

# **DETECTION AND ANALYSIS OF ONLINE HATE SPEECH USING ARTIFICIAL INTELLIGENCE**

**A Thesis Submitted  
In Partial Fulfillment of the Requirements  
for the Degree of**

**DOCTOR OF PHILOSOPHY**  
by

**Anjum**  
(2K18/PHDCO/503)

**Under the Supervision of  
Prof. Rahul Katarya  
Dept. of Computer Science & Engineering**



**Department of Computer Science & Engineering  
DELHI TECHNOLOGICAL UNIVERSITY  
(Formerly Delhi College of Engineering)  
Shahabad Daultapur, Main Bawana Road, Delhi-110042,  
INDIA**

**August 2024**

*Dedicated to*

*My loving husband Rupin and My beloved Parents and Sisters*





**DELHI TECHNOLOGICAL UNIVERSITY**  
(Formerly Delhi College of Engineering)  
Shahabad Daulatpur, Main Bawana Road, Delhi-110042, INDIA

## **CANDIDATE'S DECLARATION**

---

I hereby declare that the thesis entitled “**Detection and Analysis of Online Hate Speech using Artificial Intelligence**” submitted to Delhi Technological University, Delhi, in the partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy in the Department of Computer Science, is an original work and has been done by myself under the supervision of Prof. Rahul Katarya (Supervisor), Department of Computer Science and Engineering, Delhi Technological University, Delhi, India.

The interpretations presented are based on my study and understanding of the original texts. The work reported here has not been submitted to any other institute for the award of any other degree.

A handwritten signature in blue ink, appearing to read 'Anjum', with a horizontal line extending to the right.

**Anjum**  
**(2K18/PHDCO/503)**

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of my knowledge.

**Signature of Supervisor**

A handwritten signature in black ink, consisting of stylized initials and a horizontal line extending to the right.

**Signature of External Examiner**



**DELHI TECHNOLOGICAL UNIVERSITY**  
(Formerly Delhi College of Engineering)  
Shahabad Daultapur, Main Bawana Road, Delhi-110042, INDIA

## **CERTIFICATE**

---

This is to certify that the work incorporated in the thesis entitled **“Detection and Analysis of Online Hate Speech using Artificial Intelligence”** submitted by **Ms. Anjum (2K18/PhDCO/503)** in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy, to the Delhi Technological University, Delhi, India is carried out by the candidate under my supervision and guidance at the Department of Computer Science and Engineering, Delhi Technological University, Delhi, India.

The results embodied in this thesis have not been presented to any other University or Institute for the award of any degree or diploma.

**Prof. Rahul Katarya**  
(Supervisor)

Department of Computer Science & Engineering  
Delhi Technological University, Delhi

Date: /08/2024

Place: New Delhi

## ACKNOWLEDGMENTS

---

I address my sincere thanks to Almighty God for giving me the inner power to complete my thesis and guide me in every step of my life.


It is an immense pleasure to have the opportunity to express my heartiest gratitude to everyone who helped me throughout this research journey. With immense joy and heartfelt gratitude, I would like to extend my indebtedness to my supervisor, Prof. Rahul Katarya (Computer Science & Engineering), for his invaluable guidance, mentorship, encouragement, and patience. During the research, his motivation and encouragement have made me strive to work harder to achieve my goals. His technical expertise, precise suggestions, kind nature, and detailed, timely discussions are wholeheartedly appreciated.

My sincere thank goes to Delhi Technological University for considering my candidature for this course. I am also very thankful to Prof. Prateek Sharma, Vice-Chancellor, Delhi Technological University, Delhi, India, who has been a constant source of enthusiasm. Also, my sincere thanks reciprocate to Dr. Vinod Kumar (HoD, CSE), Prof. Rahul Katarya (Chairperson DRC, CSE) for insightful comments and valuable suggestions. My sincere thanks to all the professors, faculty, researchers, and non-teaching staff of the Department.

I also wish to take this opportunity to thank all my teachers who have taught me and shaped me into the person I am, aggravated me to be an academician, and have directly indirectly made me capable of succeeding in completing this research work. I am deeply thankful to all my colleagues and friends during my journey as a Ph.D. scholar. The engaging discussions, brainstorming sessions, and collaborative teamwork significantly impacted my growth as an independent researcher.

I would like to thank my husband Rupin who always supported me, believed in me and encouraged me in all the challenging times. His unwavering faith in my abilities has been a constant source of strength, uplifting me during moments of doubt. Finally, but most importantly, I would like to express my deepest gratitude to my parents who stood by me like a pillar of strength and always supported me to realize my goals. I will cherish their utmost love and blessings throughout my life.

Date: /08/2024  
Place: Delhi



(Anjum)

# ABSTRACT

---

The ubiquity of social media and the internet has facilitated unprecedented connectivity and information exchange, but it has also given rise to a troubling phenomenon: Online Hate Speech. A comprehensive examination of Online Hate Speech Detection is explored, along with its various dimensions, impacts, and detection methodologies. The pervasive nature of OHS within the digital age, emphasizing its detrimental effects on social cohesion, individual well-being, and democratic discourse is explored. Drawing from scholarly literature and empirical evidence, the urgent need for robust interventions to counter OHS is underscored.

The multifaceted nature of hate speech is elucidated, encompassing various forms of discrimination, prejudice, and incitement to violence. Special attention is paid to the role of anonymity, echo chambers, and algorithmic amplification in perpetuating Online Hate Speech within online ecosystems. Against this backdrop, innovative Artificial Intelligence techniques for Online Hate Speech detection, aiming to empower stakeholders with tools to identify and address hate speech effectively are proposed. Firstly, the HateSwarm feature engineering technique is introduced as a novel approach to feature selection, leveraging bio-inspired algorithms to prioritize salient linguistic cues indicative of hate speech. By enhancing the interpretability and generalizability of hate speech detection models, HateSwarm offers a promising avenue for improving algorithmic performance in real-world settings.

Building upon this foundation, the Hate-Detector model is proposed as a sophisticated tool for multilingual hate speech detection. Integrating state-of-the-art natural language processing techniques, including Bidirectional Encoder Representations from Transformers (BERT) and Multi-Layer Perceptron (MLP) architecture, HateDetector demonstrates high accuracy in identifying hate speech across diverse linguistic contexts. Through rigorous validation on annotated datasets in multiple languages, the model showcases its effectiveness in addressing the challenges of linguistic variation and cultural specificity inherent in hate speech detection.

In parallel, the scarcity of standardized multilingual hate speech datasets is addressed introducing an innovative methodology for dataset construction. Leveraging advanced techniques such as BERT embeddings, clustering, and topic modeling, this approach facilitates the systematic compilation of annotated hate speech data across languages and platforms, laying the groundwork for more robust hate speech detection models with broader applicability.

---

Furthermore, a hybrid framework, integrating Transfer Learning with the Text-to-Text Transfer Transformer (T5) and Long Short-Term Memory (LSTM) models, to enhance hate speech classification accuracy. By leveraging the strengths of both models within a unified framework, this approach enables nuanced analysis and understanding of hate speech across diverse linguistic and cultural contexts, representing a significant advancement in hate speech detection methodologies.

In conclusion, a multifaceted exploration of Online Hate Speech, combining theoretical insights with practical applications to address one of the most pressing challenges of the digital age is focused. By advancing AI techniques, constructing multilingual datasets, and proposing innovative detection frameworks, this research contributes to the ongoing efforts to combat OHS and foster inclusive online communities.

## LIST OF ABBREVIATIONS

---

<b>AI</b>	Artificial Intelligence
<b>ANN</b>	Artificial Neural Network
<b>API</b>	Application Programming Interface
<b>BERT</b>	Bidirectional Encoder Representations from Transformers
<b>BOW</b>	Bag-of-Words
<b>BPEE</b>	Byte Pair Encodings Embedding
<b>CNN</b>	Convolutional neural networks
<b>DL</b>	Deep Learning
<b>XAI</b>	Explainable Artificial Intelligence
<b>GA</b>	Genetic Algorithm
<b>GPT 2</b>	Generative Pre-trained Transformer 2
<b>GRU</b>	Gated Recurrent Units
<b>HDBSCAN</b>	Hierarchical Density-Based Spatial Clustering of Applications with Noise
<b>HOT</b>	Hate Offensive Text
<b>HSCF</b>	Hate Speech Classification Framework
<b>HS-CN</b>	Hate Speech-Counter Narrative
<b>IG</b>	Information Gain
<b>KNN</b>	K-Nearest Neighbour
<b>LRC</b>	Logistic Regression Classifier
<b>LSTM</b>	Long Short-Term Memory
<b>MCC</b>	Matthews Correlation Coefficient
<b>ML</b>	Machine Learning
<b>MLM</b>	Masked Language Modeling
<b>MLP</b>	Multi- Layer Perceptron
<b>MSE</b>	Mean Squared Error
<b>NB</b>	Naïve Bayesian
<b>NLP</b>	Natural Language Processing
<b>OHR</b>	Online Hate Research

<b>OHS</b>	Online Hate Speech
<b>OOB</b>	Out-of-Bag
<b>PCT</b>	Profanity Check Technique
<b>PE</b>	Positional Embedding
<b>POS</b>	Part-of-Speech
<b>PSO</b>	Particle Swarm Optimization
<b>RDT</b>	Regression Decision Tree
<b>ReLU</b>	Rectified Linear Unit
<b>RNN</b>	Recurrent Neural Networks
<b>SE</b>	Segment Embedding
<b>SMN</b>	Social Media Network
<b>SVM</b>	Support Vector Machine
<b>TF-IDF</b>	Term Frequency - Inverse Document Frequency
<b>UMAP</b>	Uniform Manifold Approximation and Projection

## LIST OF TABLES

---

<b>Table 2.1</b>	A Detail list of online Hate Speech Datasets.....	18
<b>Table 2.2</b>	Various features used in Online Hate Speech .....	29
<b>Table 2.3</b>	Traditional frameworks of OHS.....	37
<b>Table 2.4</b>	Deep Learning Methods for OHS .....	44
<b>Table 2.5</b>	Comparison of state-of-the-art techniques .....	48
<b>Table 3.1</b>	Test Result without applying the proposed algorithm on baseline models .....	71
<b>Table 3.2</b>	Test results after applying the proposed Hate-Swarm algorithm on baseline models....	72
<b>Table 4.1</b>	The comparative analysis of proposed and existing techniques .....	98
<b>Table 5.1</b>	Keywords and its Score Value .....	106
<b>Table 5.2</b>	Bigram and Unigram Score Value .....	107
<b>Table 5.3</b>	Extracted Tweets from Twitter .....	107
<b>Table 5.4</b>	Tweets extracted for different Languages.....	108
<b>Table 5.5</b>	Sample Tweets and Its Label.....	109
<b>Table 5.6</b>	Annotation for different text data .....	109
<b>Table 6.1</b>	Sample tweets from distinct datasets .....	120
<b>Table 6.2</b>	Classification Accuracy by the Proposed Approach .....	127
<b>Table 6.3</b>	Comparison with the other techniques for the Hindi tweets.....	128
<b>Table 6.4</b>	Comparison with the other techniques for English Tweets .....	128
<b>Table 6.5</b>	PMI for the proposed Approach .....	129
<b>Table 7.1</b>	Tweets for English and Hindi Languages .....	140
<b>Table 7.2</b>	Tweets for different Languages .....	141
<b>Table 7.3</b>	Number of each sample into different categories.....	142
<b>Table 7.4</b>	Sample Test Results for different tweets in different languages .....	144
<b>Table 7.5</b>	Classification Accuracy by the Proposed Approach.....	146
<b>Table 7.6</b>	Comparison with the other techniques for English Tweets .....	147
<b>Table 7.7</b>	Comparison with the other techniques for the Hindi tweets .....	149



## LIST OF FIGURES

---

<b>Figure 1.1</b> Hate Speech content on Twitter .....	1
<b>Figure 1.2</b> Different communities spared hate-speech text.....	3
<b>Figure 1.3</b> Hate Speech Detection using Machine learning and Deep learning .....	4
<b>Figure 2.1</b> Hate Speech content on Twitter.....	12
<b>Figure 2.2</b> Types of Hate Speech on online Social media.....	12
<b>Figure 2.3</b> Taxonomy of Cyber Crime .....	13
<b>Figure 2.4</b> Search and Selection Process.....	14
<b>Figure 2.5</b> Evidence Synthesis for the literature survey .....	14
<b>Figure 2.6</b> Systematic Representation of the Manuscript .....	15
<b>Figure 2.7 (a)</b> Year-wise classification of the referred "related papers".....	16
<b>Figure 2.7 (b)</b> Content-wise distribution of OHS article .....	16
<b>Figure 2.8</b> Traditional Framework For OHS.....	30
<b>Figure 2.9</b> Deep Learning Framework For OHS .....	39
<b>Figure 3.1</b> Proposed Hate Speech Detection Methodology .....	56
<b>Figure 3.2</b> Output of Baseline Approached .....	72
<b>Figure 3.3</b> Proposed approach output .....	72
<b>Figure 4.1</b> Pre-processing of tweets in Twitter datasets by BERT .....	77
<b>Figure 4.2</b> Process flow of mBART for proposed technique .....	78
<b>Figure 4.3</b> Block Diagram of the proposed technique to detect hate speech .....	84
<b>Figure 4.4</b> Systematic representation of Preprocessing of the Twitter Dataset .....	87
<b>Figure 4.5</b> Accuracy result of HateDetetctor technique with previous techniques.....	92
<b>Figure 4.6</b> Recall of the HateDetetctor technique with previous Implemented techniques.....	93
<b>Figure 4.7</b> Precision result of the HateDetetctor technique with previous techniques.....	93
<b>Figure 4.8</b> F1-Score result of HateDetetctor technique with previous techniques.....	94
<b>Figure 4.9</b> Comparison Analysis of Accuracy of HateDetetctor technique with previous Implemented techniques.....	95
<b>Figure 4.10</b> Comparison Analysis of Recall of HateDetetctor technique with previous Implemented techniques .....	95
<b>Figure 4.11</b> Comparison Analysis of Precision of HateDetetctor technique with previous Implemented techniques .....	96
<b>Figure 4.12</b> Comparison Analysis of F1-Score of HateDetetctor technique with previous Implemented techniques .....	96

<b>Figure 5.1</b> Cyberbullying Instances over the different countries in year2022.....	100
<b>Figure 5.2</b> Proposed methodology.....	103
<b>Figure 5.3</b> German Language Hate Speech Text .....	105
<b>Figure 5.4</b> English translation for the above Text .....	105
<b>Figure 6.1</b> Working of BERT Transformer for embedding.....	112
<b>Figure 6.2</b> Prediction with BERT .....	113
<b>Figure 6.3</b> Framework for the proposed approach.....	115
<b>Figure 6.4</b> Keywords and its probabilities for Topic 0 and 1 for Ismalic Text.....	122
<b>Figure 6.5</b> Top keywords for distinct topics for Islamic Text .....	122
<b>Figure 6.6</b> Topics and Clusters for the Islamic Text .....	122
<b>Figure 6.7</b> Documents are grouped in different Topics based on keywords.....	123
<b>Figure 6.8</b> Keywords and its probabilities for Topic 0 and 1.....	124
<b>Figure 6.9</b> Classification of documents as per topics.....	124
<b>Figure 6.10</b> Similarity matrix for various topics .....	125
<b>Figure 6.11</b> Classification of Tweets in different classes .....	125
<b>Figure 6.12</b> Classification Accuracy of the proposed approach .....	126
<b>Figure 6.13</b> Comparisons with the state-of-the-art methods in terms of accuracy.....	128
<b>Figure 6.14</b> Intertopic Distance Map for Hindi tweets.....	129
<b>Figure 6.15</b> Intertopic Distance .....	129
<b>Figure 7.1</b> Architecture of T5 Model.....	133
<b>Figure 7.2</b> Architecture of CNN-LSTM model for feature enhancement .....	135
<b>Figure 7.3</b> Proposed Framework T5-LSTM HSCF.....	135
<b>Figure 7.4</b> First Five Tweets after Cleaning.....	141
<b>Figure 7.5</b> Total percent of each sample Category .....	141
<b>Figure 7.6</b> Various steps involved in training the data.....	142
<b>Figure 7.7</b> Model fitting with T5, LSTM and CNN .....	142
<b>Figure 7.8</b> Similarity Matrix for English Tweets .....	144
<b>Figure 7.9</b> Shows the loss over learning epoch on different cycles .....	145
<b>Figure 7.10</b> Performance Measures of the proposed approach.....	146
<b>Figure 7.11</b> Comparisons with the state-of-the-art methods for English language Dataset.....	147

# Table of Contents

<b>Candidate declaration</b>	<b>i</b>
<b>Certificate</b>	<b>ii</b>
<b>Acknowledgment</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Table of Contents</b>	<b>vii</b>
<b>List of Abbreviations</b>	<b>x</b>
<b>List of Tables</b>	<b>xii</b>
<b>List of Figures</b>	<b>xiii</b>
<b>CHAPTER 1- INTRODUCTION</b>	<b>1</b>
<b>1.1. Research Gaps.....</b>	<b>6</b>
<b>1.2. Motivation.....</b>	<b>7</b>
<b>1.3. Objectives.....</b>	<b>8</b>
<b>1.4. Structure of the Thesis.....</b>	<b>8</b>
<b>CHAPTER 2- LITERATURE REVIEW</b>	<b>11</b>
<b>2.1 Search and Selection Process.....</b>	<b>13</b>
<b>2.2. Review on Online Hate Speech Detection.....</b>	<b>14</b>
2.2.1. Datasets for Online Hate Speech.....	17
2.2.2. Different Types of Datasets.....	24
2.2.3. Techniques for addressing Imbalanced datasets.....	25
2.2.4. Feature Extraction in OHS.....	26
2.2.5. OHS Detection using Machine Learning Algorithms.....	28
2.2.6. OHS DETECTION USING TRADITIONAL DEEP LEARNING-BASED METHODS.....	40
2.2.7 OHS Detection using BERT and LSTM.....	46
2.2.8 Evaluation Metrics for OHS.....	50
<b>CHAPTER 3- Proposed an Efficient Hate-Swarm Algorithm for Classifying Hate Speech on Social Media</b>	<b>53</b>
<b>3.1. Introduction.....</b>	<b>53</b>
<b>3.2. Limitations of the Existing Research.....</b>	<b>55</b>
<b>3.3. Proposed Methodology.....</b>	<b>56</b>
3.3.1 Machine learning Classifier.....	56
3.3.2 Feature Extraction.....	57
3.3.3.HateSwarm: Proposed Feature Selection Technique.....	59
<b>3.4. Implementation and Results.....</b>	<b>68</b>
3.4.1 Dataset Description.....	68
3.4.2. Results and Discussion.....	69

