Hate Speech Detection on Twitter: A Comparative Evaluation of Different Machine Learning Techniques

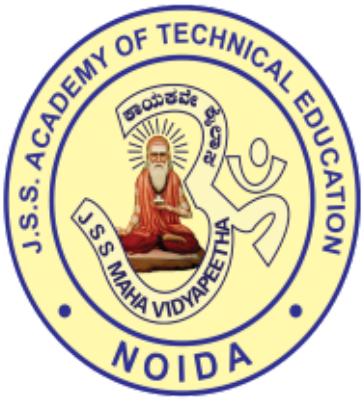
by

Aryan Rastogi (1900910100045)

Daarshik Dwivedi (1900910100058)

Arjit Kumar (1900910100043)

Abhishek Pratap Singh (1900910100008)



Under the Supervision of

Ms. Suruchi Saberwal

Department of Computer Science and Engineering

JSS Academy of Technical Education

Sector 62, Noida, Uttar Pradesh, 201301

May 2023

Hate Speech Detection on Twitter: A Comparative Evaluation of Different Machine Learning Techniques

by

Aryan Rastogi (1900910100045)

Daarshik Dwivedi (1900910100058)

Arjit Kumar (1900910100043)

Abhishek Pratap Singh (1900910100008)

Submitted to the Department of Computer Science and Engineering

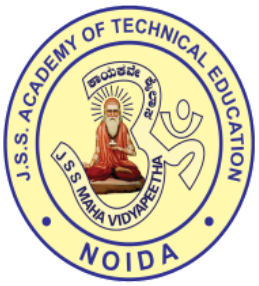
in partial fulfillment of the requirements

for the degree of

Bachelor of Technology

in

Computer Science and Engineering



JSS Academy of Technical Education

Sector 62, Noida, Uttar Pradesh, 201301

Dr. A.P.J Abdul Kalam Technical University, Uttar Pradesh, Lucknow.

May 2023

VISION OF THE DEPARTMENT:

To spark the imagination of the Computer Science Engineers with values, skills and creativity to solve the real world problems.

MISSION OF THE DEPARTMENT:

Mission 1: To inculcate creative thinking and problem solving skills through effective teaching, learning and research.

Mission 2: To empower professionals with core competency in the field of Computer Science and Engineering.

Mission 3: To foster independent and life-long learning with ethical and social responsibilities.

PROGRAMME EDUCATIONAL OBJECTIVES:

PEO1: To apply computational skills necessary to analyze, formulate and solve engineering problems.

PEO2: To establish as entrepreneurs, and work in interdisciplinary research and development organizations as an individual or in a team.

PEO3: To inculcate ethical values and leadership qualities in students to have a successful career.

PEO4: To develop analytical thinking that helps them to comprehend and solve real-world problems and inherit the attitude of lifelong learning for pursuing higher education.

PROGRAMME OUTCOMES:

Engineering Graduates will be able to:

PO1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

PROGRAMME SPECIFIC OUTCOMES:

PSO1: Acquiring in-depth knowledge of theoretical foundations and issues in Computer Science to induce learning abilities for developing computational skills.

PSO2: Ability to analyse, design, develop, test and manage complex software system and applications using advanced tools and techniques.

COURSE OUTCOMES

C 410.1	Identify, formulate, design and analyse a research-based/web-based problem to address societal and environmental issues.
C 410.2	Communicate effectively in verbal and written form.
C 410.3	Apply appropriate computing, engineering principle and management skills for obtaining effective solution to the formulated problem within a stipulated time.
C 410.4	Work effectively as a part of team in multi-disciplinary areas.
C 410.5	Consolidate the final outcome in the form of a publication.

CO-PO-PSO MAPPING

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
C 410.1	3	3	3	3	2	3	3	3	3	3	2	3	3	3
C 410.2	2	2	2	2	2	2	2	2	2	3	2	3	2	3
C 410.3	3	3	3	3	3	3	2	3	3	3	3	3	3	3
C 410.4	3	3	3	3	2	3	2	3	3	3	3	3	3	3
C410.5	3	3	3	3	3	3	3	3	3	3	3	3	3	3
C410	**2.80**	**2.80**	**2.80**	**2.80**	**2.40**	**2.80**	**2.40**	**2.80**	**2.80**	**3.00**	**2.60**	**3.00**	**2.80**	**3.00**

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature:

Name : Aryan Rastogi

Roll No.: 1900910100045

Signature:

Name : Daarshik Dwivedi

Roll No.: 1900910100058

Signature:

Name : Arjit Kumar

Roll No.: 1900910100043

Signature:

Name : Abhishek Pratap Singh

Roll No.: 1900910100008

Date :

CERTIFICATE

This is to certify that Project Report entitled "……..... .....” which is submitted by ……..... in partial fulfillment of the requirement for the award of degree B. Tech. in Department of ..... of Dr. APJ Abdul Kalam Technical University, Uttar Pradesh, Lucknow is a record of the candidate's own work carried out by him/her under my/our supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Supervisor: Mrs. Suruchi Saberwal

Date

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Professor Suruchi Sabrewal, Department of Computer Science & Engineering, JSS Academy of Technical Education, Noida for her constant support and guidance throughout the course of our work. Her sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only her cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Professor Mayank Singh, Head, Department of Computer Science & Engineering, JSS Academy of Technical Education, Noida for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature:

Name : Aryan Rastogi

Roll No.: 1900910100045

Signature:

Name : Daarshik Dwivedi

Roll No.: 1900910100058

Signature:

Name : Arjit Kumar

Roll No.: 1900910100043

Signature:

Name : Abhishek Pratap Singh

Roll No.: 1900910100008

ABSTRACT

The necessity for robust and fast detection techniques has become critical as social media hate speech has grown. This study investigates ways to identify hate speech comments present on Twitter using language processing methods. In this work, we suggest a cutting-edge method for effectively identify hate speech in tweets that combines linguistic elements and machine learning techniques. Using a sizable dataset of annotated tweets, we test our model, and we get good F1-score and accuracy. The results of this study provide ways to identify hate speech on Twitter using natural language processing techniques. They can also help shape the development of effective laws and other measures to decrease the detrimental effects of hateful speech across social networking platforms.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

1. NLP - Natural Language Processing
2. ANN - Artificial Neural Network
3. SVM - Support Vector Machine
4. RF - Random Forest
5. DT - Decision Tree
6. LR - Logistic Regression
7. GBDT - Gradient Boosting Decision Trees
8. MLP - Multi-Layer Perceptron
9. CNN - Convolutional Neural Network
10. RNN - Recurrent Neural Network
11. LSTM - Long Short-Term Memory
12. PCA - Principal Component Analysis
13. ROC - Receiver Operating Characteristic
14. AUC - Area Under the Curve
15. MAE - Mean Absolute Error
16. MSE - Mean Squared Error
17. RMSE - Root Mean Squared Error
18. F1 - F1 Score
19. SGD - Stochastic Gradient Descent
20. BOW - Bag-of-Words
21. TF-IDF - Term Frequency-Inverse Document Frequency
22. GPU - Graphics Processing Unit

CHAPTER 1

INTRODUCTION

As we witness the rapid growth of the online ecosystem, we can observe that a large amount of user data is produced every second in the form of images, videos, text posts. This data is commonly generated from social media platforms. The large amount of data generated also includes hate speech between different groups within and across countries to spread prejudices and disputes. Several social media platforms have measures to counter this problem.Twitter's rules state that users cannot use tweets to threaten or harass people as a result of their , gender, race, religion, or any other characteristic. Along with content that is blocked by gender, class, and handicap, YouTube screens information that incites hatred or hostility against specific individuals or groups.

Social media channels like Twitter and Reddit now represent the most popular mediums for the dissemination of hate speech due to their vast user base and ease of use. Twitter and Reddit have 280 million daily active users combined. The anonymity provided by social media can also encourage others to take part in hate speech, as they feel that they can express their views without fear of being made answerable for their deeds. As a result, hateful speech has increased on social media sites, posing a serious problem for online communities.

With such a wide reach of social media channels, the effects of hate spreading speech can be far-reaching, as it can lead to psychological harm, social isolation, and even physical violence. Additionally, it may foster a climate of fear and intimidation that discourages people from expressing their opinions and taking part in public debates. For disadvantaged populations that may already be marginalised and subject to discrimination, the effects of hate speech can be extremely severe.

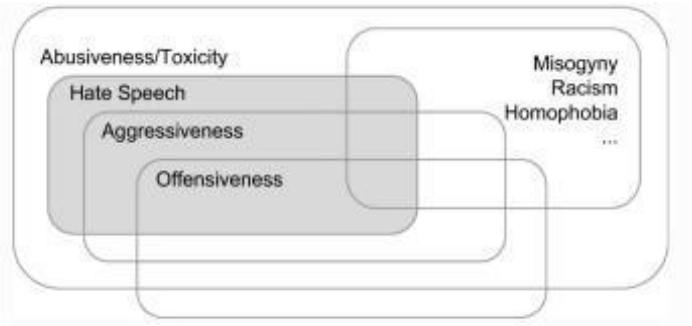


Figure 1: Relationships between abusive language phenomena (source: Poletto et al., 2020).

To combat the rise of hateful speech, many researchers have conducted studies to understand the nature and extent of this problem. Machine learning-based automated hate speech identification systems have attracted increasing attention in recent years. These systems have the potential to identify hate speech and provide early warnings to online communities, thereby reducing the spread of hateful speech. According to Jurgens et al. (2019), abusive language phenomena span a broad range and include things like microaggression, stereotyping, offence, abuse, hate speech, threats, and doxxing.The present approaches have mostly focused on a select group of problems, namely hate speech, abusive language, and offensive language. The relationships between these occurrences have only been tangentially explained, leaving us with a disjointed picture, a range of definitions, and incompatible annotations (Waseem et al., 2017). According to the definitions in earlier work Poletto et al. (2020) offer a graphical visualisation (Figure 1) of the links among abusive language.

