet BLM and Kaggle Dataset.

- In step 3, we deployed machine learning models, Logistic Regression, Random Forest Classifier, Naive Bayes, feature extraction techniques of TF-IDF, BOW, Word2Vec, which were optimized with Genetic Algorithm, on three different datasets, New Combined Datasets, Tweet BLM and Kaggle Dataset.
- In step 4, we deployed machine learning models, Logistic Regression, Random Forest Classifier, Naive Bayes, feature extraction techniques of TF-IDF, BOW, Word2Vec, which were optimized with Particle Swarm Optimization, followed by Genetic Algorithm, on three different datasets, New Combined Datasets, Tweet BLM and Kaggle Dataset.

In table 1, we have presented all the results which were conducted on the new combined dataset because the New Combined data set is the most authentic and balanced data set as no sampling technique has been utilized.

**Table 1:** Experimental Results

| Technique | Features | Model | Accuracy | Recall | Precision | F1 Score |
|---|---|---|---|---|---|---|
| **Baseline Models** | Bag of Words | Logistic regression | 0.93097 | 0.9568 | 0.90989 | 0.93276 |
| | | Naive Bayes | 0.77896 | 0.60856 | 0.92361 | 0.73369 |
| | | Random Forest | 0.93531 | 0.95476 | 0.91909 | 0.93658 |
| | TF-IDF | Logistic regression | 0.91792 | 0.9351 | 0.90415 | 0.91936 |
| | | Naive Bayes | 0.8645 | 0.83709 | 0.88582 | 0.86076 |
| | | Random Forest | 0.95891 | 0.97809 | 0.94201 | 0.95971 |
| | Word2Vec | Logistic regression | 0.86664 | 0.84422 | 0.88403 | 0.86366 |
| | | Naive Bayes | 0.80715 | 0.762 | 0.83789 | 0.79814 |
| | | Random Forest | 0.92802 | 0.8705 | 0.98377 | 0.92368 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **PSO** | Bag of Words | Logistic regression | 0.88484 | 0.9351 | 0.84981 | 0.89042 |
| | | Naive Bayes | 0.69413 | 0.66184 | 0.70786 | 0.68408 |
| | | Random Forest | 0.90441 | 0.95293 | 0.86875 | 0.9089 |
| | TF-IDF | Logistic regression | 0.85313 | 0.89924 | 0.82347 | 0.85969 |
| | | Naive Bayes | 0.79104 | 0.8971 | 0.74029 | 0.81118 |
| | | Random Forest | 0.91583 | 0.92084 | 0.91182 | 0.91631 |
| | Word2Vec | Logistic regression | 0.85711 | 0.85922 | 0.85441 | 0.85681 |
| | | Naive Bayes | 0.79823 | 0.80668 | 0.78482 | 0.7956 |
| | | Random Forest | 0.93449 | 0.98411 | 0.88334 | 0.93101 |
| **GA** | Bag of Words | Logistic regression | 0.88989 | 0.95273 | 0.84647 | 0.89646 |
| | | Naive Bayes | 0.63132 | 0.32909 | 0.83312 | 0.47181 |
| | | Random Forest | 0.90834 | 0.96556 | 0.86651 | 0.91336 |
| | TF-IDF | Logistic regression | 0.84926 | 0.90066 | 0.81685 | 0.85671 |
| | | Naive Bayes | 0.80929 | 0.746 | 0.85438 | 0.79652 |
| | | Random Forest | 0.89636 | 0.97066 | 0.84519 | 0.90359 |
| | Word2Vec | Logistic regression | 0.83715 | 0.91245 | 0.78935 | 0.84645 |
| | | Naive Bayes | 0.80518 | 0.76607 | 0.82535 | 0.79461 |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
|  |  | Random Forest | 0.70693 | 0.40698 | 0.99332 | 0.57739 |
| **PSO + GA** | Bag of Words | Logistic regression | 0.93327 | 0.96679 | 0.90613 | 0.93548 |
|  |  | Naive Bayes | 0.78099 | 0.61864 | 0.91653 | 0.73869 |
|  |  | Random Forest | 0.94561 | 0.97239 | 0.92302 | 0.94706 |
|  | TF-IDF | Logistic regression | 0.91777 | 0.94254 | 0.89816 | 0.91981 |
|  |  | Naive Bayes | 0.86868 | 0.85135 | 0.88209 | 0.86645 |
|  |  | Random Forest | 0.95917 | 0.97759 | 0.9429 | 0.95993 |
|  | Word2Vec | Logistic regression | 0.87113 | 0.90698 | 0.84298 | 0.87381 |
|  |  | Naive Bayes | 0.8072 | 0.75718 | 0.83547 | 0.7944 |
|  |  | Random Forest | 0.79374 | 0.58755 | 0.98849 | 0.73702 |