3.4.5. Conclusion.....	74
<b>CHAPTER 4 - A Novel Method for Detection of Online Hate Speech using HateDetector</b>	<b>75</b>
<b>4.1 Introduction.....</b>	<b>75</b>
<b>4.2. Proposed Method.....</b>	<b>77</b>
4.2.1. Bidirectional Encoder Representation of Transformer.....	77
4.2.2 mBART: Multilingual Encoder-Decode.....	78
4.2.3 BOW with N-gram.....	80
4.2.4. Similarity Checker and Log-Likelihood Test.....	81
4.2.5 Profanity Check Technique.....	82
4.2.6 Proposed Methodology.....	82
4.2.7. Data Set.....	90
<b>4.3. Experimental Set up and Implementation.....</b>	<b>91</b>
4.3.1 Performance Evaluation Metrics.....	92
4.3.2. Comparative Analysis with the state-of-the-Art-Methods.....	95
<b>4.4. Conclusion.....</b>	<b>98</b>
<b>CHAPTER 5 - Creation of Generalized multilingual dataset for Hate Speech Analysis</b>	<b>100</b>
<b>5.1 Introduction.....</b>	<b>100</b>
<b>5.2. Proposed Approach.....</b>	<b>103</b>
5.2.1. Text Preprocessing.....	103
5.2.2 Feature Extraction.....	103
5.2.3. Twitter API.....	103
<b>5.3. Data Collection.....</b>	<b>104</b>
5.3.1. Annotating Dataset.....	105
<b>5.4. Implementation and Result.....</b>	<b>105</b>
<b>5.5. Conclusion.....</b>	<b>110</b>
<b>CHAPTER 6 - A Versatile Framework for Hate Speech Detection for Multilingual Datasets based on BERTopic</b>	<b>111</b>
<b>6.1 Introduction.....</b>	<b>111</b>
<b>6.2. Algorithms used for Online Hate speech detection.....</b>	<b>112</b>
6.2.1. Text Pre-processing.....	112
6.2.2. Bidirectional Encoder Representations from Transformers (BERT).....	112
6.2.3. Topic Modelling.....	114
<b>6.3. Proposed Framework.....</b>	<b>114</b>
6.3.1. Pre-trained BERT Encoder.....	114
6.3.2. Hate speech Clustering using BERTopic for Unlabelled Multilingual Datasets.....	115
6.3.3. Hate Speech classification using BERTopic and Support Vector machine for Labelled Datasets.....	117
<b>6.4. Pseudocode of the proposed approach.....</b>	<b>118</b>
<b>6.5. Implementation.....</b>	<b>118</b>
6.5.1 Dataset Used.....	118

6.5.2. Evaluation Metrics.....	121
6.5.3. Process Illustration.....	121
<b>6.6. Results and Discussions.....</b>	<b>126</b>
<b>6.7. Conclusion.....</b>	<b>130</b>
<b>CHAPTER 7 - A Hybrid T5-LSTM Hate Speech Classification Framework for Multilingual Content</b>	<b>132</b>
<b>7.1. Introduction.....</b>	<b>132</b>
<b>7.2. Algorithms used for Hate Speech Detection.....</b>	<b>133</b>
7.2.1. Text Pre-processing.....	133
7.2.2. Transfer Learning using T5.....	134
7.2.3 Long Short – term memory and convolution neural network.....	134
<b>7.3. Proposed Framework Hybrid T5-LSTM Hate Speech Classification Framework (T5-LSTM HSCF).....</b>	<b>136</b>
7.3.1 T5-LSTM hybrid model.....	137
7.3.2. Proposed Pseudocode of the approach.....	138
<b>7.4. Implementation.....</b>	<b>138</b>
7.4.1. Datasets Used.....	138
7.4.2. Steps for Implementation.....	141
<b>7.5.Results and Discussions.....</b>	<b>144</b>
<b>7.6.Conclusion and Future Work.....</b>	<b>149</b>
<b>CHAPTER 8 - CONCLUSION AND FUTURE SCOPE</b>	<b>150</b>
<b>8.1. Research Summary.....</b>	<b>150</b>
<b>8.2. Future Aspects.....</b>	<b>152</b>
<b>REFERENCES</b>	<b>153</b>
<b>AUTHOR BIOGRAPHY.....</b>	<b>179</b>

# CHAPTER 1

## INTRODUCTION

---

The advent of the internet and the subsequent rise of social media platforms have fundamentally transformed the way individuals communicate, interact, and express themselves online. Platforms like Facebook, Twitter, Instagram, and YouTube have become integral parts of modern society, enabling billions of users worldwide to connect with friends, share opinions, and participate in global conversations. However, alongside the benefits of increased connectivity and information sharing, the digital landscape has also witnessed the proliferation of harmful and discriminatory content, commonly known as hate speech. Hate speech refers to any form of communication, whether written, spoken, or visual, that expresses hatred, hostility, or prejudice towards individuals or groups based on characteristics such as race, ethnicity, religion, gender, sexual orientation, disability, or other protected attributes. It can manifest in various forms, ranging from overt threats and derogatory language to subtle forms of discrimination, microaggressions, and dog whistles. When such things happen on social media to create content, blogs, or to exploit someone is called Online Hate Speech (OHS) [1]. The enormous number of posts, comments, and messages on these websites make it extremely difficult to monitor the information that is shared on them, despite the fact that they serve as a platform for individuals to express and exchange their ideas and viewpoints [2]. Moreover, many people tend to use aggressive, unwanted, and hateful language when discussing certain backgrounds, cultures, and other aspects [3].



**Fig.1 1 Hate Speech content on Twitter [4]**

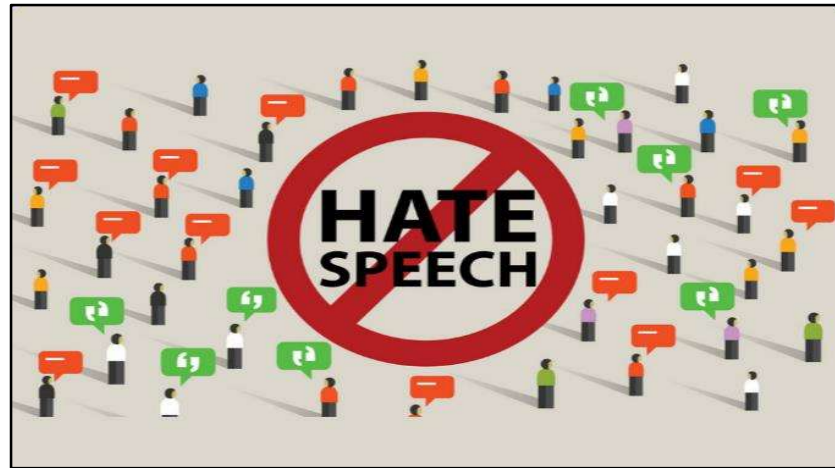
This finally leads to inhumanity which may affect the social media users mentally, as also shown in Fig. 1.1.

People of all ages use social media, and they speak a variety of languages. These languages need to be translated into a common language, such as English, which is commonly referred to as “code.” This is because English is a lingua franca language; more specifically, it is a declarative programming language that is vast, flexible, and standard. As a result, the term “multilingualism” refers to the ability of an individual to speak many languages and the presence of diverse language groups in the same geographic area [5]. Accordingly, in contrast to the general belief, most of the population uses multi-language or two languages at a minimum [6]. The minority population of the World uses a single language which is said to be monolingual. Multilingualism uses several languages as well as code-mixing [7]. Code-mixing is the mashing up of words, sentences, and phrases from two different grammatical systems within the same speech [8] (e.g., Tamil+ English= Tanglish or Hindi+ English = Hinglish). The code-mixing of Hindi, which is the most widely spoken language in South Asia, with English is known as “Hinglish,” a portmanteau derived from the two language’s names. Hinglish differs significantly from its parent languages in syntax, phonetics, grammar, and even punctuation. The accent and sentiments are Hindi, while the vocabulary is made up of several English (Roman) transliterations of Hindi words, as well as a few English terminologies was the mixing of words, phrases, and sentences from two distinct grammatical (sub)systems within the same speech event [9]. With the wide-reaching popularity of social media platforms, code-mixing has emerged as one of the significant linguistic phenomena among multilingual communities that switched languages [10].

The consequences of online hate speech are multifaceted and extend beyond the digital realm to impact real-world dynamics and social relations. Research has shown that exposure to hate speech can have profound effects on individuals' psychological well-being, leading to increased levels of stress, anxiety, and depression, particularly among members of targeted communities. Moreover, hate speech can contribute to the normalization of discriminatory attitudes and behaviors, perpetuating cycles of prejudice and intolerance within society.

Furthermore, hate speech has been implicated in exacerbating social tensions, fueling intergroup conflicts, and even inciting acts of violence and terrorism. Numerous studies have documented the role of online platforms in facilitating the radicalization and mobilization of individuals and groups espousing extremist ideologies. From white supremacist forums to jihadist propaganda channels, the internet provides a fertile ground for the dissemination of extremist narratives and the recruitment of vulnerable individuals into violent movements. Despite the recognition of the harmful effects of hate speech, addressing this phenomenon poses significant challenges for

policymakers, technology companies, and civil society organizations. Traditional methods of content moderation, relying on manual review by human moderators, are often slow, labor-intensive, and prone to errors and biases. Moreover, the sheer volume of content generated on social media platforms makes it infeasible to rely solely on human intervention for effective moderation. Fig 1.2 shows the different communities spared hate –speech text.



**Fig. 1.2. Different communities spared hate–speech text**

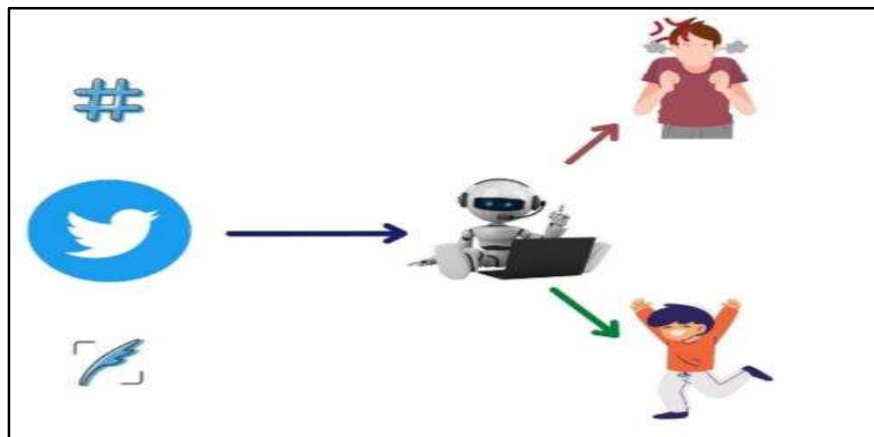
In response to these challenges, there has been a growing interest in developing automated solutions for hate speech detection using machine learning (ML) and deep learning (DL) techniques. By leveraging large datasets of annotated hate speech instances, these systems aim to train algorithms to recognize patterns and linguistic cues indicative of hate speech, enabling faster and more scalable content moderation. However, designing accurate and robust hate speech detection models requires addressing numerous technical, ethical, and societal considerations. Technically, hate speech detection is a complex and multifaceted task due to the diverse forms and expressions of hate speech, as well as the contextual nuances and cultural variations that shape its interpretation. Moreover, hate speech detection models must contend with the dynamic nature of online discourse, where language evolves rapidly, and new forms of hate speech emerge continuously. Additionally, the presence of sarcasm, irony, and other forms of linguistic ambiguity further complicates the task of automated hate speech detection.

Ethically, the development and deployment of hate speech detection systems raise concerns regarding privacy, censorship, and freedom of expression. There is a risk that automated content moderation algorithms may inadvertently suppress legitimate forms of speech or disproportionately target marginalized communities, exacerbating existing power imbalances and inequalities. Furthermore, the reliance on automated systems may obscure the human biases encoded in the training data or introduced



during the algorithmic design process, leading to discriminatory outcomes. Societally, hate speech detection intersects with broader debates about the responsibilities of technology companies, governments, and civil society in combating online harms while upholding fundamental rights and democratic values. Striking the right balance between protecting users from harmful content and preserving the openness and diversity of online discourse requires careful consideration of the trade-offs involved. Moreover, addressing hate speech necessitates a holistic approach that encompasses education, community engagement, and legislative measures in addition to technological interventions.

In light of these considerations, this research seeks to contribute to the ongoing efforts to combat online hate speech by exploring advanced ML and DL approaches tailored for hate speech detection. By leveraging the latest advancements in natural language processing (NLP), neural network architectures, and multimodal learning techniques, this study aims to develop more effective and nuanced models capable of discerning hate speech from legitimate forms of expression in diverse online contexts. Moreover, by critically examining the ethical implications and potential biases inherent in automated hate speech detection systems, this research endeavors to promote the responsible development and deployment of technology in the service of fostering inclusive and respectful online environments. Fig 1.3 shows the Hate Speech Detection using Machine learning and Deep learning.



**Fig.1.3. Hate Speech Detection using Machine learning and Deep learning**

Detecting hate speech using machine learning and deep learning techniques represents a significant advancement in combating the spread of harmful content on online platforms. As digital communication continues to evolve, the sheer volume of user-generated content poses a formidable challenge for manual moderation efforts. Machine learning and deep learning offer automated solutions that can sift through vast amounts of data, identify patterns, and categorize content based on predefined criteria, including the presence of hate speech.

The process of hate speech detection typically begins with the collection of a labeled dataset comprising examples of hate speech and non-hate speech instances. This dataset may be obtained from various sources, such as social media platforms, forums, news articles, or curated repositories. Human annotators or crowdsourcing methods are often employed to annotate the data, indicating whether each instance contains hate speech or not. Once the dataset is assembled, preprocessing steps are applied to clean and standardize the text, removing irrelevant information, such as punctuation, stopwords, and special characters. Additionally, text normalization techniques, such as stemming or lemmatization, may be applied to reduce variation and ensure consistency in the representation of words. Feature extraction follows, where relevant linguistic features are extracted from the text to represent its underlying characteristics. Features extracted during this process may include word frequencies, n-grams, syntactic structures, semantic embeddings, sentiment scores, or other linguistic attributes. These features serve as input to machine learning algorithms, which learn to map the feature space to the corresponding labels (i.e., hate speech or non-hate speech) through a process of training. Various machine learning algorithms, such as support vector machines, logistic regression, or decision trees, can be employed for this task.

Deep learning techniques, particularly neural networks, have demonstrated remarkable performance in hate speech detection tasks due to their ability to automatically learn hierarchical representations of data. Convolutional neural networks (CNNs) excel at capturing spatial patterns in textual data, while recurrent neural networks (RNNs) are adept at modeling sequential dependencies. Architectures such as long short-term memory (LSTM) and gated recurrent units (GRU) are commonly used variants of RNNs for sequential data processing. Transformer-based architectures, such as BERT and GPT, have emerged as state-of-the-art models for natural language processing tasks, including hate speech detection. To detect hate speech, the existing presentations used a meta-learning approach based on metric-based and optimization-based (MAML and Proto-MAML) methods [9], Sentiment reversal analysis[11], Small-sized Transformer model [12], used several layers of classifiers, RGWE method for sentiment analysis [13], Deep convolution neural network [14], Lateral semantics analysis [15] and so on. The above-said methods quickly adapt and generalize new languages with labeled data points and obtain their objective. However, improvement is needed to analyze the data set given out as output because many techniques use low-resource languages and not high-resource languages, many classifiers to bring out its performance, and some concentrate only on some parts of hate speech. These models leverage self-attention mechanisms to capture global dependencies in text, enabling more effective representation learning. By pretraining on large corpora of text data, transformer models can acquire rich linguistic knowledge, which can be fine-tuned for specific downstream tasks, such as hate speech detection.

However, several challenges persist in hate speech detection, including the imbalance between hate speech and non-hate speech instances in the dataset, the presence of biases in the training data, and the ethical implications of automated content moderation. Addressing these challenges requires careful consideration of fairness, transparency, and accountability in the design and deployment of hate speech detection systems. Furthermore, ongoing research efforts are focused on developing more robust and adaptive models that can generalize across different languages, cultures, and online platforms. Domain adaptation techniques, transfer learning, and cross-lingual approaches are being explored to enhance the generalization performance of hate speech detection models in diverse contexts.

### **1.1. Research Gaps**

Despite considerable advancements in Online Hate Speech Detection, there are still significant research gaps and areas for further exploration.

- **Advancing Hate Speech Detection Techniques:** While supervised machine learning (ML) and deep learning methods have shown promise in hate speech detection, there's a pressing need for refining and optimizing these techniques. Emphasis should be placed on developing sophisticated feature extraction methods and enhancing ML algorithms' performance to bolster the accuracy and dependability of hate speech classification.
- **Tackling Multilingual Challenges:** Hate speech varies across languages, yet existing detection models primarily cater to English content. Addressing this gap requires the development of techniques capable of effectively detecting hate speech in diverse linguistic contexts, including languages with limited resources. Research should explore multilingual approaches and adapt existing models to other languages.
- **Improving Contextual Understanding:** Hate speech detection demands a nuanced grasp of context and subtle linguistic cues. Current models may struggle to differentiate between hate speech, offensive language, and non-hate content, particularly in dynamic online environments. Research efforts should concentrate on enhancing contextual understanding and devising techniques to capture and analyze these nuances effectively.
- **Enhancing Scalability and Adaptability:** Hate speech detection systems must scale and adapt to handle the immense volume of online content. Existing approaches may face challenges in scalability and generalization across platforms and datasets. Research should focus on enhancing the scalability and adaptability of hate speech detection systems, ensuring their efficacy across diverse online environments.
- **Diversifying Feature Representation:** Previous research has predominantly relied on lexicon-based features for hate speech analysis, limiting results' robustness when comprehensive sentence meaning is required. Incorporating knowledge-

based and semantic features alongside lexicon-based ones can bolster model accuracy.

- **Bridging Language Gaps:** Hate speech detection research has predominantly targeted English, leaving many languages, such as Arabic, Indonesian, Italian, Turkish, Swedish, Albanian, and Hinglish, unaddressed. Efforts should expand to include these languages and develop balanced, multilingual datasets to facilitate comprehensive studies.
- **Leveraging Unsupervised Learning:** Harnessing unlabeled data for unsupervised machine learning models can expedite hate speech detection efforts, as manual labeling is time-consuming. Therefore, further exploration of deep learning models is crucial for effectively addressing hate speech challenges.

By addressing these research gaps, on hate speech detection can contribute to advancing the state-of-the-art in the field and providing valuable insights into the development of more effective and ethical hate speech detection systems.

## **1.2. Motivation**

The proliferation of hate speech and toxic content on online platforms poses a significant threat to the safety and inclusivity of digital spaces. It is imperative to develop robust automated systems capable of effectively detecting and mitigating such harmful content. However, accurately distinguishing between hate speech, offensive language, and non-hate content presents a formidable challenge due to the subtle nuances and contextual variations inherent in online communication. Previous research has demonstrated the potential of supervised machine learning models and deep learning techniques in identifying hate speech within social media datasets. These approaches have shown promising results in their ability to automatically classify text based on linguistic features and patterns. Despite these advancements, there remains a need for further research to improve the precision and scalability of hate speech detection systems.

Addressing these challenges requires multifaceted approaches. Firstly, hate speech detection often requires distinguishing between subtle linguistic cues that may overlap with offensive language or legitimate expression. By developing advanced feature extraction techniques and leveraging the latest advancements in ML and deep learning, improved accuracy and granularity of hate speech classification can be attained.

Secondly, the effectiveness of hate speech detection models heavily depends on the selection of informative features and the performance of underlying ML algorithms. This research focuses on refining feature selection methods using innovative approaches such as the HateSwarm algorithm, which combines Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) to identify the most discriminative attributes for hate speech classification.

Furthermore, hate speech manifests differently across languages and online platforms, necessitating adaptable and versatile detection methods. Therefore, the aim is to develop techniques that can effectively detect hate speech in diverse linguistic contexts and across various social media platforms, ensuring comprehensive coverage and inclusivity.

Additionally, scalability and adaptability are critical considerations in hate speech detection systems. The vast volume of online content necessitates systems that can efficiently handle diverse platforms and datasets. However, existing approaches may struggle to scale effectively across different contexts, hindering their widespread applicability and impact. Enhancing the scalability and adaptability of hate speech detection systems is essential to ensure their efficacy in detecting and addressing hate speech across various online platforms and communities.

By addressing these research gaps, this research aims to make significant contributions to the field of hate speech detection and classification. To develop more accurate, efficient, and scalable automated systems capable of identifying and mitigating hate speech in real-time is significant. Ultimately, the prime goal is to contribute to the creation of safer and more inclusive digital environments for users worldwide.

### **1.3. Objectives**

To address the identified research gaps, following objectives are formulated,

- To introduce an algorithm aimed at identifying instances of hate speech circulating online.
- To construct a comprehensive multilingual dataset tailored for the analysis of hate speech across various online platforms.
- To devise a methodology capable of detecting hate speech within multilingual datasets sourced from diverse online platforms.
- To conduct a comparative evaluation of online hate speech detection techniques against existing methodologies to ascertain their relative effectiveness and performance.