This study emphasises the critical need for efficient hate speech identification and mitigation on social networks. We want to contribute to the creation of more potent methods for preventing hate speech online by putting forth a unique approach to hate speech identification that combines machine learning algorithms with natural language processing techniques. The findings of this study may influence the creation of automated technologies for identifying hate speech as well as the formulation of regulations and initiatives aimed at fostering more civil and welcoming online communities.

The research paper is organized logically and simply to effectively convey the goals and conclusions of the investigation. A literature review that gives the context and rationale for the investigation is included after the introduction, which introduces the research challenge and motivation for the study. The data sources, preprocessing methods, and models utilized in the experiment are described in the methodology section. The methodology section of the article gives a thorough breakdown of the whole strategy adopted. The study's results are presented in the results section, which is followed by a discussion part that interprets the findings and sheds light on their relevance. The conclusion then emphasizes the research's contributions, summarises the main findings, and offers suggestions for future research. A reference list with a thorough breakdown of the sources used in the study is also included in the publication. Ultimately, the paper's structure is intended to aid the reader's comprehension and involvement with the study.

The objectives of a comparative study of machine learning models for hate speech detection can include:

1. Performance Evaluation:Analyse the effectiveness of various machine learning models. in detecting hate speech. To evaluate the efficiency of each model, it is necessary to include measures like accuracy, recall, F1-score, precision, and area under the curve of receiver operating characteristics (AUC-ROC).

2. Model Selection: Identify the most suitable machine learning model or combination of models for hate speech detection. By comparing the performance of various models, researchers can determine which models are better suited for this task and provide recommendations for selecting the most effective model for real-world applications.

3. Feature Analysis: Analyze the importance and relevance of different features or input representations in hate speech detection. This involves studying the impact of various linguistic and contextual features on model performance and understanding which features contribute the most to accurate detection.

4. Robustness Assessment: Evaluate the robustness of machine learning models against different types of hate speech, including variations in language, content, and intensity. Assess

how well the models generalize to different datasets and whether they maintain their performance in the presence of adversarial attacks or attempts to bypass the detection system.

5. Scalability Analysis: Examine the scalability of different machine learning models in handling large-scale datasets and real-time processing. Assess the computational requirements, training time, and inference speed of each model to determine their practicality for deployment in real-world scenarios.

6. Interpretability and Explainability: Examine how easily machine learning algorithms for detecting hate speech can be understood and justified. Recognise the elements that go into defining a piece of writing as hate speech, as well as how effectively the models can shed light on their thought process.

7. Generalizability and Transfer Learning: Assess the generalizability of machine learning models across different domains, languages, and cultures. Explore the potential of transfer learning techniques to adapt models trained on one dataset or language to perform well on new, unseen datasets or languages.

By achieving these objectives, the comparative study can contribute to advancing the field of hate speech detection by providing insights into the strengths and limitations of different machine learning models, facilitating the development of more accurate and robust detection systems.

Cybercrime refers to a broad range of unlawful actions carried out online and using digital technology. Several well-known forms of cybercrime include:

1. Hacking: Unauthorised entry into computer systems or networks with the intention of stealing data, interfering with business, or taking over the system being attacked.
2. Identity Theft: stealing a person's personal information, such as their SSN or credit card information, with the aim to pose as them or conduct fraud.
3. Phishing: Convincing people to divulge private information like passwords or financial information by sending false emails, texts, or websites that seem authentic.
4. Online fraud: Using misleading advertising, online auction fraud, or pyramid schemes to commit fraud online in order to get money or products.

5. Cyberbullying: Using online tools to harass, threaten, or degrade others, frequently through obnoxious and persistent behaviour.
6. Malware Attacks: Spreading harmful software such as viruses, worms, ransomware, or spyware to infiltrate computer systems, breach data security, or otherwise cause harm.
7. Distributed Denial of Service (DDoS): Flooding targeted networks or websites with traffic to interfere with their functioning and prevent people from accessing them.

Hate speech has emerged as a big issue in the internet sphere, alongside cybercrime. Any kind of speech that targets, threatens, or advocates violence against individuals or groups because of characteristics like race, religion, ethnicity, gender, or sexual orientation is referred to as hate speech. Multiple obstacles must be overcome in order to identify hate speech. These consist of:

1. Linguistic Complexity: It might be difficult to distinguish hate speech from objectionable content since it often uses nuanced language or coded phrases.
2. Contextual Understanding: Understanding the context in which a message is given, including any cultural allusions, sarcasm, or humour, is essential to deciphering its intended meaning.
3. Evolving Nature: Hate speech changes with time, including new linguistic techniques and evolving to avoid detection. As a result, detection systems must be updated often.
4. Multilingual and Multicultural Challenges: To prevent biases and errors, detecting hate speech in a variety of languages and cultural contexts calls for thorough language models.
5. Legal and Ethical Considerations: There can be a narrow line between freedom of expression and hate speech, thus determining what constitutes hate speech requires traversing complicated legal and ethical frameworks.

Advanced natural language processing (NLP) methods that can precisely analyse and interpret text in various linguistic and cultural settings while taking into account the specifics of hate speech expression are needed to address the challenge of hate speech identification. Additionally, it is essential for technology firms, decision-makers, and communities to work together to set rules, regulations, and policies that encourage responsible online behaviour and lessen the negative impacts of hate speech.

CHAPTER 2

MOTIVATION

There are numerous important reasons for conducting a comparison of machine learning algorithms for hate speech identification. These elements emphasise how critical it is to comprehend the benefits and drawbacks of various strategies and models when tackling the challenging issue of hateful speech detection. Following are the reasons for doing a comparison research of this kind:

1. Improved Performance: Hate speech detection is a challenging task due to the evolving nature of language, context, and cultural nuances. By comparing various machine learning models, researchers aim to identify the models that offer the highest accuracy, precision, recall, and F1 scores. This comparison helps in selecting the most effective models for hate speech detection, leading to improved performance and more reliable identification of abusive language.

2. Model Selection: The vast array of machine learning models available for hate speech detection necessitates careful evaluation and selection. Comparative studies allow researchers to assess the strengths and weaknesses of different models and understand their suitability for specific contexts or datasets. Through this process, researchers can make informed decisions regarding the choice of models and develop guidelines for selecting the most appropriate model based on the characteristics of the problem domain.

3. Generalizability: Hate speech detection is a multifaceted problem that spans across different platforms, languages, and communities. A comparative study helps in evaluating the generalizability of machine learning models across diverse datasets and linguistic contexts. By examining the performance of models across multiple datasets, researchers can gain insights into the robustness and adaptability of these models, providing guidance on their applicability in real-world scenarios.

4. Bias and Fairness: Bias and fairness are critical considerations in hate speech detection. Comparative studies enable researchers to investigate and understand potential biases within

different machine learning models. By evaluating the performance of models across demographic factors, linguistic disparities, and target communities, researchers can identify and address issues related to bias and fairness, promoting more equitable and unbiased hate speech detection systems.

5. Advancing the Field: Comparative studies contribute to the advancement of the field of hate speech detection by providing empirical evidence and insights into the strengths and weaknesses of different models. Such studies shed light on the key factors that impact model performance, identify areas for improvement, and guide future research directions. By building on existing knowledge and understanding the comparative performance of models, researchers can innovate and develop more effective techniques for hate speech detection.

6. Practical Applications: Hate speech detection has practical implications for online platforms, social media networks, and content moderation systems. Comparative studies help in identifying the most accurate and efficient models for implementation in real-world applications. By comparing models, researchers can provide valuable insights to industry practitioners and policymakers, enabling the development of more robust and reliable systems for combating hate speech and promoting a safer online environment.

In conclusion, a comparative study of machine learning models for hateful speech detection is motivated by the desire to improve performance, select appropriate models, enhance generalizability, address bias and fairness concerns, advance the field, and enable practical applications. Such studies play a crucial role in guiding the development of effective and ethical hateful speech detection systems, fostering a more inclusive and respectful online space.

CHAPTER 3

Literature Survey

Each paper has been analyzed to answer the questions specified earlier. A total of fourteen papers have been summarized focusing on algorithmic approaches and the formulated datasets. [1-5][11-13] are approach-based papers and [14][16] are dataset-related papers. Table 1 below provides a list of the papers.

Through investigation, it has been determined how well a number of extra-linguistic factors work with character n-grams to identify hate speech [1]. 614 Twitter users' data were gathered and annotated for the study. 16,914 tweets were included in the data, 3,383 of which were sent by 613 different individuals to express sexism, 1,972 by 9 different users to express racism, and 11,559 by no user to express either. To examine how different features affect prediction accuracy, a logistic regression classifier using 10-fold cross-validation was employed. The study discovered that demographic data, other than gender, enhanced model performance. It was discovered that character n-grams with lengths up to four and gender as an additional attribute produced the best outcomes.

Researchers in [2] evaluated the impact of annotator knowledge of hate speech on classification models by contrasting outcomes of classification derived from training on experienced and novice annotations. The dataset was created by selecting a sample of the 130k tweets that were extracted using [1]. The influence of features in the groups of novice and experienced annotators was examined using a 5-fold form of cross-validation on the features defined by the annotators. Models with comparable performance to earlier classification efforts were created using expert annotations. The study discovered that their top system performed noticeably worse on multi-class classification than Waseem and Hovy's (2016) binary classification challenge and was unable to meaningfully outperform it.

Researchers have demonstrated in this publication how fine-grained labeling can help in the assignment of identifying hate speech and have highlighted a few of the major obstacles to accurate categorization [3]. The dataset was manually coded by CrowdFlower employees and contained 25k tweets drawn from a sample of 85.4 million tweets by 33k users. To make the data less dimensional for the detection job, logistic regression using L1 regularisation was originally utilized. Afterward, a number of models from earlier research, including naive

Bayes, logistic regression, decision trees, linear SVMs, and random forests were examined. Then, to avoid over-fitting, each model was assessed with 5-fold cross-validation while only utilizing 10% of the sample.The linear SVM and logistic regression typically outperformed other models in a measurable way, according to our analysis after utilizing a grid search to loop over the models and parameters. The resulting model, which had an F1 score of 0.90, a recall of 0.90, and an overall precision of 0.91, was a logistic regression using L2 regularization. The precision and recall ratings for the despised class were 0.44 and 0.61, respectively, although it was found that over 40% of overall hate speech is misclassified.