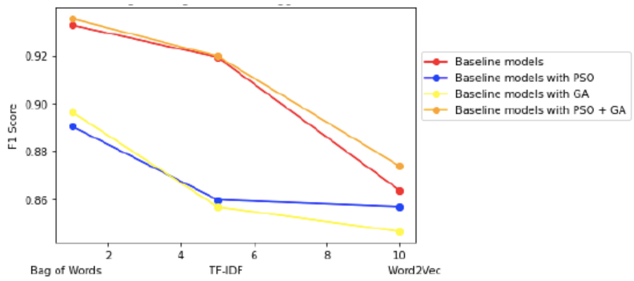


Figure 1: Logistic Regression on Kaggle Dataset

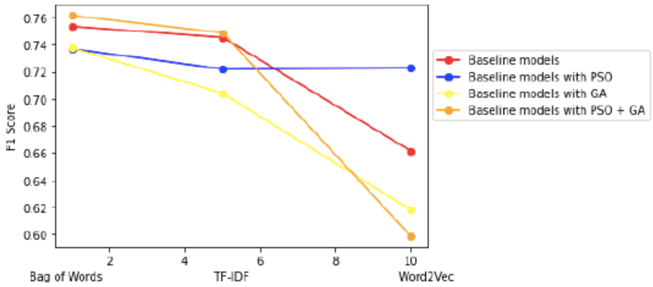


Figure 2: Logistic Regression on Tweet BLM.

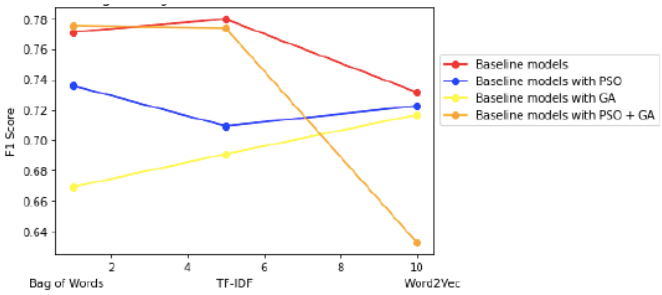
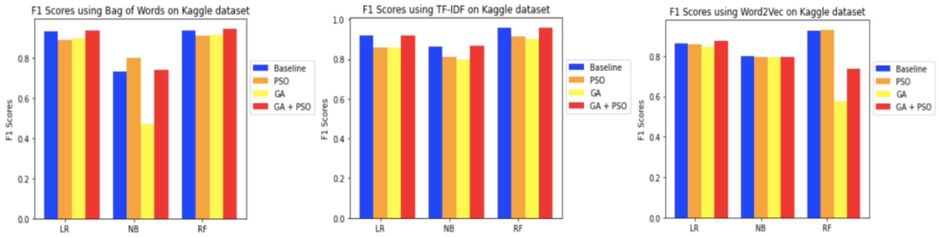
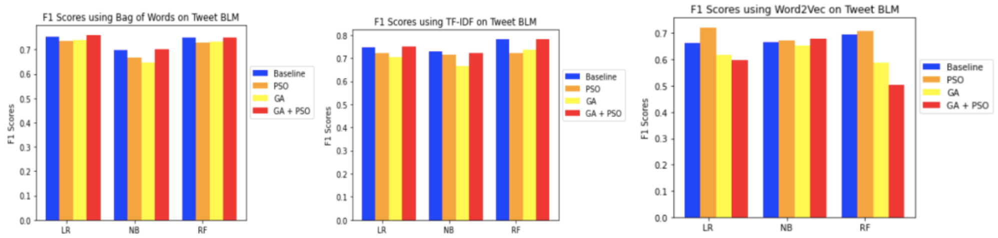


Figure 3: Logistic Regression on New Combined Dataset



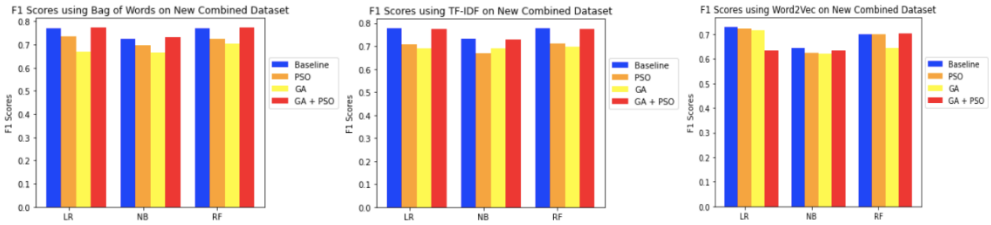
(a) (b) (c)