### **1.4. Structure of the Thesis**

**Chapter 1** provides an overview of the motivation and research objectives driving the investigation into hate speech detection. It highlights the importance of developing robust algorithms capable of differentiating between hate speech, offensive language, and non-hate content. Furthermore, it underscores the need for techniques that can adapt to diverse linguistic contexts and online platforms to ensure comprehensive coverage in hate speech detection efforts.

**Chapter 2** provides an examination of the detection of online hate speech employing various Artificial Intelligence (AI) techniques. The survey aims to elucidate current trends in identifying online hate speech within the realm of artificial intelligence. Additionally, it offers insights into the recent utilization of machine learning and deep learning algorithms for analysing data pertinent to the proposed research problem.

**Chapter 3** introduces significant advancements in hate speech detection and classification, presenting a novel approach named HateSwarm for identifying and categorizing hate speech online. The proposed algorithm combines modified Particle Swarm Optimization and Genetic Algorithm techniques, enhancing the accuracy of hate speech classification. Moreover, the algorithm prioritizes optimized data inputs, ensuring effective real-time detection and prevention of hate speech dissemination.

**Chapter 4**, a technique called 'HateDetector: Multilingual Hate Speech Detection Technique' is introduced. This innovative approach utilizes Bidirectional Encoder Representations from Transformers coupled with Multi-Layer Perceptron (MLP) to discern the nature of tweets by conducting code conversion and similarity checks, resulting in precise vector representations of tweet content. Furthermore, the sentiment or nature of each tweet, whether hateful or not, is determined using the Profanity Check Technique (PCT), which employs a ReLu activation function alongside a logistic regression classifier (LRC) to categorize resultant vectors and corresponding emojis into neutral or hate speech categories.

**Chapter 5** address the issue of the absence of standardized approaches for developing multilingual hate speech datasets by introducing an innovative methodology. A method that leverages Rapid Automatic Keyword Extraction and the Twitter web API was proposed to automatically identify keywords and extract data from diverse web platforms. By employing RAKE-based keyword extraction, high-scoring keywords are identified and used to search various web platforms for dataset creation, ensuring adaptability across different languages. The proposed methodology is implemented across four distinct languages and validated by two annotators who categorize the text into labeled data. Consequently, a versatile multilingual dataset for hate speech analysis was established across web platforms as the foundation for dataset creation.

**Chapter 6** proposed an approach for accurate prediction across cross-platform that integrates Bidirectional Encoder Representations from Transformers with topic modelling. This hybrid approach enhances topic extraction, text categorization, and topic-driven analysis, promising improved hate speech detection and management.

**Chapter 7**, a robust methodology is established through the analysis of a diverse dataset comprising Twitter posts in both English and Hindi, as well as a multilingual dataset encompassing tweets in five distinct languages. This inclusive approach acknowledges the substantial variations in hate speech across diverse linguistic and cultural contexts. The proposed framework, named Hybrid T5-LSTM Hate Speech

Classification Framework (T5-LSTM HSCF), is designed to accommodate multiple languages, playing a pivotal role in addressing this issue.

**Chapter 8** presents the conclusion that summarizes various methodologies and techniques proposed and analyzed, each contributing valuable insights into the complex landscape of hate speech detection and classification. Also, future work will be essential in further refining hate speech detection methodologies and ultimately fostering safer and more inclusive online environments.

## CHAPTER 2

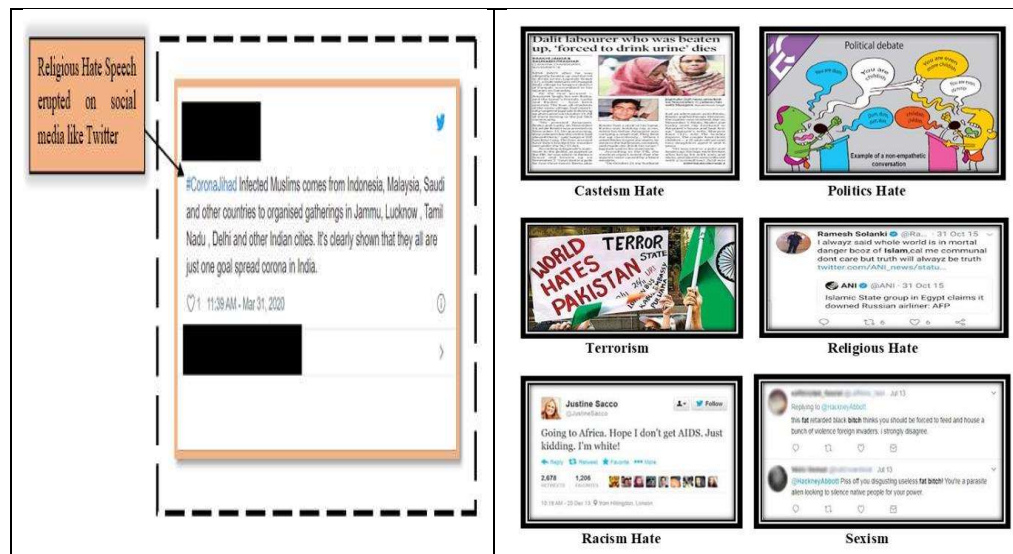
### LITERATURE REVIEW

With the advent of social media and the internet, Online Hate Speech (OHS) and toxicity were present on every social networking website in the form of images, text, and videos. With the recent advantage of mobile computing and the internet, social media provided a platform to share views and exchange information from anywhere, anytime. Social media played an essential role in the origin of online hate speech. On sites like Facebook, Instagram, and Twitter, users could hide their identity or could bully or use toxic thoughts without being noticed. The anonymity of the user on these social platforms provided the user the ability to conceal their identity and say and do whatever atrocious they wanted [16]. Hate speech in different ways was observed. The author defined hate speech as "The use of harsh and abusive words on online platforms to propagate immoral ideas such as communal or political polarity is called Online Hate Speech" [17]. "The speech which uses offensive and hateful language to target specific characteristics of a person or a community is found to be hate speech" [18]. The author defined hate speech as when insulting and derogatory language was used to target certain people with the intention to humiliate them or condescend them [19]. Hate speech was an expression that vilified and disparaged a group of people or a person based on the congregation in a social group recognized by attributes such as mental disability, race, religion, sexual orientation, or gender inequality and others [20]. Typically, hate speech promoted malevolent stereotypes and encouraged savagery against people or a group. With this concept, it was assumed that "hate speech is any speech, which attacks an individual or a group intending to hurt or disrespect based on the identity of a person". For example, in the COVID-19 pandemic, the communal harmony between Hindus and Muslims deteriorated due to a maligning campaign carried out on Twitter shown in Fig. 2.1. which described the religious hate speech content and anti-social elements that existed in our society. Certain applications of detecting hate speech content were in politics, terrorism, casteism, religion. Various types of hate speech content were shown in Fig. 2.2. Most of the work in OHS using artificial intelligence had been done in racism, sexism, and religious areas. Other areas of hate speech were untouched or either classified in the field of hate or non-hate category. Five practical ways to deal with OHS in online social networking platforms like Instagram, Twitter, Facebook were surveyed:

- Report it: Hate speech violated most site's terms of service; people could report it anonymously.
- Block it: Abusive users were blocked
- Do not share it: Forwarding any type of hate speech was wrong because offensive content could be traced back to them.



- Call it out: Understand how other people felt, and find ways to nurture empathy and compassion.
- Learn more: Hate often stemmed from ignorance, so learning from other's experiences.

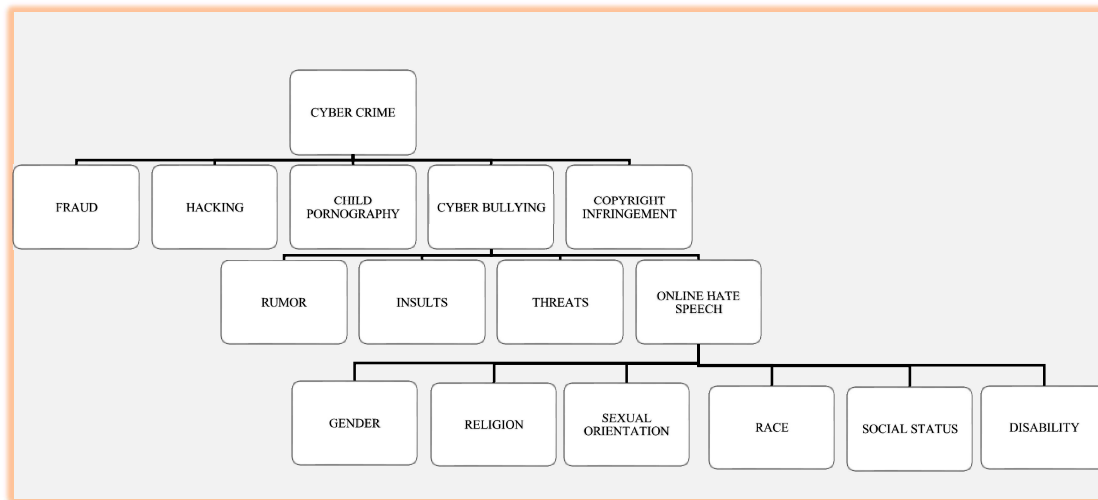


**Fig. 2.1. Hate Speech content on Twitter**    **Fig.2.2. Types of Hate Speech on online Social media**

The origin of OHS was rooted in the class of cybercrime. Therefore, a Taxonomy of cyber-crime was proposed to understand the origin of OHS in a more transparent way. The Hate problem was classified in its various forms, as shown in Fig. 2.3. It was demonstrated that hate speech was a part of the cybercrime and cyberbullying problem. Different authors defined the consequences of OHS, which could include low self-esteem, anxiety, depression, and in some cases, victims could commit suicide. Therefore, the analysis and detection of online hate speech in social media became an area of concern.

In this chapter, a survey of online hate speech identification using different Artificial Intelligence techniques is presented. The review will help to learn about the most recent trends in online hate speech in the field of artificial intelligence. It also includes an overview of recently used machine learning and deep learning algorithms for

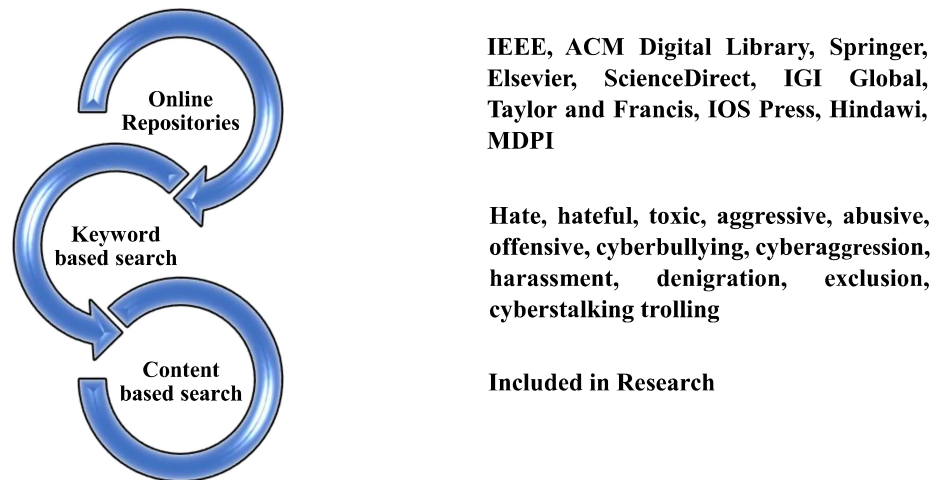
evaluating data used by the proposed research problem.



**Fig. 2.3. Taxonomy of Cyber Crime**

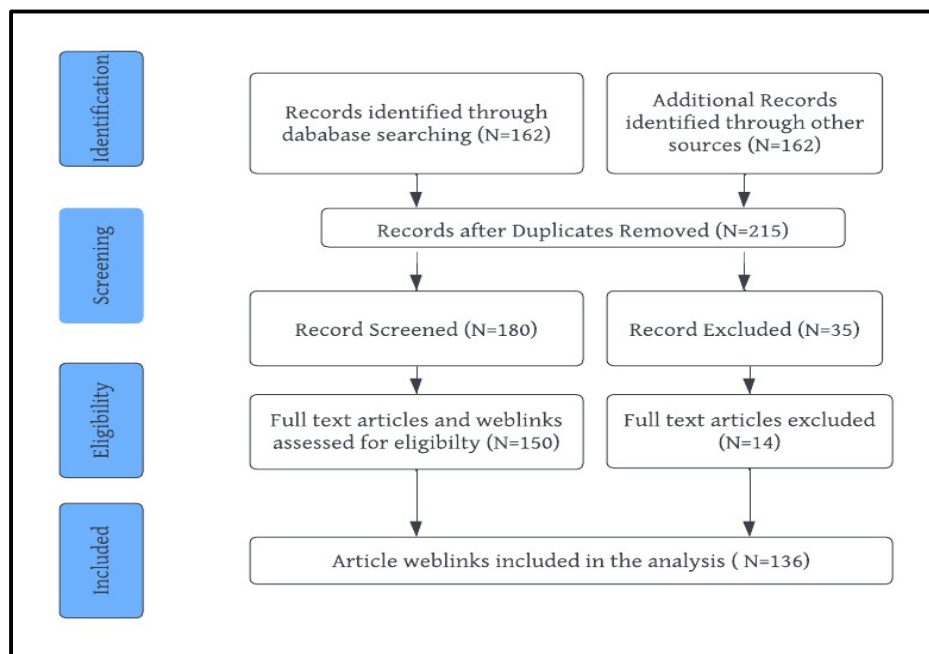
## 2.1 Search and Selection Process

The exploration process for this study revolves around Online Hate speech within the social domain, and it entailed a methodical examination of scholarly articles and specific conference proceedings spanning the years 2000 to 2024. A comprehensive range of online databases, encompassing reputable sources such as Google search engine, ACM Digital Library, IEEE Xplore Digital Library, Springer Link, google scholar, Science Direct, Research Gate, and Wiley Online Library were systematically interrogated to guarantee a comprehensive survey of the existing online hate speech literature. The investigation was specifically focused on four primary domains: hate speech classification, hate speech detection, hate speech detection for social media using machine learning, hate speech classification using deep learning. Pertinent terms were consistently gathered by scanning cited literature to discover the most detailed hate speech and other related surveys. Following that, the terminology "hate," "hateful," "toxic," "aggressive," "abusive," "offensive," and "damaging speeches," as well as "cyberbullying," "cyberaggression," "flaming," "harassment," "denigration," "outing," "trickery," "exclusion," "cyberstalking," "flooding," and "trolling" was coined. The proposed term, "online hate speech," was utilized to refer to the combination of all these concepts in the survey's remaining questions (abbreviated to OHS). Papers which had "hate speech," "cyberbullying", "OHS detection using deep learning", "toxicity in online social media", "OHS detection using machine learning," and "OHS detection using natural language processing" were also included, using them as the search keywords. The search procedure is depicted in Fig. 2.4.



**Fig. 2.4. Search and selection Process**

Approximately 200 research papers were found, and the most relevant 136 papers suitable for this research were shortlisted from the above set. The complete search methodology using the PRISMA diagram is depicted in Fig. 2.5 [21].

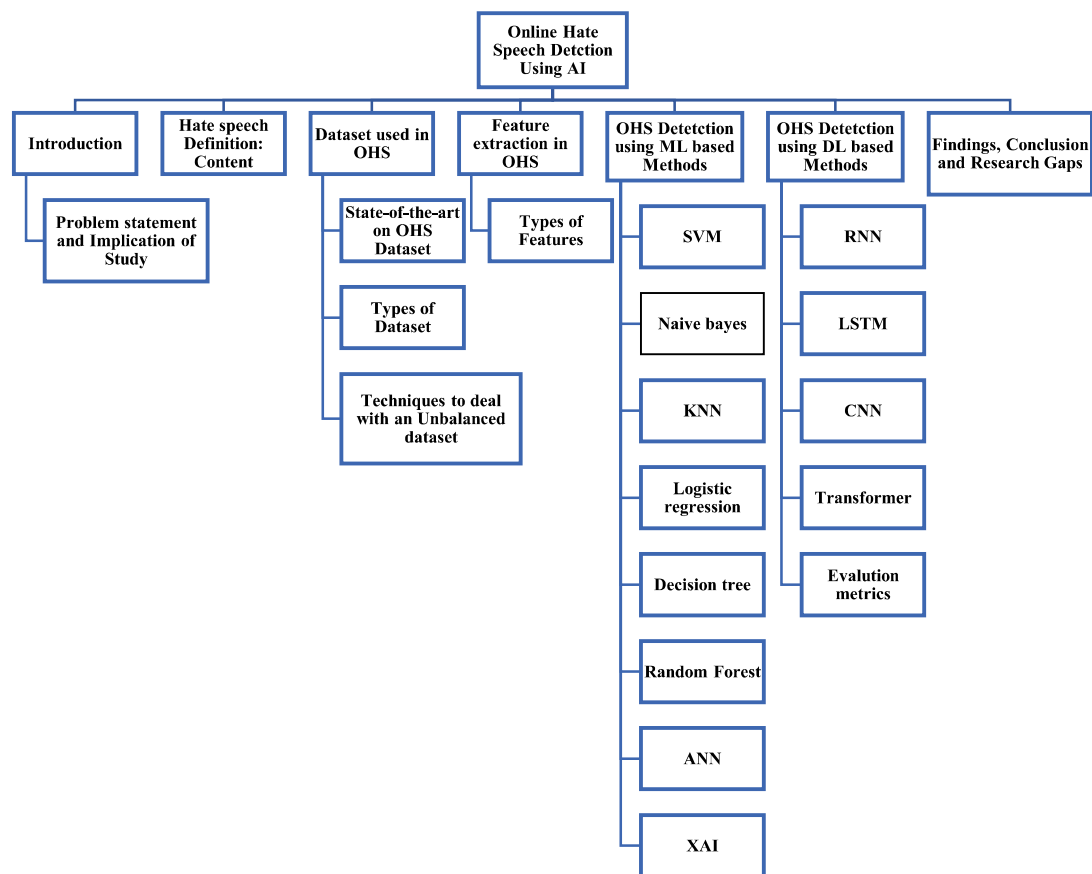


**Fig. 2.5. Evidence Synthesis for the literature survey**

## 2.2. Review on Online Hate Speech Detection

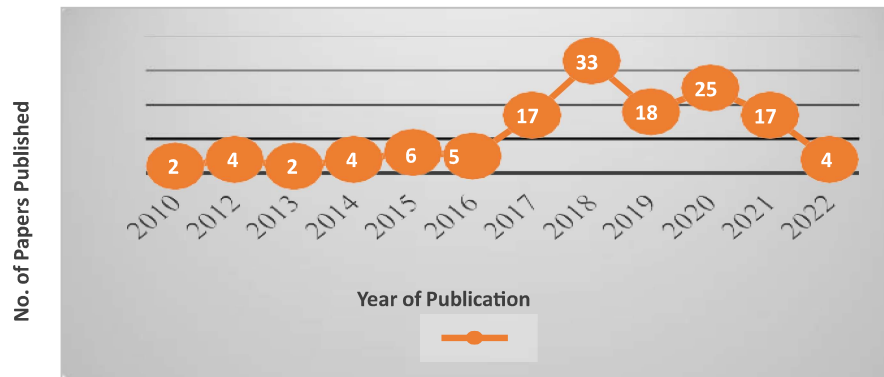
A broad perspective of online hate speech detection and analysis of toxicity detection was reviewed and presented in Fig. 2.6. The year-wise classification of online hate speech was shown in Fig. 2.7(a), and the content-wise distribution of the

referred articles was presented in Fig. 2.7 (b). It was inferred from Fig. 2.7(a) that hate speech had been a focus area (computer science and engineering) from 2016 onwards and had become a popular research area among researchers. Additionally, Fig. 2.7(b) indicated that only four survey papers had been published on Online Hate Speech (OHS) as a subject of research in computer science [22], [23].