Table 1: List of Papers

Year of Publication	**Title**
2016	Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter [2]
2016	Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter [1]
2017	Automated Hate Speech Detection and the Problem of Offensive Language [3]
2017	Deep Learning for Hate Speech Detection in Tweets [4]
2018	Detecting Online Hate Speech Using Context-Aware Models [5]
2018	Hateminers: Detecting Hate Speech against Women [11]
2018	Comparative Studies of Detecting Abusive Language on Twitter [12]
2019	A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media [13]
2020	HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection [14]
2020	Deep Learning Models for Multilingual Hate Speech Detection [15]
2020	Multilingual Twitter Corpus and Baselines for Evaluating Demographic Bias in Hate

	Speech Recognition[16]
2021	Hate speech detection using static BERT embeddings [17]
2021	HateBERT: Retraining BERT for Abusive Language Detection in English [18]
2022	Semi-supervised Stance Detection of Tweets Via Distant Network Supervision [19]

Researchers have concentrated on the challenge of classifying a tweet as racist, sexist, or neither in this work [4]. For this goal, CNN and LSTM-based architectures were employed with text processing algorithms such as (1) Char n-grams (2) TF-IDF (3) BoW 16K annotated tweets from the [1]. The 16K tweets were divided into 3383 sexist and 1972 racist tweets, with the remaining tweets being classed as neither sexist nor racist. The word TF-IDF method was determined to be superior to the character n-gram method among the baseline techniques. FastText was outperformed by LSTM, which in turn was outperformed by CNN.The approach that produced the greatest results, "LSTM + Random Embedding + GBDT," initialised tweet embeddings to vectors, trained the LSTM using back-propagation, then trained the GBDT classifier using the learnt embeddings.

In this study, researchers investigated two types of models to automatically detect hate speech: feature-based logistic regression models and neural network models [5]. 1528 Fox News user comments for a total of 10 discussion threads for 10 widely read news articles made up the dataset that was used. In the end, ensemble models, neural network models, and logistic regression models were chosen for implementation after considering both machine learning and deep learning techniques. After incorporating both lexicon-derived features and word-level n-gram features, the logistic regression model generated positive results. Both ensemble models outperformed using a single model in terms of classification performance and improved hate speech detection. With gains in the AUC score and F1-score of 2.8% and 2.5%, respectively, the results showed that the logistic regression model employing features extracted from a comment and both forms of context performed the best.Online hate speech identification was greatly enhanced by the attention mechanism and bidirectional LSTM neural network, raising AUC by 5.7%.The average score model increased the AUC score by

around 7%, whereas the max score ensemble model increased recall by more than 20% with equivalent precision and F1 score by about 10%.

The researchers in this article concentrated on the identification of English-language misogynistic tweets [11].In contrast to the test dataset's 1000 unlabeled tweets, the training dataset's 4000 labeled tweets. The English corpora were manually labeled by several annotators into three levels: (1) Gender bias (Misogyny vs Not Misogyny)(2) The target (active vs. passive); (3) The misogynistic categories for the purpose of classification. The Logistic Regression system outperformed all other model systems tested with an accuracy of 70.4%.

Researchers in [12] investigated the effectiveness of several models in identifying abusive language by comparing accuracy using the most popular machine learning classifiers as well as neural network models. "Hate and Abusive Speech on Twitter" was the dataset used [22] tweets are divided into four categories: "regular," "spam," "hateful," and "abusive." Nave Bayes (NB), and Logistic Regression (LR), a linear LR, were the conventional machine learning models that were used.BFGS improvement (3) Support Vector Machine (SVM): Linear SVM with logistic loss function and L2 regularisation constant 1. (4) Random Forests (RF): 10 randomized decision trees are used to average probabilistic forecasts. Gradient Boosted Trees (GBT) is a type of tree boosting that uses a logistic loss function with a learning rate of 1. Use of neural networks included: (1) CNN: There are three convolutional filters of varying sizes with ReLU activation and a max-pooling layer in the word-level CNN models. (2) RNN: For each recurrent unit, a GRU cell was added to the bidirectional RNN which was the starting point.The LR model was shown to be the best effective traditional machine learning model for classifying abusive language. RNNs with LTC modules have the greatest accuracy rating for neural network models. According to experimental findings, bidirectional GRU networks with LTC produced the best reliable results for identifying abusive language.

This study compiles a database that examines many facets of hate speech[14]. Twenty thousand posts from Twitter and Gab made up the dataset. Multiple annotators physically

annotate each post, and the class labels are chosen by a qualified majority. This led to the creation of a new benchmark dataset for the detection of hate speech. Each data point has a label—hate, offensive, or normal—along with information on the target community and passages of text that the annotators who support the label have highlighted. According to the report, demographic data that would have been helpful for the task of annotating the data was not taken into account during the investigation. Moreover, the English language was the main focus of this effort. It didn't take into account hate speech in multiple languages. Large-scale analyses of multilingual hate speech in 9 languages from 16 different sources were conducted by academics [15]. Models were trained in multilingual environments using multilingual word/sentence embeddings. For sentences, LASER embeddings were being used, while MUSE embeddings were being used for words. The models utilized for detection were BERT, mBERT, CNN + GRU, and For the experimental MUSE+CNN-GRU, Translation+BERT, and LASER+LR combinations. When the evaluation of the two models was compared, it was found that LASER + LR places more emphasis than mBERT on hostile terms, such as words like "pigs," and so on. The context for the hateful terms seems to be what mBERT was seeking. Complex techniques that emphasize context, like mBERT, might be more effective in less harmful places, like Twitter. Overall, it was discovered that BERT models performed significantly better in high-resource scenarios than LASER + LR did in low-resource situations. For all languages in a monolingual situation, LASER + LR demonstrated the greatest performance in low-resource settings. It was found that MUSE + CNN-GRU had the worse performance in practically all scenarios. In several languages, such as German, Polish, Portuguese, and Spanish, Translation + BERT looked to perform well. Overall, there wasn't a "one-size-fits-all" solution, but Translation + BERT seemed to be an excellent compromise. In a scenario including several languages including zero-shot analysis, mBERT outperformed LASER + LR significantly in all three languages (Arabic, French, and German). In the final six languages, LASER + LR faired better, with notably strong outcomes in Portuguese and Italian. With no Portuguese training data, zero-shot Laser + LR in Portuguese produced an F-score of 0.6567, which was near the highest score of 0.6941. (using full Portuguese training data). For languages like German, Arabic, and French, mBERT proved to score higher almost often than LASER + LR (low resource and Full Data). On the other hand, LASER + LR consistently delivered excellent results for the Portuguese language.For the other five languages, LASER + LR outperformed mBERT when using the target language's complete training data, but mBERT outperformed both methods significantly in low-resource environments.

For the purpose of detecting hate speech, researchers assembled and published a multilingual Twitter corpus containing four inferred demographic factors: age, nationality, sexuality, and racial group [16]. In English [1] [2][22], Italian [7], Polish [23], Portuguese [22], and Spanish [1] [2][22], the dataset. (1) LR:: First, TF-IDF-weighted properties of up to tri-grams were retrieved from the corpora using the 15K features that were utilized the most frequently, with a minimum feature frequency as 2. Afterward, a Logistic Regression was trained using SciKit Learn. (2) CNN (3) RNN: a recurrent neural network (RNN) classifier was developed. (4) BERT. According to empirical studies, classifiers may be biased as a result of linguistic disparities between demographic groups. This data can be used to explore bias concerns in multilingual contexts with several user characteristics and to evaluate how unbiased text classifiers are based on author-level features. In every language, CNN, LR, and RNN consistently beat the other three baseline classifiers. Additionally, in four of the five languages, CNN and RNN performed better than LR (except Spanish). The BERT findings, on the other hand, showed a greater disparity in the English dataset and were less than the other standards.

In this publication, researchers used various deep learning models in conjunction with static BERT embeddings to improve the current deep learning-based classifiers [17]. A balanced, binary version of the ETHOS dataset was used for the investigation. Static BERT embeddings were integrated with DNNs (CNN, LSTM, BiLSTM, and GRU) to extract contextual information. From a huge corpus of datasets, a static BERT embedding matrix encoding the embedding for each word was created. DNN classifiers were then applied to this matrix to detect the existence of hate. It was discovered that static BERT embeddings offered superior feature representation to fastText and GloVe. BiLSTM with static BERT embeddings surpasses all other DNNs considered in the study in all measures, according to the findings.

In order to advance research in the field of detecting hate speech, researchers have released HateBERT, a BERT that has been trained to detect abusive language phenomena[18]. The study made use of the Reddit Abusive Language English dataset, or RAL-E. The English BERT base-uncased model was retrained with 1,478,348 messages from the RAL-E dataset

using the Masked Language Model. The remaining 14,932 messages were utilized as a test set. The model was retrained using up to 512 sentence-piece tokens over 100 epochs in batches of 64 samples. Adam was employed, with a 5e-5 learning rate. One Nvidia V100 GPU was used to train a code4 for a hugging face. Due to this, two aspects of the HateBERT base-uncased BERT model have been shifted: (1) language diversity; and (2) polarity. The general BERT model was found to be less effective for hate speech detection than HateBERT. For example, when moving from general abusive language phenomena to more specialized ones, HateBERT is more portable than a generic BERT model. When switching from general abusive language phenomena (i.e., offensive language) to more specific ones, HateBERT offers superior portability than a generic BERT model (i.e., abusive language or hate speech).