Figure 4: F1 Scores on Kaggle Dataset using (a) Bag of Words (b) TF-ID (c) Word2Vec



(a) (b) (c)

Figure 5: F1 scores on Tweet BLM dataset, (a) Bag of Words (b) TF-IDF (c) Word2Vec



(a) (b) (c)

Figure 6: F1 scores on New Combined Dataset (a) Bag of Words (b) TF-IDF (c)Word2Vec

After experimenting with various feature extraction techniques, TF-IDF, Word2Vec and Bag-of-Words, we concluded that TF-IDF outperforms all other feature selection techniques. Out of all datasets used, the New Combined data set is the most authentic and balanced data set as no sampling technique has been utilized. F1 Score achieved with new combined dataset was 77.5688 per cent using Random Forest with Particle Swarm Optimization and Genetic Algorithms on TF-IDF the F1 score for tweetBLM using Random Forest on Particle Swarm Optimization + Genetic Algorithm on TF-IDF was 78.3448. For the other two datasets, Kaggle data set, and TweetBLM, Smote was used for handling class imbalance. The best results are achieved while using TF-IDF on Random Forest Classifier on the Kaggle Data set. Overall, there is no effect of utilizing Particle Swarm Optimization and Genetic algorithms on TF-IDF, and when Particle Swarm Optimization and Genetic algorithms are applied, they indicate the random behavior of Particle Swarm Optimization and Genetic Algorithm, as performing the same experiment multiple times produces different results.

## 5. Limitations

Our experimental work so far has been restrained to predicting a binary output. As we are acquainted with the fact that hate speech can be broadly broken down into the further subcategories of sexism, abuse, racism, hate and spam. The experimentation can be further expanded to a multi-classification problem to examine how competently the technique propounded can classify within those subcategories of hate. Additionally, the task of classification is very subjective and based on personal bias, because the decision on whether some text is termed as hate or not, also depends a lot on the standpoint of the annotator, the culture and customs of a nation as well as the maturity and experience of the annotator. This upheaves the questions concerning the legitimacy and ability of the annotator to precisely identify the user-generated content. This affects the accuracy and precision of our machine learning model and is, therefore, a shortcoming. On the top of that, the sarcastic comments are difficult to discern from hate comments. Since we do not explicitly handle sarcastic tweets, some of the false positives generated by our model that were falsely classified as hate were sarcastic tweets, without the intention to hurt or oppose a community. Other than that, many of these tweets that have the hate label, those which were wrongly classified are those which did not necessarily contain any indecent or abusive term, which has a strong opinion against a particular section of the society. Another major issue in this experimentation can be attributed to the way SMOTE works, which definitely is questionable because either reduction of dataset, or multiple accounting of same entries.

## 6. Conclusion and Future Scope

Combining text with offensive, sarcastic or innocent words with the strong sentiment prompts the Cyber Bullying and hate speech detection algorithms to infelicitously flag the tweets and posts. Not rectifying this issue may imply significant dissentious effects such as users' defamations. To enhance the efficiency and precision of a classification mechanism and generalize it to newer datasets, we suggest more emphasis on deep learning-based techniques and improved architectures, rather than working on enhancing feature extraction, using the genetic and evolutionary algorithms because these algorithms are firstly highly random in nature, and uses convex loss function, which makes the algorithm even more irrelevant in this context. Thus, working on GANs and LSTM might help in solving the problem.

## References

1. Hate speech - Wikipedia, https://en.wikipedia.org/wiki/Hate_speech.
2. Vidgen, B., Yasseri, T.: Detecting weak and strong Islamophobic hate speech on social media. J. Inf. Technol. Polit. 17, 66–78 (2020). https://doi.org/10.1080/19331681.2019.1702607
3. Guiora, A., Park, E.A.: Hate Speech on Social Media. Philos. (United States). 45, 957–971 (2017). https://doi.org/10.1007/s11406-017-9858-4
4. Mathew, B., Dutt, R., Goyal, P., Mukherjee, A.: Spread of Hate Speech in Online Social Media. WebSci 2019 - Proc. 11th ACM Conf. Web Sci. 173–182 (2019). https://doi.org/10.1145/3292522.3326034