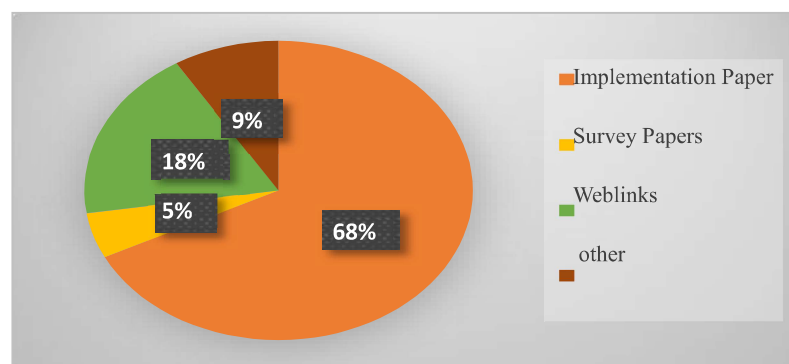


**Fig. 2.6. Systematic Representation of the Review**

backgrounds. So, in this phase, only 15 relevant papers were selected from them, which were required for the problem statement. Furthermore, only the relevant searches concerning the research problem were included. A total of 136 articles and weblinks were selected for this survey. In recent years, there has been a limited publication of survey papers in the domain of OHS utilizing artificial intelligence techniques. The study of OHS is presented by the authors of the papers[23], [24] These works primarily focus on the concept of online hate speech, techniques, features, and datasets published in the area of OHS.



**Fig. 2.7. (a) Year-wise classification of the referred "related papers"**



**Fig. 2.7(b) Content-wise distribution of OHS article**

In previous work, the basic definition of hate speech is established by the authors, considering the various connotations and concepts under which this phenomenon may occur. A comparative analysis of the resources available for hate speech research and pre-existing research from a computer science perspective is provided. The prime focus was on traditional machine learning approaches. Similarly, a concise overview of hate speech using Natural language processing techniques is presented by Fortuna et.al.[23]. Different studies on online hate speech were compared from a natural language processing perspective. The review focused on comparing various types of features used for hate speech classification, including basic syntactic features, character-level features, sentiment features, and more. It argued that relying solely on text-based features may not be accurate enough, and researchers should also consider multimodal and meta-information features for more precise results and judgments. The paper also addressed the issue of the lack of publicly available resources such as datasets. In another work, a meta-analysis of cyberbullying papers using soft computing techniques was presented. Salminen et.al. aimed to map different themes, concepts, stakeholders, and research hotspots in the field of Online Hate Research. Based on this analysis, trends and patterns in OHR, such as which countries invest more in it and how the focus of the field

changes over time, were deduced [21].

In this chapter, hate speech problem is explored in deep using AI techniques. New conceptual elements crucial for autonomous detection tasks were brought to light, such as various datasets used, various kinds of features, and models affecting the outcomes. It also identified deficiencies in the way detection tasks are currently designed, notably in terms of accounting for context and individual subjectivity. The proposed review overcomes the shortcomings of existing surveys by providing limitations of existing techniques and a systematic review of the online hate speech problem.

### **2.2.1. Datasets for Online Hate Speech**

Input data is deemed crucial in machine learning, thus utilizing relevant and accurately annotated data is imperative. In this work, datasets were collected from various reliable sources, and a comprehensive analysis of datasets are detailed in Table 2.1. Numerous researchers have utilized various types of hate speech datasets categorized by language, race, ethnicity, etc. Most of these datasets are accessible on the GitHub website. To gather data from Twitter, many researchers utilized Twitter's Streaming API as a data source for hate speech analysis, providing free access to 1% of all data. The collected data consistently includes metadata and is downloaded in JSON format, requiring conversion into a CSV file. The author presented an unbalanced 16k annotated dataset collected from Twitter, categorized as racist, sexist, or neither [22]. In another work, a Facebook crawler was employed to retrieve comments from Facebook posts, and five volunteered students annotated 6502 comments as no hate, strong hate, or weak hate[16]. Varade et.al.utilized Tumblr search APIs to obtain data from Tumblr, with two to three experienced annotators performing annotation of 2456 posts as racist, radicalized, or unknown [17]. The HatEval dataset is available from the Collab website[18]. Whisper, an anonymous app that does not store old data, was utilized by Davidson et.al. to collect real-time data using a distributed web crawler [19]. Most authors assessed the quality of the dataset using kappa and Interrater agreement. Cohen kappa is a statistical measure of inter-rater agreement concerning the agreement between two raters for categorical items. For instance, if a group of people is evaluated independently by two or more raters to determine their job suitability, Cohen kappa measures the agreement between them [20]. Table 2.1. also contains relevant details of the provided datasets. All accessible datasets are comprehensively presented, providing additional information on hate speech databases.

Table 2.1. A Detail list of online Hate Speech Datasets

Authors	Source	Language	Size	Summary
[25]	GitHub	English	6655 Tweet	The dataset is annotated by only 1
[26]	[by zeerak W] <sup>1</sup>		Dataset Distribution:	Expert and three amateur annotators.
[27]			NAACL_SRW_2016	The author found $k=0.57$ cohen kappa
[28]			None: 11559 Racism: 1969 Sexism: 3378	value for the proposed dataset.
[29]	GitHub [keras-team] <sup>2</sup> And <u>WaCky corpora</u> <sup>3</sup>	English	17567 Comments three classes as strong hate, weak hate, and No hate	Three different annotators are used to annotate the dataset, and the comment is collected from the Facebook pages. To find the level of an agreement, the author computed the Fleiss' kappa( $k=0.19$ ) inter-annotator agreement.

---

<sup>1</sup><https://github.com/ZeerakW/hatespeech/>

<sup>2</sup><https://github.com/keras-team/keras>

<sup>3</sup>[WaCky corpora](#)

[30]	GitHub [ by zeerakW] <sup>4</sup>	English	6909 Tweets Dataset Distribution: NLP+CSS_2016 Neither: 5263 Racism: 207 Sexism: 1269 Both: 52 Link: 118	Twitter API is used to collect data. To find the reliability of the dataset, the author calculated the Fleiss' kappa( $k=0.74$ )
[31] [32]	GitHub [ by T Davidson] <sup>5</sup>	English	25000 Tweets Dataset Distribution: Hate: 1430 Offensive: 19190 Neither: 4163	Three crowdflower workers coded the tweets manually. The author used Flesch Reading Ease scores and Flesch-Kincaid Grade Level to capture the quality of each tweet. The author found a 92% intercoder-agreement score.
[33]	WebScope Dataset <sup>6</sup>	English	2000 Comments Two classes as clean or abusive	All the comments were collected from yahoo's new posts page. The agreement rate of annotated data is 0.922, and Fleiss's Kappa is 0.843.

---

<sup>4</sup> <https://github.com/ZeerakW/hatespeech>, [https://github.com/AkshitaIha/NLP\\_CSS\\_2017](https://github.com/AkshitaIha/NLP_CSS_2017)

<sup>5</sup> <https://github.com/t-davidson/hate-speech-and-offensive-language>

<sup>6</sup> <https://webscope.sandbox.yahoo.com/?guccounter=1>



[34]	Stormfront And crowdflower <sup>7</sup>	English	10568 Two classes as Racist and religion	Sentence level annotator from Stormfront and to find the hate or non-hate they used the crowdflower dataset.
[35]	Tumblr dataset <sup>8</sup>	Arabic	5,569 comments. Dataset Distribution: Hate: 2512 Non_hate: 3057	To annotate the data, they arrange tasks on crowd flower websites only who speak Arabic. Two annotators were participated to annotate the data. The proposed dataset found highly imbalanced, and the inter-annotator agreement and Cohen's Kappa coefficient was 0.95.
[36]	HatEval <sup>9</sup>	English and Spanish	9k Dataset Distribution: English_train Non_hate: 5217 Hate: 3783	The data is collected from the Twitter page. Hate against immigrants and women taken into consideration only.
[37]	Kaggle <sup>10</sup>	English	Dataset Distribution: Neutral: 2898 Insulting: 1049	The data is collected from Twitter.

<sup>7</sup> [Stormfront database, crowdflower, github.com/aitor-garcia-p/hate-speech-dataset, https://data.world/crowd-flower/hate-speech-identification](https://data.mendeley.com/datasets/hd3b6v659v/2)

<sup>8</sup> <https://data.mendeley.com/datasets/hd3b6v659v/2>, [https://github.com/muhaalbadi/Arabic\\_hatespeech](https://github.com/muhaalbadi/Arabic_hatespeech)

<sup>9</sup> <https://competitions.codalab.org/competitions/19935>.

<sup>10</sup> <https://kaggle.com/c/detecting-insults-in-social-commentary>.

[38]	TRAC(Facebook) <sup>11</sup>	English & Hindi	Non-aggressive:69% Overtly aggressive: 16% Covertly aggressive: 16%	The data is collected using Facebook API from Facebook.
[39]	Hatebase database <sup>12</sup>	All Languages	N/A	Hatebase consists of all the hate words that are present in almost all languages. Example: Gender Sexual-orientation, disability class
[40]	HASOC (2019) <sup>13</sup>	Hindi, German and English	5983- Hindi 7005-English 4649-German	The dataset is classified into non-hate, offensive and hate and offensive. Also, the data was collected from Twitter and Facebook websites
[41]	Zenodo <sup>14</sup>	English, German, Spanish, French, and Greek	Approx 90k-English 62k-German 38-Spanish 39k-French 62k-greek	Each dataset contains a tweet id and their annotation. To access the dataset pre-request to the zenodo is required.

---

<sup>11</sup> <http://trac1-dataset.kmiagra.org>

<sup>12</sup> [https://hatebase.org/recent\\_sightings/](https://hatebase.org/recent_sightings/)

<sup>13</sup> <https://hasocfire.github.io/hasoc/2019/dataset.html>

<sup>14</sup> <https://zenodo.org/record/3520152#.XcL0OnUzY5k>.

[42][43]	GitHub <sup>15</sup>	Arabic	Total 6000 text Tweets. 2,526- hate. Which is divided as: [Jews-33% Shia-32% Christians-25% Atheists-24% Muslims-9% Sunnis-7%]	The author collected the data from Twitter. The dataset is classified into hate and non-hate class and contains religious hate speech. The dataset gives an accuracy of 0.79 while experimenting on GRU-based RNN with pre-trained embeddings.
[6]	GitHub <sup>16</sup>	English, French, and Arabic tweets	English-5647 French -4014 Arabic-3353	The dataset was collected based on Directness, Hostility, Target, Group and Annotator attributes. The annotator agreement scores for labelling the dataset are 0.153, 0.244, and 0.202 for English, French, and Arabic, respectively,
[44]	GitHub <sup>17</sup>	Arabic	Total 5,846 tweets Abusive-1728 Normal- 3650 Hate-468	The author collected the data from Twitter, which was Group-directed and Person-directed Tweets.

---

<sup>15</sup> [https://github.com/nuhaalbadi/Arabic\\_hatespeech](https://github.com/nuhaalbadi/Arabic_hatespeech)

<sup>16</sup> [https://github.com/HKUST-KnowComp/MLMA\\_hate\\_speech](https://github.com/HKUST-KnowComp/MLMA_hate_speech)

<sup>17</sup> <https://github.com/Hala-Mulk/L-HSAB-First-Arabic-Levantine-HateSpeech-Dataset>

[45]	File <sup>18</sup>	Arabic	Total 1,100 tweets. Percentage abusive: 0.59	The Twitter platform is used for the dataset collection.
[45]	GitHub <sup>19</sup>	English	Total 33,776 posts. Hate- 14,614 Non-hate- 19,162	The author collected the dataset from the Gap website.

---

<sup>18</sup> <http://alt.qcri.org/~hmubarak/offensive/TweetClassification-Summary.xlsx>

<sup>19</sup> <https://github.com/jing-qian/A-Benchmark-Dataset-for-Learning-to-Intervene-in-Online-Hate-Speech>

Most of the datasets utilized in OHS detection were found to be imbalanced during our investigation. Consequently, oversampling or under-sampling techniques were adopted by the researcher to utilize these datasets for classification. In the subsequent section, various techniques for sampling purposes are discussed along with their respective advantages and disadvantages.

### **2.2.2. Different Types of Datasets**

The several datasets utilized for OHS detection were discussed in section 2.3.1. In supervised machine learning, attention is given to labelled datasets, while unsupervised machine learning focuses on unlabelled datasets. Semi-supervised learning involves a combination of labelled data with a substantial amount of unlabelled data. Labelling data is characterized by labour-intensive and high-cost tasks. Thus, this section explored dataset types, further classified as balanced and unbalanced datasets. It was noted that the majority of datasets listed in Table 2.1. were unbalanced. Consequently, various sampling techniques to balance the datasets for improved results are also explored.

#### **2.2.2.1. Labelled vs Non-Labelled datasets**

Labelled datasets contain both input and output parameters. In one of the studies, charitidis et.al. amassed unlabelled multilingual data from Twitter, later employing a keyword-based approach for data annotation. Subsequently, transfer learning was utilized to cluster the data into hate and non-hate categories. Manually tagging datasets is recognized as a time-consuming and labour-intensive task, prompting interest in the development of tools capable of automatically labelling text. Conversely, in unlabelled data, the output parameter is absent, meaning tags are not associated with the data. Only raw data is available for input into classifiers, which uncover hidden parameters within the dataset [46]. Schmidt et.al. employed both labelled and unlabelled datasets for training and testing the classifier, respectively. Working with unlabelled datasets is generally less costly compared to labelled datasets and is therefore often preferred in unsupervised machine learning approaches [47].

#### **2.2.2.2. Balanced vs Unbalanced Datasets**

In scenarios where datasets are distributed nearly equally among all classes, it is referred to as a balanced dataset. For instance, consider a dataset with two classes, hate and non-hate, consisting of 10k tweets, with 4.5k belonging to hate and 5.5k to non-hate. However, in real-time applications such as medical diagnosis or fraud detection, some level of imbalance is typically observed. If this imbalance is

minimal, the dataset is still considered balanced. However, if the degree of imbalance is significant, it adversely affects the model's performance. When the majority of the dataset is dominated by a single class, it is termed as an imbalanced dataset. For example, out of the total 10k tweets, 2000 are categorized as hate while 8000 are classified as non-hate. Greevy et.al. employed an imbalanced dataset in their study, resulting in the classifier inaccurately assigning new observations to the majority class [48]. Section 2.3.2.3 delves into some commonly utilized sampling algorithms from prior research.

### **2.2.3. Techniques for addressing Imbalanced datasets**

The term "class imbalance problem" in machine learning was coined to describe issues in categorization where data groups were not evenly distributed. The nature of the problem in many application areas often indicated significant skew in the classification process of binary or multi-class classification tasks. There are two approaches to balance the imbalanced datasets: oversampling and under sampling.

#### **2.2.3.1. Under-sampling**

To alleviate the impact of an imbalanced dataset, the under-sampling technique was employed by Watanabe et.al., involving the selection of random samples from the majority class data in the training set to equalize it with the minority class. However, this approach could potentially discard valuable information by reducing samples from the majority class, leading to the loss of relevant data. An extension of the under-sampling strategy involves becoming more discerning with the examples from the majority class that are eliminated [49]. Heuristic approaches are commonly utilized in this process, aiming to identify redundant examples that should be removed or beneficial examples that should be retained [50].

#### **2.2.3.2. Oversampling**

The predictive power of classification systems was diminished by class imbalance. These algorithms were frequently focused on maximizing classification accuracy, a parameter that favoured the dominant class. Despite this, a classifier could achieve high classification accuracy even if it failed to accurately predict even a single instance of a minority class. In this technique, the number of minority class data in the training set was increased. Each point in the minority class was aimed at increasing to balance with the majority class. This approach was much more efficient than under-sampling since under-sampling resulted in the loss of some amount of data. However, oversampling was susceptible to overfitting as it attempted to duplicate examples of the minority class in the training dataset [51].

To address the overfitting issue in oversampling for binary classification, Agarwal et.al. proposed combining the k-means clustering algorithm with SMOTE. The proposed oversampling was able to identify and focus on input space regions where the generation of synthetic data was most effective by leveraging clustering [52].

#### **2.2.4. Feature Extraction in OHS**

Detection of hate speech using machine learning has been a prominent approach. The accuracy of traditional machine learning algorithms was primarily dependent on feature extraction. In this section, all handcrafted features of the machine learning algorithm were discussed. During the feature selection process, as the number of features increased, the threshold value also increased, potentially decreasing the accuracy of the model. Consequently, when provided with extensive feature data, the model became confused as it attempted to process too much information. To address this issue, not all features from the dataset were selected; instead, only specific types of features were utilized, thereby enhancing the model's accuracy. In this section, the types of features that played a crucial role in classifying text as hate or non-hate were discussed.

##### **2.2.4.1. Types of features used**

To classify text into different classes, surface level features were the initial steps to be undertaken. Most authors employed techniques such as Bag-of-Words (BOW), N-gram, char-n-gram, frequency of URL, punctuation, and capitalization. However, BOW and TF-IDF approaches did not retain semantic information due to the risk of overfitting. Waseem et.al. adopted a multi-task learning approach, incorporating various features like BOW, N-gram, and sub-word embeddings [53]. Lynn et.al. utilized the BOW technique to create dictionaries for misogynistic and non-misogynistic language [54]. Furthermore, researchers combined these features with other higher-level features to enhance the model's efficiency [48], [51], [55], [56], [57], [58], [59]. In conclusion, the performance of these features was highly predictive. Good classification results were achieved by most authors using BOW, indicating that predictive words appeared in both training and testing datasets. However, if the dataset comprised small sentences, the model could suffer from data sparsity. To address this issue, the word generalization technique was employed. In order to accomplish this task, clusters of words were considered as additional features, with brown clustering being utilized for this purpose. If new words emerged, they were assigned to one of the clusters based on some degree of similarity [56]. Sreelakshmi et.al. employed word embeddings generated using gensim's word2vec model which were found to be more beneficial compared to simple BOW and TF-IDF [60]. Additionally, a brief survey on Offensive Hate

Speech (OHS) using Natural Language Processing (NLP) was provided by Schmidt et.al.[47]. According to the author, token-level approaches outperformed character-level approaches. Furthermore, it was noted that both word embedding and paragraph embeddings utilized the same underlying concept[49], [61]. Hate speech itself carries negative connotations. If a sentence exhibits negative polarity, it could potentially indicate hate speech or offensive speech. With this assumption in mind, various approaches to sentiment analysis have been explored. Andreou et.al. presents two different approaches: a multi-step approach or a single-step approach [62]. In the multi-step approach, sentiment analysis was initially employed to detect negative polarity, which was then utilized to identify the specific dictionary of hateful words. Conversely, in the single-step approach, features were extracted solely using sentiment analysis, and the text was classified as hate or non-hate based on the polarity of the words [63]. The degree of polarity variation, including highly negative words, also played a significant role in classification. Additionally, the SentiStrength algorithm could be utilized as a feature extraction algorithm to determine the polarity type of the document [64]. Hate speech generally comprises hate words, leading authors to make the general assumption that hate speech contains such negative words (e.g., insulting words, slurs, etc.). In the lexicon approach, attention is given to hateful words by Mariconti et.al. [65]. If a word is found in the dictionary, the classifier predicts it as hate; otherwise, the sentence is classified into the non-hate category. Hatebase is commonly utilized to identify all hate or negative words across various languages. Besides the comprehensive list of hate words, authors focus on specific classes of hate, such as racism, sexism, or ethnic hate-related words. Some authors also attempt to identify hate words through manual inspection tasks. Gitari et.al. utilised a rule-based approach for subjectivity detection and to develop a hate speech classifier. Subjectivity analysis is crucial for sentiment analysis, and multi-perspective question answering is utilized for subjective clues. The bootstrapping algorithm was applied to enhance the lexicon [66]. The author primarily considers datasets related to blogs and the Israel-Palestinian conflict, focusing on race, nationality, and religion target groups. Several authors [22], [49], [51], [67], [68], [69], [70], [71], [72] incorporate the lexical approach along with other features or as baseline features. At times, the classifier frequently experienced confusion between offensive or hate speech. Identifying the semantics of sentences was crucial in hate speech detection performed by Andreou et.al. as language often included both slurs and insults [62]. Therefore, incorporating Part-of-Speech (POS) tagging provided additional semantic information to the classifier as proposed by Watanabe et.al. [49]. However, Agarwal et.al. analysed that POS tagging alone was insufficient to enhance performance; hence, some authors included additional data information such as type dependency relationships [52]. For instance, consider the sentence "Wipe out the Muslims." Here, the term "wipe



out" and "Muslims" exhibit a typed dependency between the two words. The dictionary-based approach was given by Davidson et.al. which lacked effectiveness in context-specific mapping of offensive words. Consequently, to capture opinion, the author employed a domain-based corpus approach [61]. Identifying whether a statement is hate or non-hate was not an easy task, even when linguistic features were employed. Sometimes, background knowledge or domain knowledge was necessary to classify the sentence as proposed by Plaza-Del-Arco et.al [73]. For example, consider the statement: "Put on a wig and lipstick and behave as who you really are." In this statement, hate was directed towards a boy, involving comments about his sexuality (LGBT) or gender. Hence, classifying such statements required a comprehensive understanding of the world. Sharma et.al. introduced some world knowledge using automated reasoning, but this approach necessitated a significant amount of manual coding. Modern social media was widely utilized for disseminating multimodal information, including audio, video, images, and text. Hate speech was not confined solely to textual content; a plethora of other materials circulated on social media platforms daily. To extract information from images, predictive features such as user comments were utilized to discern the semantics of the image [74]. Additionally, Sutejo et.al. explored text and acoustic speech but did not achieve satisfactory results [75].