Researchers in this publication developed SANDS, a novel framework for classifying Twitter stances that makes use of following network data and remote supervision to learn tweet stances off sparse datasets [19]. The dataset comprised over 87,000 users' 236,000 politically sensitive tweets from two different demographics (the United States and India), the graph of followers and followees, and more than 8,000 tweets with linguistic annotations. Two classifier architectures were used for the classification task: (1) a convolutional architecture that concentrated on brief, contiguous text segments and (2) a bidirectional LSTM-based architecture that encoded geographically separated textual vectors into a single representation. When faced with limited annotated data, rough textual descriptions, and a severely unbalanced class-wise sample distribution, SANDS performed significantly better than other models.On the US (India)-based datasets, SANDS achieved a macro-F1 score of 0.55 (0.49), significantly outperforming 17 baselines (including SANDS variants).

The literature survey on hate speech detection papers covers datasets, deep learning techniques (CNN, LSTM, RNN), and machine learning techniques (Logistic Regression, Random Forests, SVM, Naive Bayes, Gradient Boosting Decision Trees). BERT, a pre-trained language model, is also utilized. This taxonomy showcases the diverse approaches and methodologies employed in hate speech detection research.

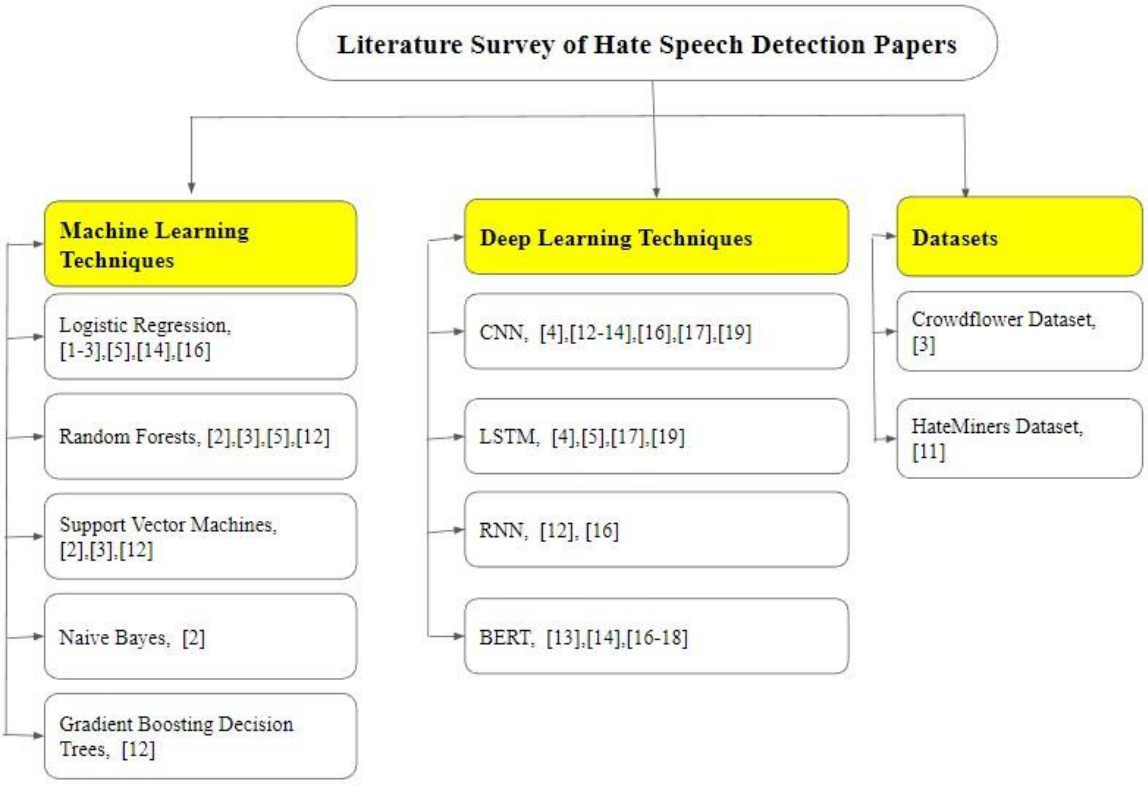


Figure 2: Literature Survey Taxonomy

CHAPTER 4

CONCLUSION FROM LITERATURE REVIEW

1. The incorporation of demographic information like gender and character n-grams, particularly up to length four, has been demonstrated to improve the effectiveness of detection of hateful speech algorithms.
2. The expertise and knowledge of annotators significantly impact classification models. Models trained on annotations from experienced annotators tend to perform better than those trained on annotations from novice annotators.
3. Logistic regression models using L2 regularization and techniques like TF-IDF have been effective in identifying hate speech, although misclassification rates for overall hate speech can still be high.
4. For classifying tweets as , sexist, racist or neither, approaches such as CNN and LSTM-based architectures with TF-IDF and random embedding techniques have shown promising results.
5. Ensemble models, combining different approaches, have demonstrated improved classification performance, including increased AUC scores and F1-scores, for hate speech detection.
6. The identification of misogynistic tweets has been achieved through manual labeling and logistic regression models, with an accuracy of 70.4%.
7. Traditional machine learning models, such as LR and SVM, have shown effectiveness in classifying abusive language and hate speech.
8. Neural network models, specifically RNNs with LSTM cells and Bidirectional GRU networks, have demonstrated high accuracy in identifying abusive language.
9. BERT-based models, particularly when combined with CNN for fine-tuning, have outperformed baselines in terms of F1 scores for hate speech detection tasks.
10. The inclusion of demographic factors, such as age, nationality, sexuality, and racial group, has been explored to evaluate bias concerns and improve text classifiers' performance in multilingual contexts.

11. Transfer learning techniques, such as combining pre-trained BERT models with supervised fine-tuning, have shown improved performance in comprehending hate speech across different datasets.
12. Multilingual analysis of hateful speech detection has been conducted using models like BERT, LASER, and Translation+BERT, where performance varied across languages, highlighting the importance of language-specific approaches.
13. Static BERT embeddings have been found to provide superior feature representation compared to fastText and GloVe embeddings, with BiLSTM models achieving the best results in hate speech detection.
14. HateBERT, a BERT model specifically trained to detect abusive language phenomena, has shown better portability and performance than generic BERT models in detecting hate speech and specialized abusive language.
15. The SANDS framework, utilizing following network data and remote supervision, has demonstrated superior performance in classifying Twitter stances, even with limited annotated data, rough textual descriptions, and unbalanced sample distributions.

Overall, the literature study highlights the effectiveness of various machine learning and deep learning models, the significance of demographic factors and transfer learning, and the need for language-specific approaches in hate speech detection and classification tasks. These findings contribute to the ongoing efforts in developing robust and accurate methods for identifying and combating hate speech in online platforms.

CHAPTER 5

Tools and Technologies Used

The project report's "Tools and Technologies Used" section lists the various software tools, programming languages, frameworks, and technologies used to carry out the project. This part offers a succinct summary of the technical tools used to create and carry out the project, demonstrating the technical know-how and infrastructure needed for its effective execution.

3.1 Tools Used:

1. Google Colaboratory

A cloud-based platform for development called Google Colab, sometimes known as Google Colaboratory, allows users to create, run, and share Python programmes. With sophisticated computing resources, including GPUs, readily available for free, it offers a seamless environment for data science and machine learning applications. Users may access Colab using a web browser, doing away with the necessity for local setup and installation. Multiple users may work concurrently on the same notebook because to its collaborative capabilities, making it a flexible tool for team tasks. Due to its simplicity of use, wide library support, and interaction with other Google services, Google Colab has become a well-liked platform for data science as well as machine learning experimentation, prototyping, and learning.

Advantages of Google Colab include the following:

1. Python Code Execution: The Python code execution environment on our platform is easy and effective. With the help of our dependable infrastructure, your code executes without errors and produces precise results, letting you concentrate on the central idea of your project without worrying about how it will be put into action.
2. Documentation Support: We recognise the value of simple and clear documentation for your initiatives. With the help of our platform's extensive documentation assistance, you can provide in-depth justifications, illustrations, and tutorials that will help people better understand your codebase and work with you and contribute to it.

3. Notebook Creation and Sharing: With our technology, creating interactive and dynamic notebooks is simple. Data exploration and analysis, testing out various techniques, and sharing your discoveries with colleagues or the larger community are all encouraged. You may quickly share your notes with others, facilitating teamwork and allowing for smooth information transfer.
4. GitHub Integration: By connecting our platform with GitHub, you may streamline your development process. Push, pull, and synchronise your code repositories with ease for version control and simple team collaboration. Maintaining a seamless development process requires being able to effortlessly keep track of changes, evaluate code, and integrate updates.
5. External Dataset Integration: With our platform, accessing and integrating other datasets is a breeze. You can quickly import and modify datasets from a variety of sources, including external APIs and public repositories, enabling you to deal with real-world data and create models that are more thorough and precise.
6. Integration with Deep Learning Frameworks: By smoothly integrating our platform with well-known deep learning frameworks, you may improve your deep learning applications. Utilise the vast libraries and pre-trained models provided by frameworks like TensorFlow, PyTorch, or Keras to expedite your development and provide cutting-edge outcomes.
7. Free Cloud Service with GPU: With the free cloud service with GPU support provided by our platform, you may benefit from enhanced processing capabilities without paying any additional fees. Run experiments quickly and efficiently, train complicated models, and perform computationally demanding tasks all while saving money.

2. Jupyter Notebook

Users may create and share documents including live code, equations, visualisations, and narrative prose using the open-source Jupyter Notebook online application. It offers a computer environment that is interactive and supports a variety of programming languages, such as Python, R, and Julia. Due to its versatility and usability, Jupyter Notebook is frequently used in the fields of data science and scientific research. Users can write and execute code directly within the notebook, view the results, and include explanatory text or visualizations to provide a comprehensive narrative. The notebook interface allows for the

creation of data-driven documents that combine code, visualizations, and textual explanations, making it a powerful tool for exploratory data analysis, prototyping, and sharing research findings. Jupyter Notebook encourages an iterative and collaborative workflow, enabling users to share their notebooks with others, facilitating collaboration, and promoting reproducibility. Its interactive nature and ability to mix code, visualizations, and text make Jupyter Notebook a versatile tool for data analysis, machine learning, and scientific computing.