5. Uyheng, J., Carley, K.M.: Characterizing network dynamics of online hate communities around the COVID-19 pandemic. Appl. Netw. Sci. 6, (2021). https://doi.org/10.1007/s41109-021-00362-x
6. Ring, C.: Hate Speech IN Social Media: An Exploration of The Problem And Its Proposed Solutions. J. Chem. Inf. Model. 53, 1689–1699 (2013)
7. Schofield, A., Davidson, T.: Identifying hate speech in social media. XRDS Crossroads, ACM Mag. Students. 24, 56–59 (2017). https://doi.org/10.1145/3155212
8. Mondal, M., Silva, L.A., Benevenuto, F.: A measurement study of hate speech in social media. HT 2017 - Proc. 28th ACM Conf. Hypertext Soc. Media. 85–94 (2017). https://doi.org/10.1145/3078714.3078723
9. Malmasi, S., Zampieri, M.: Detecting hate speech in social media. Int. Conf. Recent Adv. Nat. Lang. Process. RANLP. 2017-Septe, 467–472 (2017). https://doi.org/10.26615/978-954-452-049-6-062
10. Matamoros-Fernández, A., Farkas, J.: Racism, Hate Speech, and Social Media: A Systematic Review and Critique. Telev. New Media. 22, 205–224 (2021). https://doi.org/10.1177/1527476420982230
11. Florio, K., Basile, V., Polignano, M., Basile, P., Patti, V.: Time of your hate: The challenge of time in hate speech detection on social media. Appl. Sci. 10, (2020). https://doi.org/10.3390/APP10124180
12. Araque, O., Iglesias, C.A.: An Ensemble Method for Radicalization and Hate Speech Detection Online Empowered by Sentic Computing. Cognit. Comput. (2021). https://doi.org/10.1007/s12559-021-09845-6
13. Modha, S., Majumder, P., Mandl, T., Mandalia, C.: Detecting and visualizing hate speech in social media: A cyber Watchdog for surveillance. Expert Syst. Appl. 161, 113725 (2020). https://doi.org/10.1016/j.eswa.2020.113725
14. Alsafari, S., Sadaoui, S., Mouhoub, M.: Hate and offensive speech detection on Arabic social media. Online Soc. Networks Media. 19, 100096 (2020). https://doi.org/10.1016/j.osnem.2020.100096
15. Miok, K., Škrlj, B., Zaharie, D., Robnik-Šikonja, M.: To BAN or Not to BAN: Bayesian Attention Networks for Reliable Hate Speech Detection. Cognit. Comput. (2021). https://doi.org/10.1007/s12559-021-09826-9
16. Wullach, T., Adler, A., Minkov, E.: Towards Hate Speech Detection at Large via Deep Generative Modeling. IEEE Internet Comput. 25, 48–57 (2021). https://doi.org/10.1109/MIC.2020.3033161
17. Salminen, J., Hopf, M., Chowdhury, S.A., Jung, S. gyo, Almerekhi, H., Jansen, B.J.: Developing an online hate classifier for multiple social media platforms. Human-centric Comput. Inf. Sci. 10, 1–34 (2020). https://doi.org/10.1186/s13673-019-0205-6
18. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated Hate Speech Detection and the Problem of Offensive Language *.
19. Charitidis, P., Doropoulos, S., Vologiannidis, S., Papastergiou, I., Karakeva, S.: Towards countering hate speech against journalists on social media. Online Soc. Networks Media. 17, 1–15 (2020). https://doi.org/10.1016/j.osnem.2020.100071
20. Pranesh, R.R., Kumar, S., Shekhar, A.: TweetBLM: A Hate Speech Dataset and Analysis of BlackLivesMatter-related Microblogs on Twitter. (2020). https://doi.org/10.5281/ZENODO.4000539
21. Hate Speech and Offensive Language Dataset | Kaggle, https://www.kaggle.com/mrmorj/hate-speech-and-offensive-language-dataset
22. Okwu, M.O., Tartibu, L.K.: Particle Swarm Optimisation. Stud. Comput. Intell. 927, 5–13 (2021). https://doi.org/10.1007/978-3-030-61111-8_2
23. Marini, F., Walczak, B.: Particle swarm optimization (PSO). A tutorial. Chemom. Intell. Lab. Syst. 149, 153–165 (2015). https://doi.org/10.1016/j.chemolab.2015.08.020
24. Hassan, R., Cohanim, B., De Weck, O., Venter, G.: A comparison of particle swarm optimization and the genetic algorithm. Collect. Tech. Pap. - AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf. 2, 1138–1150 (2005). https://doi.org/10.2514/6.2005-1897
25. Whitley, D.: A genetic algorithm tutorial. Stat. Comput. 4, 65–85 (1994). https://doi.org/10.1007/BF001753
26. Shi, Y., and Eberhart, R. C., “A Modified Particle Swarm Optimizer,” Proceedings of the

International Conference on Evolutionary Computation, Inst. of Electrical and Electronics Engineers,Piscataway,NJ,1998,pp.69–73.