All features utilized in various research on different algorithms for Offensive Hate Speech (OHS) detection were analyzed. Identifying the best features in traditional machine learning was deemed a crucial task. Therefore, we comprehensively discussed all features in Table 2.2., as employed in previous OHS research. It was found that surface-level features, linguistic features, and lexicon features were the most frequently extracted features, outperforming other existing features when coupled with AI techniques.

### **2.2.5. OHS Detection using Machine Learning Algorithms**

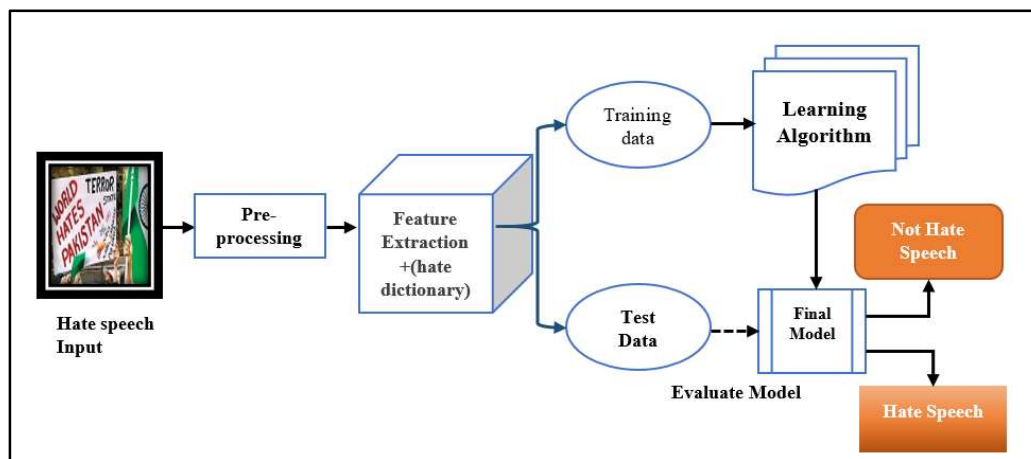
Various machine learning techniques have been adopted to address the issue of Offensive Hate Speech (OHS). The general framework of the OHS detection methodology was illustrated in Fig.2.8. The data underwent initial preprocessing, which involved removing punctuation, tokenization, stopwords, and stemming or lemmatization to render it suitable for mining and feature extraction. Subsequently, features were extracted using various techniques such as Bag-of-Words (BOW), TF-IDF, word embeddings, etc.

**Table 2.2. Various features used in Online Hate Speech**

S.No.	Type of Feature	Classes of features	References Cited
1.	Surface Level	Bag-of-word	[72], [76], [77], [78], [79]
2.		Negation	[48], [80], [81]
3.		unigram	[21], [58], [74], [82]
4.		n-gram	[24], [56]
5.		Frequency of URL mention	[83], [84]
6.		Token Length and Capitalization	[67]
7.		Non- English Words	[85], [86], [87]
8.	Word Generalizations	Set of words (Clustering)	[19], [24], [48]
9.		Word Embeddings	[51], [55], [88]
10.		Tf-Idf	[56], [58], [59]
11.	Sentiment Analysis	Positive and Negative polarity	[49], [89]
12.		Neutral words	[58], [78]
13.	Lexical Resources	General Hate Related terms	[66], [90]]
14.		Contextual Information	[90], [91]
15.	Linguistic Features	n-gram+ POS information	[53], [92], [93]
16.		Dependency Relationships	[94], [95]
17.		Syntactic Feature and Semantic feature	[96]
18.	Knowledge base feature	Heteronormative Context	[88]
19.	Meta Information	Background information about the user of the Post	[97]
20.		No of Post by User	[97]
21.		No of reply by user	[98]
22.		Location	[68]

23.		Correlation between the number of post and hate speech	[49]
24.	Multimodal	Images	[55]
25.	Information	Audio	[55]
26.		Video and Audio Content	[55]

Following preprocessing, features were extracted from the processed data. The subsequent step involved passing the processed data through our trained classifier, which categorized them into positive or negative classes. This classification process facilitated the identification of instances of offensive or non-offensive speech within the dataset.



**Fig. 2.8. Traditional Framework For OHS**

### 2.2.5.1. Support Vector Machine

The support vector machine (SVM) was pioneered by Vladimir Vapnik in the '90s. SVM utilizes the kernel trick to model nonlinear decision boundaries, drawing a decision boundary near the extreme points in the dataset. Consequently, the SVM algorithm essentially functions as a frontier that optimally segregates the two classes. Greevy et.al. employed SVM to detect racist text using different kernel functions on Bag-of-Words (BOW), bigrams, and parts of speech (POS) to identify the most effective technique. The highest accuracy was attained using BOW with the polynomial function, whereas POS performed inferiorly compared to BOW and bigrams [48]. Watanabe et.al. noted that SVM performed exceptionally well on surface-level features and yielded the highest accuracy results in binary classification [49].

In another study, Warner et.al. collected data from Yahoo newsgroup posts and the American Jewish Congress. A template-based strategy was utilized to generate features from the corpus, treating the problem as word-sense disambiguation and employing SVM light classifier with a linear kernel function. However, the proposed results using this classifier were not accurate, and the inclusion of bi-grams and tri-grams degraded the classifier's performance. Additionally, long linguistic patterns were not detected, resulting in low recall and precision values [92].

Ombui et. al. presented an annotation framework for hate speech in tweets collected during the Kenyan election. The framework was developed for extracted text, employing bootstrapping and n-gram techniques to identify hateful tweets from the 394k collected data. Krippendorff's alpha was used for the reliability of annotated tweets. The paper also utilized the duplex theory of hate, featuring passion, distance, and commitment as part of the hate speech framework. Out of 394k tweets, 94% were labelled as ethnic [88]. However, the authenticity of the data, including fake news and propaganda, was not addressed, and the framework was only applicable to short messages. SVM emerged as one of the majorly adopted techniques by researchers due to its effectiveness in various classification tasks [58], [61].

#### 2.2.5.2. Naïve Bayes

Naïve Bayes is a supervised learning algorithm utilized for both binary and multiclass classification problems. It is rooted in the Bayes theorem formulated by Thomas Bayes, operating under the naïve assumption that features are independent of each other. This simplistic yet effective assumption simplifies the algorithm.

The Bayes theorem equation is represented as follows:

$$P(A|B) = (P(B|A) * P(A))/P(B) \quad (2.1)$$

Where:

- $P(A|B)$ : The probability of event A occurring given that event B is true.
- $P(A)$ : Prior probability, representing the probability of event A occurring before event B.
- $P(B)$ : Prior probability, representing the probability of event B occurring before event A.
- $P(B|A)$ : The probability of event B occurring given that event A is true.

In the detection of hate speech, Rodriguez et.al. utilized naïve Bayes by extracting surface-level features and lexicon features. It was found that the voting classifier yielded superior results compared to the lexicon-based approach for classification [99]. Similarly, Diwhu et.al. employed at least three annotators to annotate hate words and compared the results. Standard pre-processing techniques such as TF-IDF and n-

grams were utilized afterward. Naïve Bayes exhibited comparable accuracy to other classifiers [24].

Furthermore, through the use of hard ensemble, Biere et.al. achieved the highest accuracy of 78.3% with naïve Bayes compared to other classifiers on an unbalanced dataset. This highlights the effectiveness of naïve Bayes in addressing classification tasks, particularly in scenarios with imbalanced data [22].

### **2.2.5.3. K-nearest neighbor**

The k-Nearest Neighbor (KNN) algorithm is recognized as one of the simplest and most widely used classification algorithms. This method is applied when data points are categorized into multiple classes, aiming to predict the classification of a new sample point. KNN operates on the fundamental concept of similarity and is particularly suitable for addressing nonlinearly distributed data points, where conventional linear classification methods are inadequate. To determine similarity between data points, KNN typically calculates metrics such as Euclidean distance or Manhattan distance. Subsequently, an object is classified based on the majority vote of its nearest neighbors, with the object being assigned to the class most prevalent among its neighboring points. Rodriguez et.al., Betweenness Centrality was utilized to identify prominent pages on Facebook. only few studies have been conducted in the realm of hate speech detection utilizing KNN, highlighting an area with potential for further exploration and research [99].

### **2.2.5.4. Logistic Regression**

Logistic regression (LR) is a statistical method utilized to address binary classification and multiclass classification problems, where the output variable  $y$  belongs to the set  $\{0,1\}$ . This regression technique estimates the relationship between the dependent and independent variables. Consequently, LR is predominantly applied when the dependent variable or output is in binary or categorical format. Davidson et.al. applied logistic regression utilizing surface-level features, resulting in comparable results. However, limited research has been identified on the utilization of word generalization and knowledge-based features in logistic regression. This underscores an area with potential for further investigation and experimentation in the domain of logistic regression for classification tasks [61].

### **2.2.5.5. Decision Tree**

They are used for both classification and regression tasks. They make decisions by recursively partitioning the input space based on feature values. The various methods used to construct decision tree are explained below.

**a. Entropy:**

Decision Trees use entropy as a measure of impurity in a dataset. The entropy ( $H$ ) is calculated as:

$$H(S) = -p^+ \log_2(p^+) - p^- \log_2(p^-) \quad (2.2)$$

where  $p^+$  and  $p^-$  are the probabilities of positive and negative classes, respectively.

**b. Information Gain:**

Information Gain ( $IG$ ) is used to determine the effectiveness of a feature in reducing entropy. For a dataset  $S$  and a feature  $A$ :

$$IG(S, A) = H(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} H(S_v) \quad (2.3)$$

where  $S_v$  is the subset of  $S$  for which feature  $A$  takes value  $v$ , and  $\text{values}(A)$  are the possible values of feature  $A$ .

**c. Gini Impurity:**

Another impurity measure used in decision trees is Gini Impurity ( $G$ ). For a dataset  $S$ :

$$G(S) = 1 - \sum_{c \in \text{classes}} (P_c)^2 \quad (2.4)$$

Where  $(P_c)$  is the proportion of instances of class  $c$  in  $S$ .

**d. CART Algorithm for Binary Classification:**

The CART algorithm uses Gini Impurity to split the dataset  $S$  into two subsets  $S_{left}$  and  $S_{right}$  based on a feature  $A$  and a threshold  $t$ :

$$G(S, A, t) = \frac{S_{left}}{S} G(S_{left}) + \frac{S_{right}}{S} G(S_{right}) \quad (2.5)$$

The algorithm chooses the split that minimizes  $G(S, A, t)$ .

**e. Regression Decision Tree:**

For regression tasks, the decision tree minimizes the Mean Squared Error (MSE) as the impurity measure. Given dataset  $S$  and target values  $y_i$ .

$$MSE(S) = \frac{1}{|S|} \sum_{i=1}^{|S|} (y_i - y_s)^2 \quad (2.6)$$

where  $y_s$  is the mean target value of  $S$ .

Decision Trees recursively split the dataset based on features and thresholds to create a tree structure that can make predictions for unseen instances. The choice of impurity measure depends on the specific algorithm and task at hand. DT is used by Davidson et.al. and surface-level features were the first choice of the research to use in the classification process (Davidson et al., 2019).

### 2.2.5.6. Random Forests

It is an ensemble learning method that combines multiple decision trees to improve predictive accuracy and control overfitting. Each tree in the ensemble is built on a subset of the training data, and the final prediction is obtained through a voting or averaging process. Various steps of random forest are explained.

#### a. Bootstrapped Dataset:

Random Forest constructs multiple decision trees, and each tree is trained on a bootstrapped subset of the original training data. It involves randomly sampling with replacement, creating a new dataset  $S_i$  for each tree  $i$ .

$$S_i = \text{BootstrapSample}(S) \quad (2.7)$$

#### b. Feature Randomization:

At each node, a random subset of features is used for splitting. If the original dataset has  $m$  features, a subset  $m_{rand}$  is chosen randomly.

$$m_{rand} \leq m \quad (2.8)$$

#### c. Decision Tree Training:

For each bootstrapped dataset  $S_i$ , a decision tree  $T_i$  is trained using feature randomization. The training involves recursively splitting nodes based on the selected features until a stopping criterion is met.

$$T_i = \text{TrainDecisionTree}(S_i) \quad (2.9)$$

#### d. Voting (Classification) or Averaging (Regression):

For classification, the final output is determined through a majority vote. For regression, the final output is the average of the predictions made by individual trees.

$$\text{FinalPrediction} = \frac{1}{N} \sum_{i=1}^{N_{trees}} \text{prediction}(T_i) \quad (2.10)$$

#### e. Out-of-Bag (OOB) Error Estimation:

The performance of the Random Forest can be estimated using out-of-bag samples, which are instances not included in the bootstrapped dataset for each tree. The OOB error is computed by evaluating the predictions on these out-of-bag samples.

$$\text{OOB Error} = \frac{1}{N} \sum_{i=1}^N L(y_i, \text{AveragePrediction}(\{T_j | x_i \notin S_j\})) \quad (2.11)$$

where  $N$  is the number of instances,  $y_i$  is the true label of instance  $i$ , and  $L$  is the loss function.

The ensemble of decision trees was employed by Mariconti et.al. to analyze the video platform for identifying hatred within multimodal data. A maximum accuracy of 0.94% was achieved using a weighted-vote ensemble[65]. Additionally, Wang et.al.

focused on detecting hateful content on Twitter and Whisper. Given that Whisper is an anonymous mobile application, data spanning nearly one year was collected from the Whisper app, alongside a 1% random sample from Twitter, accessible to all users. They introduced a computational method for hate speech detection, involving the division of sentences into four parts: I, Intensity, user intent, and hate target [100]. However, it is important to acknowledge the possibility of biases inherent in data collected from online social networks.

#### **2.2.5.7. Artificial Neural networks**

Artificial neural networks (ANN) were utilized, consisting of interconnected nodes forming structures through directed links. A basic ANN typically comprises only one hidden layer. Within this framework, a perceptron serves as a simple neural network, further categorized into single-layer and multilayer perceptrons, with the latter including hidden layers and networks. Raufi et.al. extracted features were inputted into a simple ANN classifier. Subsequently, a genetic-based approach was employed to detect hate speech in the Albanian language. This approach leveraged the capabilities of artificial neural networks to analyze and classify textual data for hate speech detection [55].

#### **2.2.5.8. Explainable Artificial Intelligence**

Explainable artificial intelligence (XAI) is technology utilized to decode the reasoning behind neural networks and present it in a format understandable by humans [101]. As neural networks grow increasingly complex with a multitude of parameters and feature engineering becomes obsolete, the demand for justifiable deep learning models has become paramount. XAI has gained prominence, particularly in the field of computer vision, with visualizations like class activation maps becoming increasingly popular. Class activation maps are generated by overlaying the features of a layer in a deep neural network (DNN) onto the image being classified. This process highlights the significance a model places on specific regions or pixels within the image. Such visualizations aid data scientists in designing models that prioritize relevant features in decision-making, thereby enhancing the model's reliability. Despite its significance, the adoption of XAI has been limited, although there has been a recent surge in interest. Mathew et.al. introduced a benchmark dataset containing tweets, each labeled with a class (hate, offensive, normal), a target community, and the rationale behind its class labels. The author further demonstrates that models performing well according to traditional metrics such as accuracy, macro F1-score, and AUROC score may not necessarily excel in explainability metrics such as plausibility, comprehensiveness, and sufficiency[101].

In Table 2.3, a comparison was presented regarding various traditional machine learning approaches and their respective advantages and disadvantages.



Table 2.3. Traditional frameworks of OHS

Authors	Approach	Language	Dataset	Merits	Limitation
Raufi et al. [55]	ANN	Albanian	3620 words from Albanian forums	The highest accuracy achieved is 94 %, with a 60-30 spilled.	In the Long Run, many word features will become irrelevant. Their current system is developed on "per word" based detection, where deeper language constructs are not in their scope.
Martins et al [58]	RF, NB, and SVM	English	Davidson and Warmsley.	Finds the best accuracy with SVM compared to NB and RF i.e., 80.56%.	emphasis was on emotional features only. Also, fixed vocabulary was found on hatebase, Semantic features arenot considered.
Sharma et al. [74]	Machine Learning and NLP	English	Data set available on Kaggle.	Real-time tweets are extracted from multiple online sites and created a new dataset.	Prepares only data but does not build a classifier.

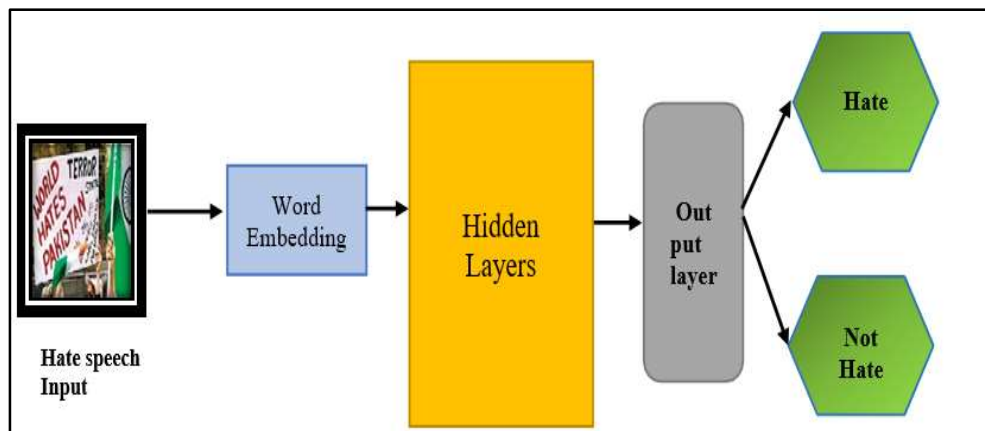
Pelzer et al. [57]	NLP and Automated reasoning	Swedish	Avpixlat and Samha IIsnytt	NLP+AR approach is more easily adapted to other languages.	NLP+AR technique finds very small hateful comments compared to manual inspection.
Davidson et al. [19]	Logistic regression, SVM and Sentiment Lexicon	English	Davidson and Warmsley dataset.	Overall F1 score of 90 is achieved with SVM and LR.	40% of hate speech is misclassified.
Diwu et al [24]	SVM, J48, Naïve Bayes, Random Forest, Random Tree	Turkish	Twitter	Accuracy is in the range of 60 on almost all models.	Complete lexical approach will fail if new vocabulary is observed in the data.
Rodriguez et al [51]	Sentiment Analysis, (VADER) Emotional Analysis (JAMIN), K- means clustering	English	Collected 1000 comments from each page from the Facebook using FB graph API	The author proposes a new way of dealing with hate speech.	The method is not completely automated.