Here are some advantages of Jupyter Notebook for data science:

1. Interactive and exploratory
2. Easy visualization
3. Documentation and storytelling
4. Code sharing and collaboration
5. Flexible language support
6. Interactive widgets
7. Kernel separation

3.2 Technology Used:

Machine Learning

The field of AI known as "machine learning" is concerned with developing statistical models and algorithms that enable computers to learn from data without being explicitly programmed. It entails the creation of computer models that can automatically decipher complicated relationships and patterns in data and use that understanding to forecast the future or take action.

Here's an overview of the machine learning process

1. Data collection: Any machine learning effort must start with the collection of pertinent and excellent data. This information may be unstructured (text, photos, audio, etc.) or organised (in clearly specified formats like tables). For the machine

learning model to be trained successfully, the data must be adequate and representative.

2. Data preprocessing: Preprocessing is frequently necessary after data has been obtained. This entails preparing the data for analysis by cleaning it by eliminating noise, addressing missing values, normalising or scaling features, and so on.
3. Feature selection and engineering: The properties or attributes of the data that are utilised to produce predictions are referred to as features. The performance of the machine learning model is enhanced in this stage by the selection or engineering of pertinent features. In feature engineering, existing features may be modified or combined to produce fresh ones that more accurately depict the underlying patterns in the data.
4. Model selection and training: Algorithms that learn from data and generate predictions are known as machine learning models. There are many different kinds of models, including neural networks, support vector machines (SVM), decision trees, and linear regression. The particular problem and data properties influence the model choice. In order for the model to understand the underlying patterns and change its internal parameters to minimise the prediction errors, it is given labelled data (input data with known associated outputs) during training.
5. Model evaluation: After the model has been trained, it must be tested to determine how well it performs and can generalise. The effectiveness of the model on unobserved data is measured using evaluation measures like accuracy, precision, recall, F1 score, or mean squared error. The assessment aids in the detection of possible problems such as underfitting (failure to capture the underlying patterns in the data) or overfitting (doing well on training data but badly on fresh data).
6. Model tuning: Hyperparameter adjustment can be done if the model's performance is unsatisfactory. Hyperparameters are variables that are established before to the training process rather than ones that are learnt from data. By finding the ideal compromise between complexity and generalisation, adjusting these hyperparameters can improve the model's performance.
7. Model deployment and prediction: The model may be placed into a production setting to make predictions on fresh, unforeseen data after being trained and assessed. The learnt patterns are then applied to the input data by the model to provide predictions or judgements. The result will vary depending on the particular issue at hand, whether

it be picture classification, sales forecasting, product recommendations, or anomaly detection.

8. Monitoring and maintenance: To maintain the model's accuracy and dependability after deployment, its performance has to be regularly checked. The model might need to be updated or retrained with fresh data over time to take into account evolving trends or needs.

Machine learning is a rapidly evolving field with various techniques and algorithms designed to handle different types of problems.It has uses in many different industries, including healthcare, banking, natural language processing, fraud detection, image and audio recognition, and many more.

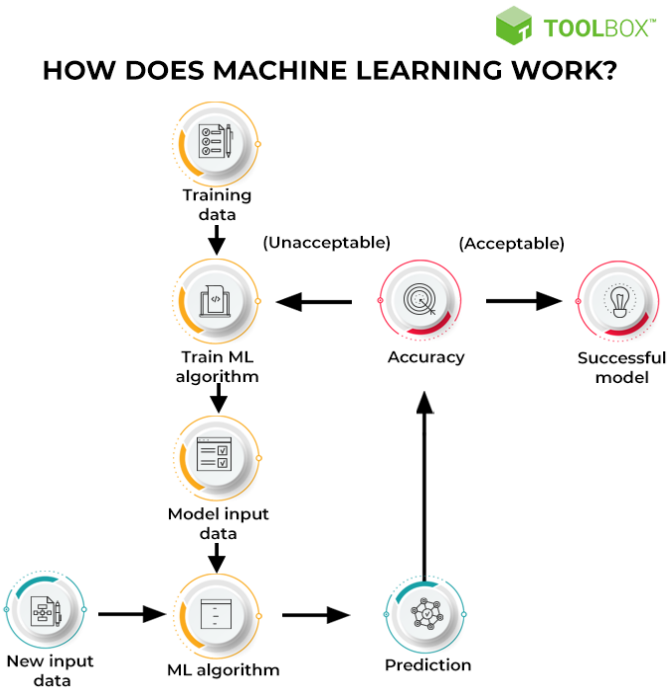


Figure 3: General Machine Learning Methodology

Algorithms Used:

- Logistic Regression:

By calculating the likelihood that a certain event will occur depending on input data, the classification process known as logistic regression may be used to predict discrete outcomes. It is appropriate for issues involving binary classification since it uses a logistic function to represent the connection between the dependent variable and independent factors.

Formula (Binary Classification): p = 1 / (1 + e^(-z)), where z = $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta\square x\square$

Formula (Multiclass Classification): p(y = k) = e^(z□) / (Σe^(z□)), where z□ = $\beta_0\square + \beta_1\square x_1 + \beta_2\square x_2 + ... + \beta\square\square x\square$

- Random Forest:

An ensemble learning technique called Random Forest uses many decision trees to provide predictions. Using random selections of the training data and characteristics, it creates a large number of decision trees, and then it combines the predictions from each tree to get the final prediction. The resilience, scalability, and capacity for handling highly dimensional data of Random Forest are well recognised.

- Naive Bayes:

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem and the assumption of independence among features. Despite its simplicity, Naive Bayes is effective in many real-world applications. It is particularly popular for text classification tasks, such as spam filtering or sentiment analysis.

Formula: P(y|X) = (P(X|y) * P(y)) / P(X), where P(y|X) is the posterior probability, P(X|y) is the likelihood, P(y) is the prior probability, and P(X) is the evidence.

- Adaboost:

Adaboost (Adaptive Boosting) is an ensemble learning method that iteratively combines weak classifiers to create a strong classifier. It assigns higher weights to misclassified samples in each iteration, allowing subsequent weak classifiers to focus

on the previously misclassified instances. Adaboost is known for its ability to handle complex datasets and improve classification performance.

$w_{i+1} = w_i * \exp(-\alpha_i * y_i * h_{xi})$, where w_i is the weight of the i-th sample, α_i is the weight for the weak classifier, y_i is the true label, and h_{xi} is the prediction of the weak classifier.

- Gradient Boosting Decision Trees:

  Gradient Boosting Decision Trees is another ensemble learning method that combines weak decision trees. It builds trees sequentially, with each tree aiming to correct the mistakes made by the previous ones. It iteratively fits the new tree to the residuals of the previous trees. Gradient Boosting Decision Trees is widely used and often achieves state-of-the-art performance in various machine learning competitions.

- Support Vector Machines:

  Support Vector Machines (SVM) is a powerful classification algorithm that finds an optimal hyperplane to separate data points of different classes. It maps the input data into a higher-dimensional feature space and constructs a decision boundary that maximizes the margin between classes. SVM can handle complex decision boundaries and is effective in both linear and nonlinear classification tasks when properly tuned.

  Formula: $y = \text{sign}(w \cdot x + b)$, where w is the weight vector, x is the input vector, and b is the bias term

- TF-IDF:

  TF-IDF (Term Frequency-Inverse Document Frequency) is a numerical representation of the importance of a term in a document collection. It considers both the frequency of a term in a document (TF) and its rarity across the entire collection (IDF). TF-IDF assigns higher weights to terms that appear frequently in a document but rarely across other documents. This technique is widely used in information retrieval, text mining, and document classification tasks to identify important keywords and reduce the impact of common terms.

- GridSearchCV:

  GridSearchCV is a method for hyperparameter tuning in machine learning models. GridSearchCV systematically tries all possible combinations of hyperparameters to find the best configuration that optimizes the model's performance based on a specified evaluation metric. It automates the process of finding the optimal hyperparameters and helps improve the model's accuracy by fine-tuning the parameters.

CHAPTER 6

System Design and Methodology

The suggested technique used to categorize tweets into two groups—"hate speech and non-hate speech"—is explained in this section. The whole research technique is shown in Fig. 4. Data collection, data preprocessing, feature engineering, data splitting, base model creation, best base model selection, hyperparameter tuning of selected models, and classification model assessment are the eight main procedures for the research methodology as indicated in this image. In the sections that follow, each process is covered in full.

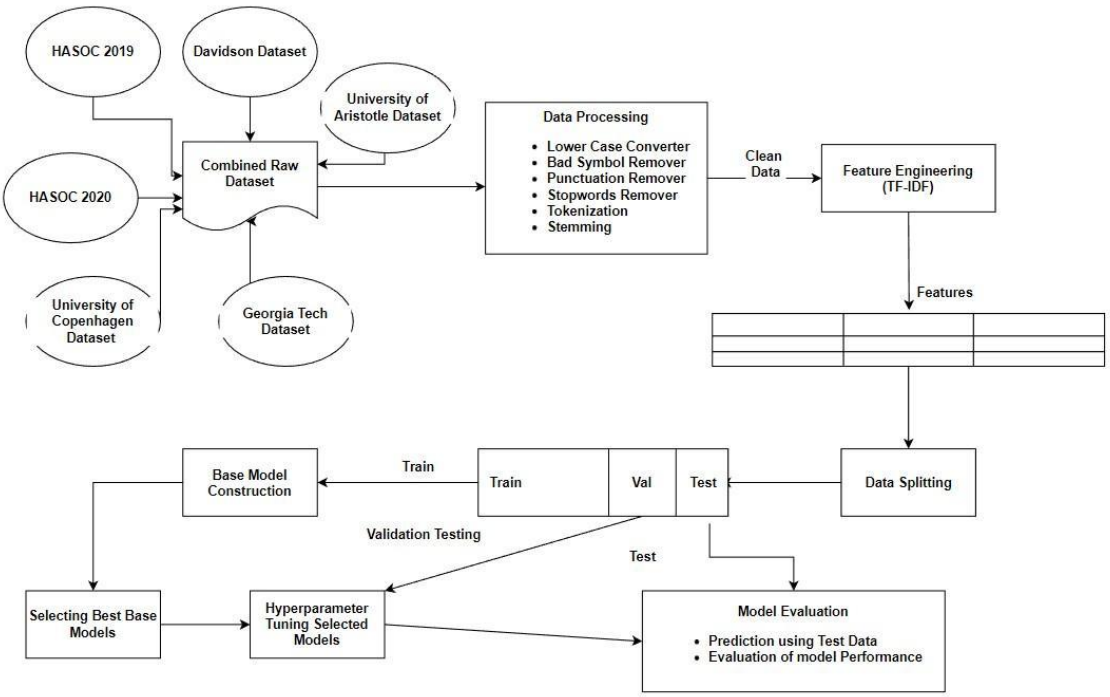


Figure 4: System Design

4.1 Data Collection

For this study, we gathered a dataset of hate speech tweets that were made publicly available. The University of Aristotle Hate Speech Dataset, University of Copenhagen Hate Speech Dataset, HASOC 2019, HASOC 2020, and Georgia Tech Dataset were combined

with the Davidson Dataset to create this dataset. The tweets in the merged dataset are categorised into two separate groups., namely hate speech and non-hate speech. There are 30270 tweets in this dataset. Of these, 22.9% of tweets fall under the category of hate speech, while 77.1% fall under the category of non-hate speech.

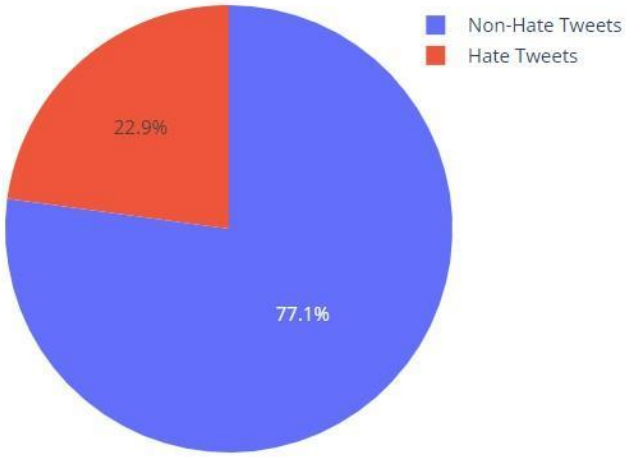


Figure 5: Class Distribution of combined dataset

Table 2: Datasets used

Name of Dataset	**Detail of Instances Used**
Davisdon Dataset[24]	Non-hate tweets: 23353 Hate Tweets count: 1430
University of Aristotle Hate Speech Dataset[20]	1614 Hate Speech Tweets used
University of Copenhagen Hate Speech Dataset[1]	2684 Hate Speech Tweets used
HASOC 2019 [21]	58 Hate Speech Tweets used
HASOC 2020 [22]	131 Hate Speech Tweets used
Georgia Tech Hate Speech Dataset [23]	1500 Hate Speech Tweets used

The Davidson Dataset [24] is a widely used dataset for hate speech detection, consisting of tweets labeled into three categories: hate speech, offensive language, and neither. It provides a diverse collection of Twitter data, allowing researchers to evaluate the performance of machine learning models in identifying various forms of abusive language. The dataset is valuable for understanding hate speech dynamics and developing effective detection systems.

The University of Aristotle Hate Speech Dataset [20] is another notable resource that focuses on hate speech detection. It comprises posts from various social media platforms, including Twitter, Facebook, and forums. The dataset offers labeled instances of hate speech, offensive language, and non-hate speech, enabling researchers to explore the complexities of hate speech detection across different online platforms.

The University of Copenhagen Hate Speech Dataset [1] provides a collection of tweets annotated for hate speech, offensive language, and neither. It covers multiple languages, including English, Danish, and Greek, facilitating cross-lingual analysis of hate speech detection models. This dataset enhances the understanding of hate speech detection in diverse linguistic contexts.

The HASOC (Hate Speech and Offensive Content) datasets, namely HASOC 2019 [21] and HASOC 2020 [22], are specifically designed for hate speech and offensive content detection in social media. These datasets consist of multilingual data from Twitter, Facebook, and Instagram, covering languages such as English, German, and Hindi. They offer valuable resources for evaluating the performance of machine learning models in detecting hate speech across different social media platforms and languages.

The Georgia Tech Hate Speech Dataset [23] is a collection of tweets annotated for hate speech, offensive language, and neither. It focuses on hate speech targeting women and racial minorities. This dataset provides insights into the intersectionality of hate speech and offers researchers the opportunity to develop models that address specific forms of targeted hate speech.

These datasets play a crucial role in advancing research on hate speech detection by providing labeled instances for model training and evaluation. They enable researchers to develop and compare machine learning models across different platforms, languages, and target communities, ultimately contributing to the development of more accurate and robust hate speech detection systems.

4.2 Text Processing

Several research studies have explained that using text preprocessing makes better classification results [17]. Therefore, in our dataset, we used several preprocessing approaches to remove illuminating and noisy information from the tweets. We converted the tweets into lowercase during preprocessing. Additionally, we used pattern-matching algorithms to exclude all URLs, usernames, white spaces, hashtags, punctuation, and stop-words from the collected tweets. In addition, we have carried out tokenization using preprocessed tweets. Each tweet is first tokenized into words or tokens, and then words are further transformed by the Porter stemmer into their root forms, such as outraged to insult.

The details of the steps involved are mentioned below:

1. Lower case conversion: In this NLP step, all text is converted to lowercase. It helps in standardizing the text and ensures that words with different cases are treated as the same. For example, "Hello" and "hello" will be considered identical after lower case conversion.
2. Punctuation removal: Punctuation marks such as commas, periods, exclamation marks, and question marks are removed from the text. This step helps eliminate noise and allows the focus to be on the essential words and their meanings.
3. Hashtags and retweets removal: In the context of social media text, hashtags and retweets are commonly used. Removing hashtags involves eliminating the '#' symbol and extracting the actual text. Retweets, indicated by the "RT" symbol, are also removed to ensure that the content focuses on the original message rather than sharing information from other users.
4. Bad symbols removal: Bad symbols refer to any characters or symbols that do not contribute to the text's meaning or may hinder analysis. This step involves removing

symbols such as dollar signs, ampersands, asterisks, or any other non-alphanumeric characters that are irrelevant to the text analysis.

5. Stopwords removal: Stopwords are commonly occurring words in a language that do not carry significant meaning and are often filtered out during text processing. Examples of stopwords include "a," "an," "the," "is," and "are." Removing stopwords helps reduce noise and focuses on more informative words in the text.

6. Tokenization: Tokenization involves breaking down a text into individual units called tokens. These tokens can be words, phrases, or even sentences. Tokenization is a crucial step as it lays the foundation for further analysis, such as counting word frequencies or creating language models.

7. Stemming: Stemming is a process that reduces words to their root or base form by removing suffixes or prefixes. For example, stemming would convert "running" to "run" or "dogs" to "dog." Stemming aims to unify similar words and reduce the dimensionality of the data, making it easier for analysis.

8. Lemmatization: Lemmatization aims to find the base form of a word using more sophisticated linguistic rules and a vocabulary. Unlike stemming, lemmatization considers the context and meaning of words. For example, lemmatization would convert "better" to "good" and "went" to "go." Lemmatization helps maintain the integrity of the text and improves the accuracy of subsequent analyses.

4.3 Feature Engineering

Algorithms used in machine learning are unable to comprehend the categorization rules created from the raw text. In order to comprehend classification criteria, these algorithms require numerical qualities. As a result, feature engineering is a crucial stage in text categorization. In this step, the main characteristics of the raw text are extracted, and the qualities are then represented quantitatively. The feature engineering method TF-IDF [18] has been used in this investigation.

Working of TF-IDF:

Text data is represented numerically using the TF-IDF (Term Frequency-Inverse Document Frequency) method. It encapsulates the significance of words in a text in comparison to a group of papers.

The frequency of a term (word) inside a text is gauged by the TF component of the TF-IDF. It determines the proportion between the total number of words in a document and the number of times a given term appears in that text. It makes sense that a phrase would be more significant in describing the substance of a document if it appeared more frequently in that text.

The TF-IDF's IDF component calculates a term's rarity across a group of documents. The logarithm of the inverse of the ratio between the total number of documents and the number of documents containing the phrase is computed. As a phrase appears in more papers, the IDF value declines, suggesting that popular terms are less informative than uncommon words.

To convert data into vectors using TF-IDF, the following steps are typically followed:

1. Tokenization: The text data is divided into individual tokens (words, phrases, or n-grams) to establish a vocabulary.
2. Term Frequency (TF) calculation: The frequency of each phrase used in each text is calculated. As a consequence, a matrix is created, with each row representing a document, each column a phrase, and the cell value representing the frequency of the term inside the document.
3. Inverse Document Frequency (IDF) calculation: One may get the IDF value for each term by computing the logarithm of the value of the inverse document frequency, which is equal to the sum of documents divided by the total number of documents containing the word.
4. TF-IDF calculation: Each term's TF and IDF scores are calculated for each document by multiplying the terms' TF and IDF values. This results in a TF-IDF matrix where each cell has the TF-IDF score for the relevant phrase in the document and each row represents a document.

The resulting TF-IDF matrix represents the text data in vector form, with each document represented as a vector of TF-IDF values. These vectors capture the relative importance of terms within each document and enable further analysis, such as similarity measurement, clustering, or classification, based on the content of the documents.

4.4 Exploratory Data Analysis

Any research concerning hate speech must use exploratory data analysis (EDA). EDA includes looking at the data to find outliers, trends, and patterns. It enables researchers to comprehend the data's features more thoroughly and to spot any problems or biases that could exist. In the context of research on hate speech, EDA may entail determining the frequency and distribution of hate speech across various demographic groups, identifying the terminology or expressions that are most often used in hate speech, and assessing the environment in which hate speech is expressed. Examining the connections between hate speech and other factors, including the frequency of hate crimes or the usage of hate speech in political discourse, may also be part of EDA. Ultimately, EDA is a crucial instrument for understanding the type and extent of hate speech and for guiding the creation of successful solutions to this issue.