Watanabe et al [49]	Unigram and pattern classification, J48 graft	English	two from Crowdflower and one from GitHub.	An accuracy equal to 87.4% for the binary classification of tweets into offensive and non-offensive and an accuracy equal to 78.4% for the ternary classification of tweets into, hateful, offensive and clean.	Richer dictionary of hate speech patterns can be used for the better classification
Greevy & Smeaton [102]	SVM	English	Yahoo	Polynomial proved to be the most effective kernel function for both BOW and POS.	Computationally expensive, Pos performed worse than Bow and bigram.
Sreelakshmi et al [60]	SVM-radial bias, Random Forest, SVM-linear	Hindi English code mixed data	10000 data from different sources.	It is found that character-level features give more information for code-mixed classification.	Classification of the tweets has been not done on multi-classification.

### 2.2.7. OHS DETECTION USING TRADITIONAL DEEP LEARNING-BASED METHODS

In the realm of machine learning, both traditional methods and deep learning techniques are employed for model training and data classification. In traditional machine learning, features are manually extracted, whereas in deep learning, this manual feature extraction step is bypassed. Instead, data is fed directly into deep learning algorithms like CNNs, which then autonomously predict object classifications. Consequently, deep learning, a subtype of machine learning, directly engages with raw data such as images and typically involves more complex computations. For instance, Fig. 2.9. illustrates how a deep learning model distinguishes between hate speech and non-hate speech text inputs. A key component of deep learning is the deep neural network, characterized by multiple hidden layers that facilitate the extraction of high-level features from the dataset. At each layer, input data undergoes transformations, yielding progressively detailed representations.

Deep learning is often perceived as a "black box" by researchers because it obviates the need for explicit feature engineering. However, up until 2019, relatively little research had been conducted on utilizing deep learning for hate speech detection. This scarcity of research in deep learning may be attributed to factors such as limited labeled data availability and a lack of high-performance GPU resources. Nonetheless, a notable shift towards deep learning occurred in 2020, as evidenced by a majority of research papers focusing on deep learning rather than traditional machine learning methods. Various types of deep learning models employed in prior literature for addressing OHS-related concerns are discussed further.



**Fig.2.9. Deep Learning Framework For OHS**

### 2.2.7.1. Recurrent Neural Network

The sequence of information cannot be captured by Artificial neural network is captures by RNN. RNN, a type of neural network, is utilized for capturing sequence or time-series data information. Variable size input can be taken, providing variable size output and displaying effective performance with time-series data. It is a class of artificial neural networks where connections between nodes form a directed graph, allowing information flow back into previous parts of the network. Consequently, each model in the layers depends on past events, facilitating information persistence. To identify sentences as hate speech or not, tests were conducted with RNN, data partitioning, epoch, learning rate, and batch size. All these parameters had an impact on the system's performance. UTFPR models were employed by Paetzold et.al. to process the text. Subsequently, character embeddings were fed into the RNN layers. The proposed system is based on compositional RNN. Even when the input data is noisy, the proposed model remains robust, but the dataset used to feed the RNN is very small, and the classifier's performance may be affected if a large dataset is utilized [103]. Social media platforms such as Facebook, Twitter, and Instagram are increasingly serving as ubiquitous platforms for individuals to share and express their opinions [99]. Online social networks, particularly Twitter, wield considerable influence over a person's image, as highlighted by Saksesi et.al. who employed an RNN DL-based approach to detect hate speech text in Twitter data. Subsequently, 1235 posts were analyzed using case folding, tokenization, cleansing, and stemming. The data were collected from Twitter accounts via the Twitter API. Using RN and LSTM (Long Short-Term Memory), not only single data but also entire sequences of data can be processed simultaneously. Word2vec was utilized to convert sentences into vector values or to discern semantic meaning. Testing the data with epoch resulted in high precision of 91% and recall of 90%, with an accuracy of 91% [80]. Pitsilis et.al. [117] presents machine learning with a hybrid NLP approach, employing killer NLP with ensemble deep learning to analyze the data, resulting in a system accuracy of 98.71%. Addressing the issue of identifying speech promoting religious hatred on Arabic Twitter, Albadi et.al. created an Arabic dataset of 6000 tweets annotated for hate speech detection and developed various classification models using a lexicon-based, n-gram, and deep learning-based approach. However, GRUs were utilized instead of LSTMs due to their faster training and potential to achieve better performance on datasets with a limited number of training examples. The GRU (gated recurrent unit)-based RNN model yielded the best results for the evaluation metrics [104]. Lee et.al. showcases how psychologists have explored the connection between hate and personality. The author employed a text-mining strategy that fully automates the personality inference process. A deep learning algorithm called PERSONA was developed to identify hate speech online [105].

### 2.2.7.2 Long Short-Term Memory

LSTMs, a modified version of RNNs, are utilized for learning long-term dependencies, particularly in time series analysis. They possess the capability to process diverse data types such as images, speech, and video. Comprising gates including Input, Output, and Forget, LSTMs manage the receipt, output, and decision-making regarding information retention, respectively. In RNNs, the challenge of vanishing gradients arises as errors are propagated back through multiple layers, hindering the establishment of long-term dependencies. LSTMs effectively address this issue, resulting in significantly improved accuracy compared to RNNs. For OHS classification, the LSTM classifier was employed by Sazany et.al. alongside the FastText library, yielding comparable results to sentiment analysis [81]. Badjatiya et.al. involved utilizing the GloVe embedding method combined with an LSTM classifier, achieving high accuracy through learned embeddings [106]. Furthermore, Sutejo et.al. employed two models: a Textual model and an Acoustic model, noting that the LSTM model performed better with textual data than with acoustic data [75]. For distinguishing between hateful and neutral content, NLP classifiers with paragraph2vec were utilized. Experimentation demonstrated improved performance with an increased number of hidden layers, with a specific configuration of five hidden connected units and two hidden layers achieving a 0.99 AUC over 200 iterations [107]. Ensembling of LSTM classifiers was explored by Pitsilis et.al. along with combining various features, resulting in a high F-score of 0.9320 [91]. Additionally, in the context of the Hinglish language, Varade et. Al. discovered that specific hyperparameter settings enabled the LSTM classifier to achieve a maximum recall value of 0.7504 [17].

#### 2.2.7.2.1 Convolution Neural Network

CNN, a subclass of Deep Neural Networks, is predominantly utilized for analyzing visual imagery. It involves converting the three layers of an image into a vector of suitable size, followed by training a DNN on this representation. Beyond visual imagery, CNNs find application in various domains such as video understanding, speech recognition, and natural language processing. For instance, Park et. Al. employed CNN to detect instances of racism and sexism in speech. The proposed model underwent testing using 10-fold cross-validation, yielding a 78.3% f-score [63]. Andreou et.al. incorporated text features, including surface-level, linguistic, and sentiment features, into deep learning classifiers, implementing an ensemble-based novel approach. This approach achieved an accuracy of 0.918. The performance of such deep learning systems is influenced by parameters such as batch size, epoch, and learning rate. Moreover, research indicates that larger training datasets lead to improved results [62]. Additionally, Modha et.al. introduced a CNN-based web browser plugin designed to visualize online aggression on platforms like Twitter and

Facebook. This tool provides a means to effectively monitor and analyze aggressive behavior in online social networks[108].

#### **2.2.7.2.2 Transformer Methods**

The transformer, as evidenced by the work of, emerged as a groundbreaking innovation in the field of natural language processing (NLP). Unlike its predecessors, transformers have the capability to capture long-term dependencies. Unlike LSTMs and RNNs, transformers do not process data sequentially. Instead, they incorporate the position of each word into its embedding. Initially introduced for machine translation, transformers consist of two components: an encoder and a decoder. In text classification tasks like hate speech detection, only the encoder is relevant. In the encoder, inputs are first fed into a self-attention layer, which generates embeddings considering the relevance of each word to others in the sentence. These embeddings are then processed through neural networks, with multiple layers of self-attention and neural networks stacked to form the encoder. The decoder, similar to the encoder but with the addition of an Encoder-Decoder attention layer, is utilized to determine the inputs relevant to a specific output, for hate speech detection, embeddings obtained from pre-trained models such as BERT (Bidirectional Encoder Representations from Transformers) are widely utilized. BERT, trained using the masked language modelling (MLM) technique, predicts masked words within sentences to learn contextual representations. Horev et.al. explored the efficacy of fine-tuning BERT for hate speech detection. Additionally, the comparison and experimentation with various LSTM and BERT-based models led to the conclusion that transformers, particularly a streamlined version called DistilBERT, outperform other models in terms of accuracy. Plaza-del-Arco et.al. performed a comparative analysis of traditional machine learning models, deep learning models, and transfer learning-based models for hate speech classification in Spanish was conducted [73]. Transfer learning models, especially pre-trained monolingual language models like BETO, outperformed traditional machine learning models, highlighting the importance of language-specific models for hate speech detection. GPT-2 (Generative Pre-trained Transformer 2), is a language modelling transformer trained on a massive dataset of web text. Unlike BERT, GPT-2 is utilized for generating sentences rather than creating embeddings. GPT-2 relies on autoregression, producing tokens sequentially and incorporating each token as input for the next. Despite its limitations in utilizing context on both sides, GPT-2 has demonstrated excellent results as depicted by Wullach et.al. [109]. Furthermore, Behzadi et.al. proposed a pipeline model using transfer learning and Compact BERT variants, with focal loss used as the cost function to address class imbalance. Ensemble learning with different features, including TF-IDF and sentiment-based features, was employed to improve overall accuracy[110] [111]. A brief summary is presented in Table 2.4.

Table 2.4. Deep Learning Methods for OHS

Reference	Approach	Language	Dataset	Merits	Limitation
Saksesi et al.[80]	RNN	Indonesian	1235 words from Twitter	PR-91%, RC-90%, Accuracy-91% was attained	Better results can be achieved if the size of the data increased.
Albadi, Kurdi, & Mishra [104]	RNN plus GRU	Arabic	600 tweets from Twitter	GRU Based RNN performs the best with an accuracy of 0.79.	State of the art analysis is missing
Sazany et al. [81]	LSTM, FastText algorithm	Indonesian	713 Twitter political post	97.39 f1-score was attained	Model configuration, such as the classifier, number of layers, training batch size is not analyzed.
Vigna et al. [50]	LSTM, SVM	Italian	17567 comments collected from Facebook	Binary classifier obtained comparable results as that of sentiment analysis.	Both SVM and LSTM are not able to discriminate between the three classes.
Badjatiya et al. [106]	CNN, LSTM, FastText	English	16K tweets	CNN is performed better than LSTM, which was better than FastText	Not applicable



Park & Fung [63]	HybridCNN	English	Waseem and Hovy 2016 English dataset (20k)	10-fold cross-validation performed with 78.3% f-score.	More precise results can be explored if training the two-step classifiers on separate datasets
Pactzold et al. [103]	RNN	English and Spanish	HatEval website	The proposed model is robust, even when the input data is noisy.	More reliable ways of re-training pre-trained compositional models can be tested.
Sutejo & Lestari [75]	LSTM	Indonesian	text-2273 and audio-2469	Textual model gives the best result as that of the acoustic model.	CBOW(87.98%) performed better than word n-gram and their combination
Andreou et al. [62]	Ensemble-based classification using CNN DNN RNN	English	Davidson et al.	An accuracy of 0.918 was attained.	The proposed system is not compatible with any cross-lingual.

### 2.2.8 OHS Detection using BERT and LSTM

As the number of social media platforms continues to multiply, offering features like anonymity, easy accessibility, and opportunities for online community building and debates, the challenge of detecting and monitoring hate speech is becoming an increasingly significant concern for society, individuals, policymakers, and researchers. M.S. Jahan et al. conducted an extensive literature review in this field, emphasizing the application of natural language processing and deep learning methodologies. It emphasizes key terminology and core methodologies, with prime focus on deep learning architectures [112]. PRISMA guideline for systematic reviews was considered encompassing literature from the past decade. Certain limitations in current research were identified and insights into future research directions were provided concerning hate speech detection and tracking [113]. W. Yin et al. provided a summary of the extent to which existing hate speech detection models can be generalized and explored the underlying reasons for the challenges faced in achieving this generalization. Additionally, it reviewed previous efforts made to address the primary obstacles and suggests directions for future research [114]. F.E. Ayo et.al. centered their research around the development of a comprehensive metadata architecture. This architecture was proposed to categorize hate speech into predefined score groups using semantic and fuzzy logic analysis [115]. Conversely, N. Chetty et.al. focused on the examination of hate speech pertaining to gender, religion, and race, particularly within the context of cyberterrorism. The proposed approach does not specifically concentrate on Twitter datasets, and can be generalized to vast domain of data [116]. Apart from this, A. Matamoros-Fernández explored various aspects related to hate speech, including geographical considerations, the diversity of social media platforms involved, and the qualitative or quantitative research methods employed by researchers in this field [117]. Another comprehensive survey was provided by F. Alkomah which provided empirical evidences on hate speech detection [118]. S.Mac Avaney et.al. identified and addressed the challenges that online automatic hate speech detection approaches encounter when dealing with text data. These challenges encompass several aspects, including the nuances in language, variations in definitions and limitations. A significant concern of lack of interpretability was noted that concerns with understanding the rationale behind the decisions made by these systems can be a complex endeavour. To tackle these challenges, a multi-view Support Vector Machine (SVM) approach was proposed. This approach attains significant performance levels close to the state-of-the-art but also offers simplicity and produces decisions that are more readily interpretable compared to neural methods. Also, the complexities associated and offer potential solutions to enhance the effectiveness of the approach was proposed [119].

Cao et al. introduced a distinct deep learning approach for detecting hate speech on online social networks. The proposed method incorporates word embedding, emotions,

and subject-related information across three important publicly available datasets. The proposed model, DeepHate, outperforms existing state-of-the-art approaches in hate speech detection, as indicated by the research findings. After that it is predicted that the DeepHate model will integrate non-textual elements and adopt more advanced techniques to enhance the representation of sentiment and subjects in online postings [120]. Roy et al. introduced an automated method centered on Deep Convolutional Neural Network (DCNN) based on GloVe embedding [121]. Zhou et al. introduced multiple techniques, including Embeddings from Language Models (ELMo), BERT, and CNN, for text classification applied to the SemEval 2019 Task 5 datasets. They integrated these classifiers using fusion methods to enhance the overall classification performance. In the future, embedding technologies from ELMo or BERT could potentially replace the basic word vector expression in CNN [122]. Plaza-del-Arco et al on identifying Spanish hate speech on social media. They compared the accuracy of various Deep Learning methods with recently pre-trained language models and traditional machine learning models. The findings revealed that the monolingual pre-trained language model (BETO) demonstrated superior performance compared to mBERT and XLM [123]. Miok et al. applied the Bayesian method with Monte Carlo dropout within the attention layers of transformer models to offer calibrated reliability estimates. Their research demonstrated the model's ability to detect hate speech across different languages. Additionally, they explored how emotional dimensions could enhance the information captured by the BERT model. Looking forward, they aim to explore other Bayesian approaches for transformer networks, including options like SWAG [124]. Michele et al. introduced an innovative neural architecture that has shown effective performance in multiple languages, such as English, Italian, and German. They conducted a comprehensive analysis across these three languages to enhance their understanding [125]. Arango et al. analysed the noTable disparity between existing literature and real-world applications, aiming to assess the generalizability of prior studies to different datasets. Their results highlighted methodological shortcomings and a considerable dataset bias. Common concerns identified included data overfitting and sampling flaws [126]. M. Mozafari et.al. proposed an innovative transfer learning method centred on the advanced language model BERT. Their research focused on harnessing BERT's capabilities. To evaluate their approach, they utilized two datasets annotated for racism, sexism and hate speech. The results demonstrated significant improvements in precision and recall compared to existing methods. As a result, their model shows promise in mitigating biases in data annotation and collection processes, leading to a more precise and effective hate speech detection model transfer learning approach centred around a pre-existing, state-of-the-art language model called BERT [127]. A brief summary is presented in Table 2.5.

Table 2.5. comparison of state-of-the-art techniques

Author and Year	Features	Approach	Dataset	Results
S.Mussiraliye va et al, 2023 [128]	BOW, TFIDF, Word2vec	Bidirectional Long Short Term Memory (BiLSTM), CNN, LR, DT, SVM	Twitter dataset <sup>20</sup> , Cyberbullying Classification Dataset <sup>21</sup> , Hate Speech and Offensive Language Dataset <sup>22</sup>	90.2% Accuracy
H. Saleh et al, 2022 [129]	BERT	Bi-Directional LSTM	Davidson-ICWSM[41], Waseem-EMNLP [42], Waseem-NAACL [43]	93%, Accuracy
Ferraro, G. et al, 2023 [130]	Word-embedding	bi-LSTM	WASEEM data set [44]	78.09, Accuracy
Sayani Ghosal and Amita Jain. 2023 [131]	Emotion and hate Lexicon	parts of speech tagging, Euclidean distance, and the Geometric median methods	Bengali datasets [45]	78%, Accuracy
N. & Djeflal et al, 2023 [132]	FastText, GloVe	Bi-LSTM, Bi-GRU	Kaggle [46]	98.63 ROC-AUC score

<sup>20</sup> <https://www.kaggle.com/datasets/andrewmvd/cyberbullying-classification>

<sup>21</sup> <https://dl.acm.org/doi/abs/10.1007/s10579-020-09488-3>.

<sup>22</sup> <https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset>

Gurevych, I. et al., 2023 [133]	Word-embedding	CNN, BiLSTM, and mBERT	Stormfront dataset [47]	75% Score	F1	
S. Khan et al., 2022 [134]	Word-embedding	Convolutional, BiGRU, and Capsule network-based deep learning model	Founta [48],	93% Accuracy	Accuracy	
P. William et al., 2022 [135]	BOW, TFIDF, Word embeddings	SVM	Personality Prediction [49]	79% accuracy	accuracy	
Shakir Khan et al., 2022 [136]	BERT	BiLSTM with deep CNN and Hierarchical Attention-based deep learning model	Founta [48]	89% accuracy	accuracy	

### 2.2.9 Evaluation Metrics for OHS

Evaluation metrics play a crucial role in assessing the performance of traditional machine learning models, offering valuable insights into their quality. State-of-the-art online hate speech detection techniques often rely on various metrics such as F1 score, precision, recall, and accuracy to gauge the effectiveness of model parameters[49], [53], [67], [77], [92].

#### A. Precision

It measures the proportion of relevant information retrieved from the total retrieved information. It is calculated using the formula:

$$P = Precision = \frac{TP}{TP+FP} \quad (2.12)$$

Where:  $P$  represents precision,  $TP$  stands for true positive, and  $FP$  stands for false positive.

#### B. Recall

It assesses the percentage of total relevant information correctly identified by the classifier. The recall score is computed as:

$$R = Recall = \frac{TP}{TP+FN} \quad (2.13)$$

Where,  $R$  denotes recall,  $FN$  represents false negative and  $TP$  stands for true positive.

#### C. F1-Score

It is the harmonic mean of precision and recall. F1 score has become the preferred choice of measuring the performance of machine learning models. This can be attributed to the fact that F1 score gives equal weightage to both precision and recall and it punishes models that lack even in one of them.

$$F1\ Score = \frac{(2*P*R)}{P+R} \quad (2.14)$$

In multiclass classification there are mainly two methods of calculating F1 score, namely Micro averaged F1-score and Macro Average F1- score.

#### D. F1 Micro Averaged

This metric is simply calculated by taking the harmonic mean of Micro precision and Micro recall. An important feature of this metric is that it assigns equal value to each label, the repercussion of which is the not enough attention is given to minority classes in case of imbalanced datasets. Since Imbalanced datasets are seen in abundance in the

domain of hate speech detection, the use of Micro Averaged F1-score should be minimized.

$$\text{Micro Averaged Precision} = \frac{\sum TP}{\sum TP + \sum FP} \quad (2.15)$$

$$\text{Micro Averaged Recall} = \frac{\sum TP}{\sum TP + \sum FN} \quad (2.16)$$

### E. F1 Macro Averaged

This is calculated by simply taking the mean of F1 scores obtained on each class individually. This metric assigns equal value to each class and thus should be the preferred metric in the context hate speech detection where datasets are generally imbalanced and models are expected to be proficient in detecting all classes.

### F. Confusion Matrix

It is a performance measurement matrix comparing the actual and predicted observations through the values of False Positives (FP), True Negatives (TN), True Positive (TP), and False Negative (FN) labels (Matrix 1).

$$\text{Confusion Matrix:} = \begin{bmatrix} TP & FN \\ FN & TN \end{bmatrix} \quad (2.17)$$

### G. Accuracy

Is the measure which tells how efficiently the classification models produce the results correctly.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (2.18)$$

### H. Comprehensiveness

In XAI, we essentially try to predict the factors which led to a model's decision. To calculate the comprehensiveness, the factors predicted by the XAI model is first removed from the datapoint. In the context of hate speech detection, the equivalent of this is removing the words predicted by the XAI model. Now, this new modified datapoint is then fed into the model. The change in the model's confidence in prediction is noted. A change implies that the factors predicted by the model indeed contributed to the model's decision[137].

### I. Sufficiency

These metric measures how important the extracted rationales (words or phrases in the context of Hate speech detection) for the model to make a prediction[137].

#### **J. Matthews correlation coefficient (MCC)**

It tries to find the relation between the true and predicted values. Higher value of the coefficient shows the better results. Whenever the given dataset is highly imbalance in that case it is found that MCC has given best results compared to the accuracy [138]. Its value always lies between -1 to 1. The given formula is Shown in equation 2.18.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TP + FN)(TN + FP)(TN + FN)}} \quad (2.19)$$

Both precision and recall are very important and the most used evaluation metrics in traditional machine learning and deep learning classification. We can calculate the accuracy  $y$  by providing the given values to TN, TP, FP, and FN. By getting the values of precision and recall from equations 1 and 2, we can calculate the F1 score that is used to test the accuracy of the parameter. Some authors also used AUC (area under the curve) to compute the performance of the model. The aforementioned metric evaluation formulas were used by mostly all other authors mentioned in Related works to evaluate the performance of their Machine Learning model.



## CHAPTER 3

### **Proposed an Efficient Hate-Swarm Algorithm for Classifying Hate Speech on Social Media**

This chapter presents a novel approach to combat the growing problem of harmful information and hate speech on social media and other internet platforms. In this chapter, a novel feature engineering technique called HateSwarm is proposed. This technique employs bio-inspired algorithms to select efficient features for the binary classification of non-hate and hate speech. The baseline machine learning models were trained on these features and the performance of proposed algorithm was evaluated on two benchmark datasets.

#### **3.1. Introduction**

The rise of technology and the widespread availability of the internet has fundamentally changed the way people communicate and access information. With the click of a button, individuals can now easily share and access news and information on a global scale. Social media, in particular, has emerged as a powerful platform for people to express their opinions and connect with others. As a result, interactions among various groups of people from different parts of the world have become more accessible and effortless than ever before. However, this newfound accessibility has also led to the dissemination of harmful information and hate speech, posing serious threats to individuals' mental health. In this context, there is a growing need for intelligent systems that can detect and classify toxicity on the internet in real-time to prevent its spread. Besides its huge benefits, this has also led to the deliberate sharing of hateful and toxic content [139]. Individuals can make harmful and false remarks using harsh or obscene language against anyone without any real constraints or procedures. The purpose of such behavior is to damage the person's reputation and standing in the community. By manipulating people's emotions and persuading certain groups of people to start controversies, this toxicity has presented severe challenges to society [140]. The phrase "hate speech" refers to language that disparages an individual or a group of individuals in an effort to harm that person's reputation in society because of their gender, race, ethnicity, skin color, nationality, political activity, sexual orientation, other geographic features. Hate speech has been more prevalent in recent years, both in-person and online because communications are delivered and received virtually instantly, social media networks (SMNs) are the quickest method of communication. Hateful content is bred and propagated on social and Internet platforms, which finally leads to hate crimes. Such hate speech can have severe social, economic, and political consequences. This can prove highly harmful to specific individuals, groups, Businesses, and Governments and can quickly tarnish their reputation. Hate speech refers to disparaging or discriminating rhetoric directed

towards a group of people based on their origin, sexual orientation, ethnicity, religious affiliation, socioeconomic status, race, gender, or a variety of other characteristics. Online hate speech (OHS) is defined as such conduct when it occurs on social media platforms, blogs, creative work, and other online media [141]. The definition of hate speech varies from country to country, but it is generally understood to include expressions of hostility or disparagement of a person or a group because of a trait shared by both the individual and the group, such as race, color, nationality, sexuality, disability, belief, or sexual orientation. Thus, a significant part of the content on social media is damaging and demeaning to user's mental health over time [48]. The widespread access of social media websites to individuals from linguistically distinct regions and cultures has led to a blend of natively spoken languages with English, popularly known as code. The term "multilingualism" describes both a person's propensity for using many languages and the coexistence of various linguistic groups in a single geographic location [142]. Contrary to popular belief, the majority of people on the planet speak more than one language. Only a small percentage of the world's population is monolingual. When speaking to one another, multilingual occasionally combine aspects of several languages, creating a dialogue with mixed coding.

It is crucial to stop the circulation of such hateful content and toxic information. It can be noticed that some hate speech features are almost similar to offensive features, and thus a foremost challenge for us is to differentiate between hate, non-hate, and offensive. Previous research shows that supervised Machine Learning (ML) models have proven efficient in detecting hate speech in online social media [143]. Alongside that, deep learning models have also been used in many previous research works to deal with the various research problems on hate speech. Our research also aims to devise a reliable ML-dependent system that can automatically detect hate speech in social media with high precision and accuracy. For the identification of hateful speech on social media sites and other online forums., our studies have suggested an effective and deployable ML model with an efficient feature extraction technique. An essential step in order to identify hate speech and build an ML model is to prepare high-quality data using efficient feature extraction and selection methods. This approach significantly reduces the computational cost of training a model on a big text dataset while simultaneously enabling us to construct a less sophisticated model that is simpler to comprehend, less prone to overfitting, and more accurate. The supervised ML model in this study included a variety of text feature extraction techniques, including term frequency-inverse document frequency, Word2Vec, and Doc2Vec. Producing feature representations of the text data is one of the goals of using these methodologies. The most beneficial attributes among these features were selected using an amalgamation of modified PSO and GA, also referred to as the proposed HateSwarm algorithm in this paper. It is suggested to combine modified PSO with GA to address optimization problems. GA is an evolution-based algorithm, whereas PSO is a swarm-based algorithm. The best text features were found and selected using population-based

heuristic search approaches. We have performed and evaluated our method on two benchmark open-sourced datasets and evaluated the efficacy of our method was measured using the performance metrics. The performance of a baseline ML models was examined in our research both with and without the proposed feature engineering approach. We investigated the performance of multiple ML models to discover which feature selection method in order to achieve the highest possible level of classification. This chapter provides significant contributions to the field of hate speech detection and classification. The following highlights the key contributions of research;

- A novel approach for detecting and classifying hate speech on the internet by proposing a feature engineering technique named HateSwarm. The proposed algorithm leverages modified Particle Swarm Optimization and Genetic Algorithm approaches to enhance the performance of hate speech classification.
- The proposed algorithm is data-centric, with a focus on optimized and improved data. This approach ensures that the algorithm can effectively detect and classify hate speech in real-time, preventing its spread and potential harm to individuals.
- Conducted a thorough analysis of the performance of proposed approach on various machine learning baseline models, including Support Vector Machines, Decision Trees, and Logistic Regression. This analysis allows us to demonstrate the efficacy of our approach compared to other state-of-the-art techniques.
- Tested the proposed approach on multiple datasets, ensuring that the algorithm can detect and classify hate speech across different contexts and domains. This comprehensive evaluation of our approach provides strong evidence of its generalizability and effectiveness in real-world applications. The proposed algorithm was trained on various hate speech datasets that could classify non-hate and hate speech with a 92% accuracy.

### **3.2. Limitations of the Existing Research**

- From the study, it has been observed that the existing research covers mostly lexicon (simple keywords) based features for the hate speech analysis. Which restricted the results because the models will not be suitable if whole meaning of the sentence is needed. So, knowledge-based feature, semantic features can be taken into consideration with lexicon-based features. By this, the accuracy of the model can be increased.
- It was found that with the increase in the number of features, the threshold value increases, which in turn may decrease the accuracy of the model. Therefore, whenever large feature data is given, the model gets confused because it is learning too much information. So, in order to resolve this

situation, only the relevant features are selected from all sets of features, which increases the accuracy of the model.

### **3.3. Proposed Methodology**

The dissemination of harmful information and hate speech has become a significant problem on social media and other internet platforms. The quick spread of such toxicity can cause severe damage to people's mental health. Therefore, the need for an intelligent system capable of detecting and classifying hate speech in real time has become more pressing than ever. In the field of machine learning, the quality of data used for training models plays a crucial role in determining the accuracy of predictions. While traditional approaches focus on developing complex algorithms for prediction, our approach centres around optimizing the dataset itself through effective feature engineering and transformation. By enhancing the quality and relevance of the data, we have created a more robust and accurate model without relying solely on algorithmic improvements. Our data-centric approach has allowed us to achieve significant improvements in accuracy, surpassing conventional benchmark datasets. In this research, a novel amalgamation of the modified version of the PSO and GA algorithm called HateSwarm is proposed to select the most suitable data features that can improve the ML model's performance. Our methodology is divided into three main parts: feature extraction, feature selection, and then model building. Feature extraction and selection are part of our feature engineering process [144].

#### **3.3.1 Machine learning Classifier**

Supervised machine learning models learn to predict the corresponding class using the labeled data. The labeled inputs are mapped to the output based on their features. After selecting suitable features using the proposed HateSwarm algorithm, the following three baseline ML models were trained. All these models input the text's data features and output the class to which the given text belongs. Our problem statement was classified as binary (hate or hate speech) (0 vs. 1). Three baseline models viz. Support Vector Classifier (SVC), Decision Tree Classifier (DT), and Logistic Regression (LR) were trained. Then, these models were selected because of their ability to deal with the large dimensionality of data. These models were separately trained twice with each set of features, with and without using the proposed algorithm for feature selection.

##### **3.3.1.1. Support Vector Classifier**

SVC is a powerful and robust linear classifier that finds a hyperplane that will maximize the distance between the two classes and separate them by determining the decision boundary. Using the support vectors, the distance that is furthest from each one of these groups is selected. Hinge loss as the cost function is used to penalize the misclassifications. This algorithm can be easily applied to nonlinear problems. SVC

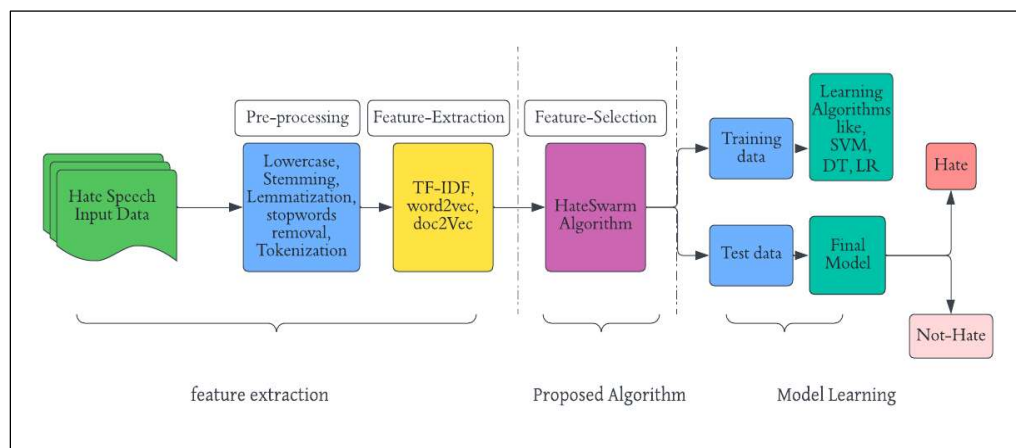
works fine even if the dimensionality of the Data is large and is impacted less due to outliers [48].

### 3.3.1.2 Decision Tree Classifier

DT is also a classification algorithm that segments the data based on its features to predict the results. It splits the data into multiple subsets (mostly two subsets) at every step using various impurity measures such as Gini Index and Entropy. This splitting based on features continues until the last nodes called leaves are pure and belong to a single class. Pruning is used to prevent overfitting in DT. DTs are easy to interpret and require no data preprocessing. It deals well with a large number of features [145].

### 3.3.1.3 Logistic Regression

LR is a binary classification algorithm trained to learn a bunch of parameter values. Multiple iterations are run to fit an LR model on the dataset and find those parameter's values. In LR, each feature is given an important value using weights. Like SVM, it works well even with the large dimensionality of data and suffers less impact of the outliers due to the sigmoid activation function. LR predicts the probability of that data point belonging to either class [146]. Fig. 3.1 describes the proposed methodology. In first section of the Fig., various methodologies were utilized to optimize the performance of hate speech classification. A comprehensive overview of the feature extraction techniques is presented that were used to transform raw textual data into numeric representations that can be utilized by machine learning algorithms. In next section, our proposed algorithm is introduced, which leverages a combination of modified Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) approaches to select the most significant and valuable features. The final subsection of this section outlines the various baseline machine learning models trained on the collected and pre-processed dataset, which serves as the foundation for our experimental evaluation.



**Fig. 3.1. Proposed Hate Speech Detection Methodology**

### 3.3.2 Feature Extraction

Data cleaning and pre-processing is a crucial step in preparing raw data for machine learning model development. Raw data is often high-dimensional and noisy, containing irrelevant features and redundant information. Pre-processing and cleaning this data help to improve the accuracy of the model and reduce training time. In this study, various text cleaning and pre-processing techniques were employed to ensure the quality of our data. First, stop-words were removed, which are common words that do not add significant value to the text. Then, unwanted special characters, numbers, and punctuations that could cause issues during model training were removed. Additionally, text normalization techniques such as lowercasing, stemming, and lemmatization were applied to further enhance the quality of the data. Lowercasing refers to the conversion of all characters in the text to lowercase, which helps to reduce the size of the vocabulary and improve the consistency of the data. Stemming involves reducing words to their root form, which helps to group similar words together and reduce the complexity of the data. Lemmatization, on the other hand, is the process of converting words to their base or dictionary form, which helps to preserve the meaning of the words while reducing the size of the vocabulary. By employing these text cleaning and pre-processing techniques, high quality and well-suited data quality was ensured for use in machine learning models.

---

#### *Text cleaning Steps:*

---

- 1. Convert all characters to lowercase*
  - 2. Remove URLs and mentions (words that start with '@')*
  - 3. Remove hashtags but keep the words after them*
  - 4. Replace contractions (e.g., "can't" -> "cannot")*
  - 5. Replace slang and informal words with their standard equivalents (e.g., "u" -> "you")*
  - 6. Remove non-alphabetic characters and punctuation, except for exclamation and question marks*
  - 7. Tokenize the text into individual words*
  - 8. Remove stop words (common words that don't carry much meaning, such as "the" and "and")*
  - 9. Perform stemming or lemmatization to reduce words to their base form*
  - 10. Combine the remaining words back into a single string and return the cleaned text.*
- 

After the cleaning of text data, various feature extraction techniques were used namely, TF-IDF, Word2Vec, and Doc2Vec for feature extraction due to their effectiveness in capturing important textual features and representing them as numerical features. TF-IDF is a widely used technique that assigns weights to each word based on its frequency in the document and its inverse frequency in the corpus. Word2Vec and

Doc2Vec are both neural network-based techniques that can represent words and documents in a vector space, capturing the semantic meaning and context of the text. The CBOW Word2Vec model was used for hate speech dataset, while PV-DBOW was used for Doc2Vec. These techniques were selected based on their performance in previous studies and their potential to improve hate speech classification accuracy.

- **TF-IDF: Term Frequency** – Inverse Document Frequency is a text feature extraction technique. Each word in the document is assigned a specific weight based on its frequency in that particular document and the total number of documents out of the whole corpus that contains that word. The higher the weight of the more important is the term in the document. This statistical measure estimates the importance of a term in a document and the whole corpus [147].
- **Word2Vec:** Unlike TF-IDF, in Word2Vec, each word is represented by a vector of floating-point values. These vector representations are trained to acquire multiple relations and analogies between words. Thus, it captures the actual semantic meaning and the word's context concerning the document's other words. Vector representations of the Word2Vec method are trained using Neural Networks. It can be trained using two techniques, viz. CBOW (Continuous Bag of Words) and Skip-Gram. In our experimentations, CBOW Word2Vec model was used to extract text features from hate speech Dataset [148].
- **Doc2Vec:** This text feature extraction technique is very similar to Word2Vec. As in Word2Vec, each word is projected as a vector of fixed size. Doc2Vec learns to project the whole document in a vector space. It creates an N-dimensional Vector of the whole document, and similar to Word2Vec, it also used Neural networks to train these Vectors. It can be implemented in two ways, viz. Paragraph Vector - Distributed Memory (PV-DM) and Paragraph Vector - Distributed Bag of Words (PV-DBOW). In our experimentations, PV-DBOW was used[149].

### 3.3.3. HateSwarm: Proposed Feature Selection Technique

Feature selection in the classification task can be substantive: given all features (F) consisting of m available features, find a feature subset S consisting of n relevant features, where  $n \leq m$  and  $S \subset F$  without replacement. Irrelevant features are eliminated, and the number of features selected reduces, thus fulfilling enhanced efficiency and increased accuracy. When dealing with enormous amounts of data, one of the most fundamental and often applied techniques is feature selection. This method is used to get rid of features that aren't being used and aren't needed. These insignificant characteristics do nothing but make the ML model less effective and increase the likelihood of errors.

PSO and GA, which are swarm-based and evolutionary search strategies, can be used to solve highly nonlinear optimization problems [150]. These are the heuristics methods that are inspired by the collaborative behavior of biological populations. They both formulate a set of probabilistic and deterministic rules to find the most optimum point. This is a population-based search method where the information is shared among the population members based on these rules. These optimization algorithms are required to solve complex objective function that is noisy, nonlinear, high dimensional, and computationally expensive. PSO and GA are stochastic optimization algorithms that begin searching for the optimum points from a randomly generated population that evolves over successive iterations. Over successive iterations, they find a more superior solution to the optimization problem [151].

Animal species such as birds, ants, fishes, and hawks travel in groups and interact locally with each other and their global environment. These groups show an intelligent global behavior known as Swarm Intelligence. This Swarm Intelligence inspired the PSO optimization algorithm. On the other hand, GA was inspired by natural selection principles that include GA consisting of three leading operators viz. Selection, Crossover, and Mutation[152].Using these GA propagates from one generation to another. These computational models have a wide range of real-life applications. These algorithms have been successively applied in many engineering applications, fuzzy logic control [153], [154], Energy-Storage Optimization, robotics, Artificial Intelligence etc. These algorithms are one of the advanced feature selection methods in ML.

Our proposed feature selection technique integrates modified PSO and GA to identify a subset of discriminative hate features from a larger feature space. The proposed approach was utilised in order to identify the features that were most appropriate and effective for training the ML model. Feature selection using the proposed algorithm contains two steps. Firstly, modified PSO was used to select the text features, and then the features selected by it were then given as an input to the GA to finally determine the most optimum set of features for model training. This approach made our ML model more Data-Centric, focusing more on data processing and feature than on optimizing the model parameters. Working on data can significantly improve the prediction results as it can ensure high data quality and accelerate the model training process.