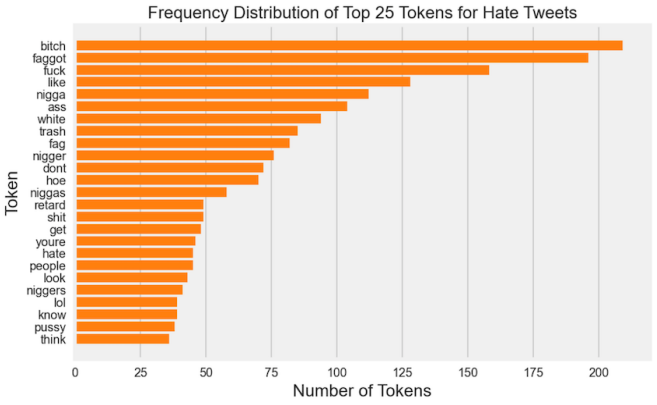


Figure 6: Frequency distribution of top 25 Tokens for Hate Tweets

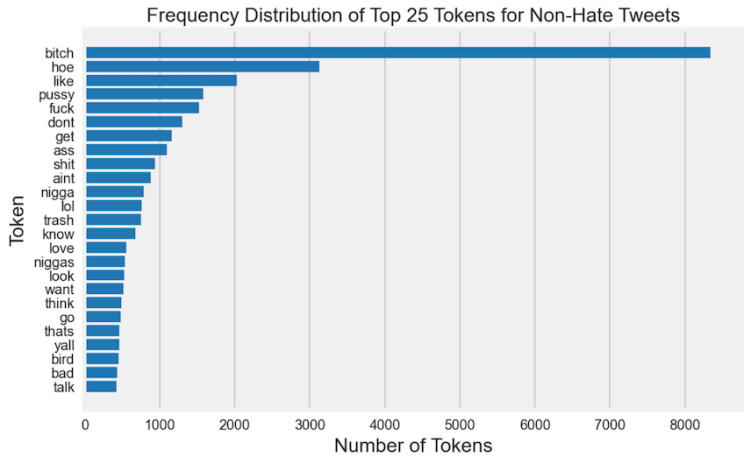


Figure 7: Frequency distribution of top 25 Tokens for Non-Hate Tweets

There are common terms and expressions used in both forms of communication, according to an exploratory examination of the top 25 words in both the Hate and Non-Hate categories. This demonstrated that the vocabulary used to communicate hate speech and non-hate speech are not fundamentally dissimilar. It also implied that some of the most popular terms in everyday speech may be appropriated and used to spread hate. This reinforced the necessity of supporting civil discourse and excellent communication in addition to monitoring hate speech in hopes of preventing the propagation of hate expression through social media.

4.5 Data Splitting

Table 3 displays the whole dataset's class-wise distribution as well as the data set following splitting (i.e. Training set, Validation set, and Test set). To divide the preprocessed data into training and validation sets, we utilized a 75:25 ratio, or 75% for training and 25% for validation. To create the test data, the validation dataset was once more divided in half using a 60:40 ratio. To learn the rule base, the classification model has to be trained using data for training. To evaluate the model's effectiveness, the validation data is used. Using test data on hypothetical scenarios, the categorization model is also assessed.

Table 3: Data Split

	Class Label	Total Instances	Training Instances	Validation Instances	Testing Instances
0	Non-Hate Tweets	23353	17514	3503	2336
1	Hate Tweets	7417	5562	1113	742

4.6 Machine Learning Models

It is thought that there isn't a single classifier that excels on all types of datasets when applying machine learning to a task. As a result, it is advised to test a variety of classifiers on a master feature vector to see which produces the best outcomes. So, we chose the Multinomial Naive Bayes, LR [28], SVM [26], RF [25], Adaboost [27], and GBDT as our six distinct classifiers.

4.7 Classifier Evaluation

In this stage, the classifier that was developed will be used to predict the class of unlabeled text using the test set. The effectiveness of the created classifier is assessed using a range of performance measures. Here is a brief description of a few common performance indicators for text categorization.

1) Precision: Another name for precision is the positive projected value. It is the percentage of predicted positives that really turn out to be positive. Equation (1) summarizes the precision calculation formula.

$$Precision = \frac{TP}{(TP+FP)} \quad (1)$$

2) Recall: The percent measure of favorable outcomes that actually occur and are anticipated to occur. Observe "(2)."

$$Recall = \frac{TP}{(TP+FN)} \quad (2)$$

3) F1 Score: It represents the harmonic mean of recall and accuracy (as shown in Equation 3). Precision and recall are given equal weight in the standard F1 Score. Equation (3) presents the formula to calculate f1 score.

$$F1\ Score = \frac{2 \times (precision \times recall)}{(precision+recall)} \quad (3)$$

4) Accuracy: It is the number of cases that were appropriately categorized. The formula is given in equation (4).

$$Accuracy = \frac{(TP+TN)}{(TP+FP+TN+FN)} \quad (4)$$

CHAPTER 7

RESULTS

We have trained our aggregated datasets on six classifiers and the model training and testing results and shown in below mentioned table. Further to this, the hyperparameter tuning of these classifiers has been performed using GridSearchCV to obtain optimal parameters and again the model training and testing results have been evaluated and presented in table 5.

Table 4: Base Model Testing Results

	Accuracy	**F1 Score**	**Recall**	**Precision**
Multinominal Naive Bayes	0.86	0.61	0.44	0.97
Random Forest[25]	0.88	0.69	0.56	0.91
Logistic Regression[28]	0.89	0.74	0.64	0.87
Support Vector Machine[26]	0.90	0.77	0.71	0.85
Adaboost Classifier[27]	0.88	0.73	0.66	0.82
Gradient Boosting Classifier	0.87	0.67	0.52	0.95

Table 5: Hyperparameter Tuned Model Results

	Accuracy	**F1 Score**	**Recall**	**Precision**
Logistic Regression w/GridsearchCV[28]	0.92	0.82	0.77	0.86
Random Forest w/GridsearchCV[25]	0.90	0.73	0.59	0.96
Support Vector Machine w/GridsearchCV[26]	0.92	0.83	0.78	0.88

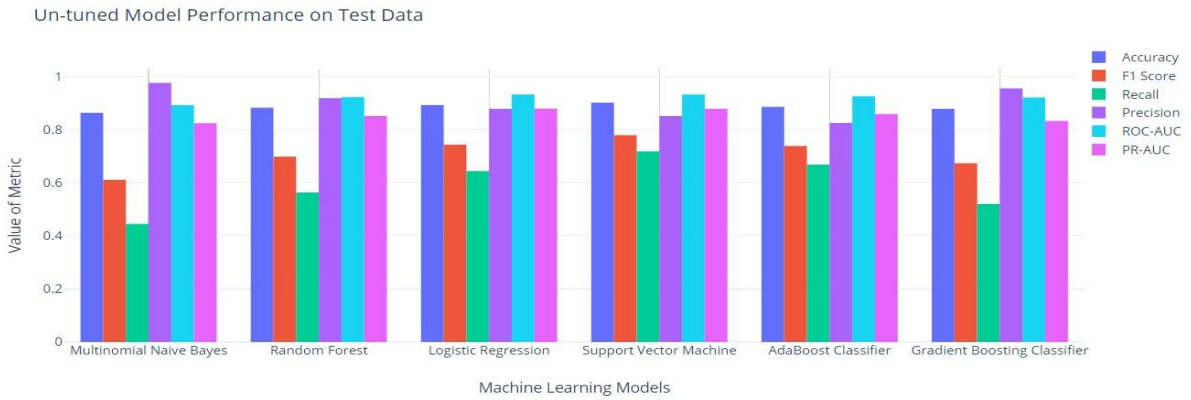


Figure 8: Untuned Model Performance

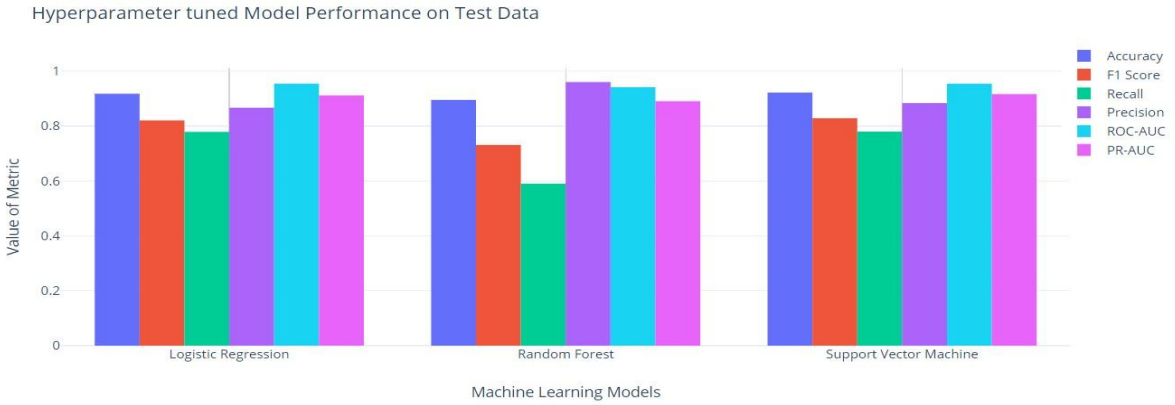


Figure 9: Hyperparameter Tuned Model Performance

In terms of F1-score, accuracy, and recall, a comparison of the results in Tables 3 and 4 revealed that the hyperparameter tweaked models performed much better than the basic models. The greatest F1 score was 0.820634 for the LR[28] approach using GridsearchCV, with recall and precision values of 0.778976 and 0.867000, respectively. Similar to how the SVM model with GridsearchCV worked well, with an F1 score of 0.828640, recall of 0.77987, and accuracy of 0.883910, it likewise performed well. Our findings highlight the significance of hyperparameter optimization in hate speech identification as it may greatly enhance model performance.

The performance of the base models, on the other hand, varied. With an F1 score of 0.611111, the multinominal Naive Bayes model performed poorly in terms of detecting instances of hate speech. The RF[25] model fared better, earning an F1 score of 0.698413, but its recall was still just 0.563342, which was not particularly high. The gradient boosting classifier demonstrated the trade-off between accuracy and recall by having the highest precision (0.955446), but also the lowest recall (0.520216).

Overall, our findings highlight the utility of hyperparameter tweaking in enhancing model performance as well as the necessity of prioritizing F1 score, precision, and recall above accuracy in hate speech identification.

CHAPTER 8

CONCLUSION

The experimentation leads to the conclusion that detecting hateful speech is a difficult undertaking, especially when working with unbalanced datasets. A dataset comprising about 22k non-hate tweets and 8k hate tweets was produced for the study by combining hate and non-hate tweets.

Using default settings, the performance of six binary classification models was assessed. It was found that the SVM[26] model, which had an accuracy of 90.19%, was the most accurate, followed by Logistic Regression, which had an accuracy of 89.28%. It was pointed out, nonetheless, that accuracy is not the ideal statistic to assess a model's performance with an unbalanced dataset.

The F1 Score, Recall, and Precision results indicated that the SVM model performed superior to the other models. The model obtained F1 scores of 0.779, 0.718 for recall, and 0.851 for precision. Also, it was discovered that the SVM model's performance greatly increased following hyperparameter adjustment, with an F1 Score of 0.829, Recall of 0.780, and Precision of 0.884.

In conclusion, the study contends that SVM , particularly when working with unbalanced datasets, is a viable model for the detection of hateful speech on social media. While assessing the model's performance on an unbalanced dataset, it is critical to place an emphasis on measures like F1 Score, Recall, and Precision rather than accuracy.

CHAPTER 9

FUTURE SCOPE

The comparative study of machine learning models for hate speech detection opens up several avenues for future research and development. Here are some potential areas of focus:

1. Model Ensembling: Investigating the effectiveness of ensembling techniques, such as model stacking, bagging, or boosting, to combine the predictions of multiple machine learning models. Ensembling has the potential to improve overall performance by leveraging the strengths of different models and mitigating their weaknesses.

2. Transfer Learning: Exploring the application of transfer learning techniques to hate speech detection. Pretrained models, such as BERT or GPT, trained on large-scale language tasks, can be fine-tuned on hate speech datasets. This approach can help leverage the knowledge learned from general language understanding tasks and adapt it to hate speech detection, potentially improving performance and reducing the need for large annotated datasets.

3. Multimodal Approaches: Investigating the integration of textual information with other modalities like images, audio, or video in hate speech detection. Hate speech often extends beyond text, and considering multiple modalities can provide richer context and improve the accuracy of detection.

4. Online Learning: Developing machine learning models that can continuously learn and adapt to evolving hate speech patterns in real-time. Hate speech evolves quickly, and models that can adapt to changing trends and emerging forms of abusive language can be valuable in maintaining effective detection systems.

5. User-level Analysis: Moving beyond document-level analysis and focusing on user-level analysis to capture patterns of behavior and identify repeat offenders. Understanding the behavior of individuals and their interaction patterns can provide valuable insights for building more comprehensive and effective hate speech detection systems.

6. Addressing Bias and Fairness: Investigating methods to ensure fairness and mitigate biases in hate speech detection models. Models should be evaluated for potential biases towards certain demographic groups and efforts should be made to develop fair and unbiased detection systems.

7. Multilingual and Cross-cultural Analysis: Extending the study to include hate speech detection in multiple languages and across different cultural contexts. Hate speech varies across languages and cultures, and developing models that can effectively detect hate speech in diverse contexts is essential for global applications.

By exploring these future directions, the comparative study of machine learning models for hate speech detection can contribute to the development of more accurate, robust, and inclusive systems that can effectively combat hate speech and promote a safer online environment.

References

1. Waseem, Z., & Hovy, D. (2016). Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. North American Chapter of the Association for Computational Linguistics.

2. Zeerak Waseem. 2016. Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter. In Proceedings of the First Workshop on NLP and Computational Social Science, pages 138–142, Austin, Texas. Association for Computational Linguistics.

3. Davidson, T., Warmsley, D., Macy, M.W., & Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. International Conference on Web and Social Media.

4. Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017). Deep Learning for Hate Speech Detection in Tweets. Proceedings of the 26th International Conference on World Wide Web Companion.

5. Gao, L., & Huang, R. (2017). Detecting Online Hate Speech Using Context Aware Models. Recent Advances in Natural Language Processing.

6. Kumar A, Jaiswal A. Systematic literature review of sentiment analysis on Twitter using soft computing techniques. Concurrency and Computation: Practice and Experience. 2020 Jan 10;32(1):e5107.

7. Poletto F, Basile V, Sanguinetti M, Bosco C, Patti V. Resources and benchmark corpora for hate speech detection: a systematic review. Language Resources and Evaluation. 2020 Sep 30:1-47.

8. Anam MC, Hafiz M. Surat Edaran Kapolri Tentang Penanganan Ujaran Kebencian (Hate Speech) dalam Kerangka Hak Asasi Manusia. Jurnal Keamanan Nasional. 2015 Dec 28;1(3):341-64.

9. Nobata C, Tetreault J, Thomas A, Mehdad Y, Chang Y. Abusive language detection in online user content. InProceedings of the 25th international conference on world wide web 2016 Apr 11 (pp. 145-153).

10. Hayaty M, Adi S, Hartanto AD. Lexicon-Based Indonesian Local Language Abusive Words Dictionary to Detect Hate Speech in Social Media. Journal of Information Systems Engineering and Business Intelligence. 2020 Apr 27;6(1):9-17.

11. Saha, P., Mathew, B., Goyal, P., & Mukherjee, A. (2018). Hateminers : Detecting Hate speech against Women. ArXiv, abs/1812.06700.

12. Younghun Lee, Seunghyun Yoon, and Kyomin Jung. 2018. Comparative Studies of Detecting Abusive Language on Twitter. In Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), pages 101–106, Brussels, Belgium. Association for Computational Linguistics.

13. Mozafari, M., Farahbakhsh, R., Crespi, N. (2020). A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media. In: Cherifi, H., Gaito, S., Mendes, J., Moro, E., Rocha, L. (eds) Complex Networks and Their Applications VIII. COMPLEX NETWORKS 2019. Studies in Computational Intelligence, vol 881. Springer, Cham.

14. Binny, Mathew., Punyajoy, Saha., Seid, Muhie, Yimam., Chris, Biemann., Pawan, Goyal., Animesh, Mukherjee. (2020). HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection. arXiv: Computation and Language,
15. Aluru, S.S., Mathew, B., Saha, P., & Mukherjee, A. (2020). Deep Learning Models for Multilingual Hate Speech Detection. ArXiv, abs/2004.06465.

16. Caselli, T., Basile, V., Mitrovic, J., & Granitzer, M. (2021). HateBERT: Retraining BERT for Abusive Language Detection in English. ArXiv, abs/2010.12472.

17. Dutta, S., Caur, S., Chakrabarti, S., & Chakraborty, T. (2022). Semi-supervised Stance Detection of Tweets Via Distant Network Supervision. Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining.

18. Shaikh, S. and S.M. Doudpotta, Aspects Based Opinion Mining for Teacher and Course Evaluation. Sukkur IBA Journal of Computing and Mathematical Sciences, 2019. 3(1): p. 34-43.

19. Ramos, J. Using tf-idf to determine word relevance in document queries. in Proceedings of the first instructional conference on machine learning. 2003. Piscataway, NJ.

20. Founta, A., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., Vakali, A., Sirivianos, M. and Kourtellis, N., 2018. Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior

21. Mandl, T., Jain, R., Illa, M., Zampieri, M., & Bhardwaj, A. (2019). Overview of the HASOC track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages. In Working Notes Proceedings of the Conference and Labs of the Evaluation Forum (CLEF) (Vol. 2380). CEUR-WS.org.

22. Mandl, T., Goyal, P., Jain, R., Illa, M., Zampieri, M., & Bhardwaj, A. (2020). Overview of the HASOC track at FIRE 2020: Hate Speech and Offensive Content Identification in Indo-European Languages. In Working Notes Proceedings of the Conference and Labs of the Evaluation Forum (CLEF) (Vol. 2696). CEUR-WS.org.

23. Banda, J. M., Tekumalla, R., Abrahao, B., Al-Garadi, M. A., & Lewis, D. (2020). COVID-19 misinformation detection using sparse deep learning models. Applied Intelligence, 1-15.

24. Banda, J. M., Tekumalla, R., Abrahao, B., Al-Garadi, M. A., & Lewis, D. (2020). COVID-19 misinformation detection using sparse deep learning models. Applied Intelligence, 1-15.

25. Xu, B., et al., An Improved Random Forest Classifier for Text Categorization. JCP, 2012. 7(12): p. 2913-2920.

26. Joachims, T. Text categorization with support vector machines: Learning with many relevant features. in European conference on machine learning. 1998. Springer

27. Ying, C., et al., Advance and prospects of AdaBoost algorithm. Acta Automatica Sinica, 2013. 39(6): p. 745-758.

28. Wenando, F.A., T.B. Adji, and I. Ardiyanto, Text classification to detect student level of understanding in prior knowledge activation process. Advanced Science Letters, 2017. 23(3): p. 2285-2287.

PAPER NAME

Final Content of Major_Report v7.docx

WORD COUNT

10864 Words

CHARACTER COUNT

62608 Characters

PAGE COUNT

45 Pages

FILE SIZE

493.9KB

SUBMISSION DATE

May 27, 2023 4:14 PM GMT+5:30

REPORT DATE

May 27, 2023 4:15 PM GMT+5:30

15% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 7% Internet database
- 5% Publications database
- Crossref database
- Crossref Posted Content database
- 12% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 8 words)