Algorithm 3.1 in this research, modified by introducing an additional term  $c_3$  is a constant that controls the particle's velocity update and influences its movement in the search space. In that, this additional term helps the particle to explore the search space beyond its local best and global best positions by adjusting its velocity based on the position of the iteration best particle. This modified fusion will help us achieve a faster convergence rate in reasonable computational time, making it more flexible and robust. The proposed approach will find the optimum global point by avoiding getting



trapped in a local optimum point. By introducing this term, the modified PSO algorithm can overcome the problem of premature convergence and reach a more optimal solution. This additional term helped us to improve the optimization process and achieve faster convergence.

---

**Algorithm 3.1: HateSwarm**

---

**Input: X:** All the hate features vector collected from feature extraction technique

**Output: H:** Subset of n relevant hate features vectors

$T_{itr1}$  → Number of iterations in modified PSO

$E_{stop}$  → Early Stopping condition in modified PSO

$T_{itr2}$  → Number of iterations in GA

$N_{Popu}$  → Population of Particle

$Plateau_{itr}$  → Number of iterations with no improvement to consider plateau

$Plateau_{tolerance}$  → Tolerance for the fitness change to be considered as a plateau

**Procedure Modified PSO Algorithm**

Set  $k = 0$ ,  $e = 0$

Set Parameters  $N_{Popu}$ ,  $c_1$ ,  $c_2$ ,  $c_3$ ,  $\beta$ ,  $[V^{min}, V^{max}]$ ,  $[X_g^{min}, X_g^{max}]$ ,  $P_{best}$ ,  $G_{best}$ ,  $I_{best}$ ,  $X_{train}$ ,  $Y_{train}$ ,

$F$ ,  $k = 1$

Randomly initialize the positions and the velocities of the particles ( $X_{g,j}^0$ ,  $V_{g,j}^0$ )

prev\_fitness = infinity

plateau\_counter = 0

**while**( $k < T_{PSO}$ )

**for** each particle  $i$  in swarm do

        Take hardman product of  $X_{train}^i$  with  $X_g^i \rightarrow X_p^i = X_{train}^i \circ X_g^i$

        Calculate fitness  $f(X_p, Y_{train})$  according to equation 1 for each particle  $i$ ;

        Update Local best  $P_{best,j}^k$ , Global best  $G_{best,j}^k$  and Iteration best  $I_{best}$

        Update particles and velocities  $X_g^i$  and  $V_g^i$  according to equations 2,3 and 4

**end for**

**If**  $k > 1$  and  $abs(fitness - prev\_fitness) \leq Plateau_{tolerance}$

```

Calculate fitness value for updated  $G_{best}^U f(G_{best}^U)$ ;
If  $f(G_{best}^U) == f(G_{best})$ 
    e += 1
    if e > Estop
        break
    end if
end if
end if
    k = k + 1
end while
while(k < TGA)
    for each particle i in swarm do
        for each training sample j do
            Take hardman product of  $X_{train}^j$  with  $X_g^i \rightarrow X_p^i = X_{train}^j \circ X_g^i$ 
        end for
        Calculate fitness  $f(X_p, Y_{train})$  according to equation 1 for each particle and
        Update F
    end for
    Select H particles with maximum fitness  $\cup_{i=1}^{i=h} X$ 
    For each selected particle  $g_i$  in H do:
        Off =  $g_i @ g_{i+1}$ 
        Off[rand()]!
        Update NPopu
    end if
    k = k + 1
end while

```

---

$$Fisher'sScore = \frac{(mean^1 - mean^2)^2}{(variance^1 + variance^2)} \quad (3.1)$$

Fisher's Score is a statistical measure that quantifies the discriminatory power or relevance of features for classification problems. The fisher's score as a fitness formula in PSO for ranking the features was used. To compute the Fisher's Score for each feature in the dataset equation 3.1.is used. It measures the separability of classes by considering the means and variances of features for different classes of hate speech. Where,  $mean^1$  and  $mean^2$ : Mean values of the feature for each class whereas,  $variance^1$ ,  $variance^2$ : Variances of the feature for each class. The higher the Fisher's Score, the more discriminatory or relevant the feature is for classification.

$$V_i^{k+1} = w * V_i^k + c_1 * r_1 * (P_{best,i}^k - X_i^k) + c_2 * r_2 * (G_{best,i}^k - X_i^k) + c_3 * r_3 * (I_{best,i}^k - X_i^k) \quad (3.2)$$

In equation 3.2,  $X_i^k$  is the current feature subset being evaluated by the particle i at iteration k. It represents a combination of features used to detect hate speech.

$I_{best,i}^k$  the individual best feature subset encountered by particle i throughout the optimization process. It represents the combination of features that has shown the best performance in terms of minimizing hate speech specifically for that particle.

Whereas  $I_{best,i}^k - X_i^k$  shows the difference between the individual best feature subset and the current feature subset. It quantifies the discrepancy between the particle's current feature subset and its own historical best.

The new introduced term  $c_3$  is the coefficient that controls the influence of the individual best term on the particle's velocity update. It determines how much importance is given to the particle's own historical best feature subset.

The term  $r_3$  is a random number between 0 and 1, which introduces stochasticity into the movement. It helps in exploring the search space and prevents particles from getting stuck in local optima.

This whole term  $c_3 * r_3 * (I_{best,i}^k - X_i^k)$  is represent the contribution of the individual best term to the particle's velocity update. It influences the particle's movement by encouraging it to explore feature combinations that are closer to its own historical best. Including this term in the velocity update equation 2 enables particles to leverage their individual successes and experiences with feature subsets that have shown promise in hate speech detection. It allows particles to explore and exploit feature combinations that have worked well for them individually, contributing to the overall optimization process and enhancing the effectiveness of hate speech detection.

$$V_{ij}^{k+1} = \frac{1}{e^{-V_{ij}^{k+1}}} \quad (3.3)$$

$$X_{ij}^{k+1} = \begin{cases} 0, & rand() \geq V_{ij}^{k+1} \\ 1, & rand() < V_{ij}^{k+1} \end{cases} \quad (3.4)$$

$C_p \rightarrow$  One crossover Point

$$\text{Offspring1}[i] = \text{Parent1}[i] \text{ if Mask}[i] == 1 \text{ else Parent2}[i] \quad (3.5)$$

$$\text{Offspring2}[i] = \text{Parent2}[i] \text{ if Mask}[i] == 1 \text{ else Parent1}[i] \quad (3.6)$$

In equation 3.5 the Offspring1 and Offspring2 are the two new offspring generated from the parents. Parent1 and Parent2 are the two parents being crossed over. Where Mask is a binary string of length equal to the number of features, with 0s and 1s indicating which parent is selected for each feature. The mask is randomly generated for each offspring.

The following gives a brief summary of the proposed algorithm's main steps:

1. **Set the initial parameters:**

- NPopu: This parameter determines the population size of particles. It represents the number of candidate feature subsets that will be evaluated.
- c1, c2, c3: These constants are acceleration coefficients used in the PSO algorithm. They control the impact of the particle's own best position (Pbest), the global best position (Gbest), and the iteration's best position (Ibest) on the particle's velocity update.
- $\beta$ : This parameter is a coefficient used in the velocity update equation 2 to balance the particle's current velocity with the global best position.
- [Vmin, Vmax]: These values represent the minimum and maximum range for the velocity of the particles. The velocity is clamped within this range during the update.
- [Xgmin, Xgmax]: These values represent the minimum and maximum range for the positions of the particles. The positions are constrained within this range.
- Pbest: This is an array that stores the local best positions of each particle. It keeps track of the best feature subset found by each particle so far.
- Gbest: This is an array that stores the global best position among all particles. It represents the best feature subset found in the entire swarm.
- Ibest: This is an array that stores the best position for each iteration. It keeps track of the best feature subset found at each iteration.
- Xtrain, Ytrain: These are the training data and corresponding labels used to evaluate the fitness of feature subsets. F: This is the fitness function that

measures the quality of a feature subset. It takes the feature subset ( $X_p$ ) and the training labels ( $Y_{train}$ ) as inputs and returns a fitness score.

- $T_{itr1}$ : This parameter specifies the number of iterations in the modified PSO algorithm.  $E_{stop}$ : This is the early stopping condition in the modified PSO. It determines the maximum number of iterations without improvement before stopping the algorithm.
- $T_{itr2}$ : This parameter specifies the number of iterations in the GA.  $Plateau_{itr}$ : This parameter determines the number of iterations with no improvement to consider as a plateau, indicating a possible stagnation of the algorithm.
- $Plateau_{tolerance}$ : This parameter sets the tolerance for the fitness change to be considered as a plateau. If the fitness change is below this tolerance, it indicates a possible plateau.

## 2. Initialize the particles:

- Set  $k = 0$  and  $e = 0$ : These variables represent the current iteration number and the early stopping counter, respectively.
- Randomly initialize the positions ( $X_{0g,j}$ ) and velocities ( $V_{0g,j}$ ) of the particles: Each particle is assigned a random position and velocity vector, representing a candidate feature subset. These initial positions and velocities are within the specified ranges  $[X_{gmin}, X_{gmax}]$  and  $[V_{min}, V_{max}]$ .

## 3. Perform the modified PSO iterations:

- Set  $prev\_fitness$  to infinity and  $plateau\_counter$  to 0: These variables are used to monitor the fitness changes and detect plateaus in the optimization process.
- Enter a loop while  $k < T_{itr1}$ : This loop controls the number of iterations in the modified PSO algorithm.
- For each particle  $i$  in the swarm, do the following:
  - Take the element-wise product (hardman product) of  $X_{traini}$  with  $X_{gi}$ : This step combines the training data ( $X_{traini}$ ) with the particle's current position ( $X_{gi}$ ) to obtain a modified feature subset  $X_{pi}$ .
  - Calculate the fitness  $f(X_p, Y_{train})$  according to equation 1 for each particle  $i$ : The fitness function evaluates the quality of the modified feature subset  $X_{pi}$  by comparing it to the training labels  $Y_{train}$ .

- Update the local best  $P_{kbest,j}$ , global best  $G_{kbest,j}$ , and iteration best  $I_{best}$ : These variables store the best positions found by each particle, the best position found in the entire swarm, and the best position found at each iteration, respectively.
  - Update the particles' positions ( $X_{ig}$ ) and velocities ( $V_{ig}$ ) according to equations 2, 3, and 4: The positions and velocities are updated based on the current positions, velocities, acceleration coefficients, and the global and local best positions.
  - If  $k > 1$  and  $\text{abs}(\text{fitness} - \text{prev\_fitness}) \leq \text{Plateautolerance}$ : This condition checks if the fitness change is below the tolerance, indicating a possible plateau.
    - Calculate the fitness value for the updated  $G_{Ubest}$ : The fitness function is evaluated for the updated global best position  $G_{Ubest}$ .
    - If  $f(G_{Ubest})$  is equal to  $f(G_{best})$ : This condition checks if the fitness value for the updated global best position is the same as the current global best position.
      - Increment  $e$  by 1: If the condition is true, it suggests a possible plateau, and the early stopping counter  $e$  is incremented.
      - If  $e > EStop$ : If the early stopping counter exceeds the specified threshold, the algorithm breaks out of the loop.
    - Increment  $k$  by 1: This updates the iteration number.
4. **Perform the GA iterations:**
- Enter a loop while  $k < T_{itr2}$ : This loop controls the number of iterations in the genetic algorithm (GA) phase.
  - For each particle  $i$  in the swarm, do the following:
    - For each training sample  $j$ , take the element-wise product of  $X_{trainj}$  with  $X_{gi}$ : This step combines each training sample with the particle's current position to obtain modified feature subsets  $X_{pi}$  for evaluation.
    - Calculate the fitness  $f(X_p, Y_{train})$  according to equation 1 for each particle and update  $F$ : The fitness values are calculated for the modified feature subsets, and the fitness array  $F$  is updated.
  - Select  $H$  particles with the maximum fitness and store their positions in the set  $H$ : This step selects the  $H$  particles with the highest fitness values and keeps their corresponding positions as the best feature subsets.
  - For each selected particle  $g_i$  in  $H$ , do the following:
    - Generate a new offspring  $Off$  by applying crossover ( $g_i @ g_{i+1}$ ) and mutation ( $Off[\text{rand}()]!$ ): This step creates a new offspring by

performing crossover and mutation operations on the selected particles' positions.

- Update the population size  $N_{Popu}$ : The population size  $N_{Popu}$  is adjusted based on the offspring generated.
- Increment  $k$  by 1: This updates the iteration number.

The algorithm begins by initializing various parameters that will define the behavior of the optimization process. These parameters include the number of particles, which represents the different candidate subsets of features used for identifying hate speech. These subsets are obtained from techniques like tf-idf, word2vec, and doc2vec. Each particle represents a specific subset of features. To explore the feature space and find an optimal subset of features, the algorithm adjusts the velocities and positions of the particles. The "velocity" of a particle determines its speed and direction within the particle swarm optimization (PSO) algorithm. It governs how quickly and in which direction a particle moves in the solution space. The "position" of a particle corresponds to a set of selected features. By adjusting the velocities, the algorithm aims to find the best combination of features that accurately classifies hate speech. To initiate the optimization process, the algorithm creates a swarm of particles with random positions and velocities. These particles collectively traverse the solution space, seeking better solutions. The algorithm updates the velocities of the particles based on factors such as the particle's previous velocity, its distance from its own best-known solution (local best), and the swarm's best-known solution (global best). This update guides the particles toward improved solutions in the search space. The fitness of each particle, which represents its quality as a solution, is evaluated using a fitness function. This function considers the classification performance of the feature subset selected by the particle. It typically utilizes measures such as true positive and true negative rates, which indicate the accuracy of the classification model. Throughout the optimization process, the algorithm keeps track of the best solutions found so far. This includes the local best, which is the best solution found by an individual particle, the global best, which is the best solution found by the entire swarm, and the iteration best, which is the best solution found during a particular iteration. These best solutions represent feature subsets that yield high classification performance. The velocities and positions of the particles are updated using a modified PSO algorithm that incorporates information from the best solutions. This update process helps direct the particles toward even better solutions in the search space, as guided by the performance of the previously identified best solutions. After a certain number of iterations of the modified PSO algorithm, the algorithm switches to a genetic algorithm (GA) for further refinement of the feature subset. The GA selects a subset of particles with the highest fitness values and applies a crossover operator, which combines their features to create new particles. The use of a single crossover point in the GA provides simplicity, encourages exploration of different feature combinations, and helps maintain the adjacency between features to prevent premature convergence. The

algorithm continues to iterate through the GA until a stopping criterion is met, such as reaching a predetermined number of iterations or achieving satisfactory performance. At the end of the algorithm, the selected feature subset is the one chosen by the particle with the highest fitness value. This subset represents the most optimal combination of features for accurately classifying hate speech. Finally, the selected feature subset is used to train a classification model. The model learns patterns and relationships from the chosen features and can subsequently classify new data, effectively identifying instances of hate speech based on the knowledge acquired during the optimization process. Applying them will make our data more meaningful and less scarce, and noisy. The computations done by modified PSO and GA can help us efficiently formulate high-quality data features that will improve the results on binary classification between hate and not hate speech. These algorithms were used to improve performance on the benchmark datasets and baseline ML models. Using these selected features, three popular baseline ML models, viz. Support Vector Classifier, Decision Tree Classifier, and Logistic Regression were trained. These baseline models were simple to set up and can provide considerable results in the classification tasks. These models are discussed in the following subsection.

### **3.4. Implementation and Results**

All testing and model creation was performed using Python (version 3.6) [22] and its machine learning (ML) module Scikit-Learn [23]. All computations were performed on Windows 10 Home Edition with a 2.20 GHz Intel Core i7 8750H CPU, 8GB of RAM, and a 4GB NVIDIA GeForce 1050 TI GPU. Google Collaboratory Notebook [24] was employed for some of the complicated calculations and model training. Further in this section, the two datasets collected from the open-sourced repositories are discussed, along with experimentation results obtained after training the ML models on these Datasets, and a brief discussion and comparison report on the results obtained. This section is further divided into three subsections to incorporate all this information.

#### **3.4.1 Dataset Description**

Our research collected two benchmark datasets from Github and hate-speech-data website open-sourced repositories to train and test our proposed approach performance. All these two datasets consisted of three classes Hate Speech and Non-hate and offensive. To conduct this research, non-hate and offensive were merged into one category and make the data into two classes hate and non-hate. Numeric features were extracted using TF-IDF, Word2Vec, and Doc2Vec from these news articles and were used to predict whether these articles were hate or non-hate. Each dataset was trained and evaluated separately on multiple ML models. These datasets are described as follows:



- T.Davidson Datasets: The dataset contains 24,783 tweets that have been separated into three categories: hate speech (5.8 percent), offensive language (77.4 percent), and neither category (16.8 percent). This open-sourced dataset was collected from github. In the final data set, the tweets that were classified as either containing Offensive Language or Neither were included in the Non-Hate Category, whilst the tweets that were classified as either containing Hate Speech or Neither were included in the Hate Category[155].
- Combined Dataset: The following dataset is a mixture of two previously existing datasets. The T.davidson dataset is the first one, and the second dataset has taken from Zenodo website. The two baseline datasets were combined as a solution to the issue of overfitting datasets, which arose from the fact that the existing datasets contained significant inconsistencies on a large scale.

Simple oversampling does not add any new information to the model because it is just duplicating the existing examples, making it vulnerable to overfitting, which can also lead to low bias and high variance results. Therefore, in order to tackle the problem of oversampling, SMOTE was introduced by the author. SMOTE works on the principle of nearest neighbour and evaluates the average of it by considering the examples that are close in the feature space without duplicating the data points. By using this technique, synthetic examples were created using skew and rotation in the feature space rather than duplicating them [156], [157]. In order to balance the dataset, SMOTE is used.

### 3.4.2. Results and Discussion

To compare the performance measure of the proposed algorithm for feature selection and the improvement they showed in the binary classification, experiments were conducted by training and testing two separate sets of ML models. Both the sets contained the same combinations of feature extraction method and ML model. The first set of ML models was trained without deploying the proposed feature selection method, and the second set of models used the proposed algorithm for feature selection. In the subsequent binary classification test, the performance of all trained models was assessed using Accuracy, Precision, Recall, and F1 Score. After these experimentations, we formulated an elaborated report comparing the performances of the two sets of models. Below Equations show the mathematical formulas of these evaluation metrics.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.7)$$

$$Precision = \frac{TP}{TP+FP} \quad (3.8)$$

$$Recall = \frac{TP}{TP+FN} \quad (3.9)$$

$$F1\ SCORE = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3.10)$$

In the above formulas, TP: True Positive, FP: False Positive, TN: True Negative, and FN: False Negative.

Three separate sets of text feature extracted using Word2Vec, Doc2Vec, and TF-IDF were used to train and test the three baseline ML models viz. SVC, LR, and DT. For each set of extracted features, three ML models were trained. This process was repeated for two datasets separately. As for each set of features, these three ML models were trained, we trained a total of nine ML models corresponding to each dataset. Table 3.1 displays the Accuracy, Precision, Recall, and F1 Score of all the baseline models trained without using the proposed algorithm for feature selection. All these models have trained separately again after applying the proposed algorithm for feature selection on the extracted features. Table 3.2 displays the resultant metrics of these models. Along with comparing results in these two Tables 3.1 and 3.2, it was also found which combination of features and ML model performed the best in each dataset.

Given the resultant metrics in Table 3.1 and Table 3.2, it can be observed that all the models trained using the proposed algorithm for feature engineering outperformed simple baseline models in every evaluation metric. From both the Tables, it can be observed that in most cases, Logistic Regression achieved better prediction accuracy and F1 score than the other two classifiers. It is also evident from the Tables that the TF-IDF method of feature extraction performed better than the other two feature extraction methods. The same can be observed in Fig.3.2 and 3.3. These Figs. show the performance of LR with all three feature extraction methods with and without using the proposed algorithm for feature selection on each dataset.

**Table 3.1. Test Result without applying the proposed algorithm on baseline models**

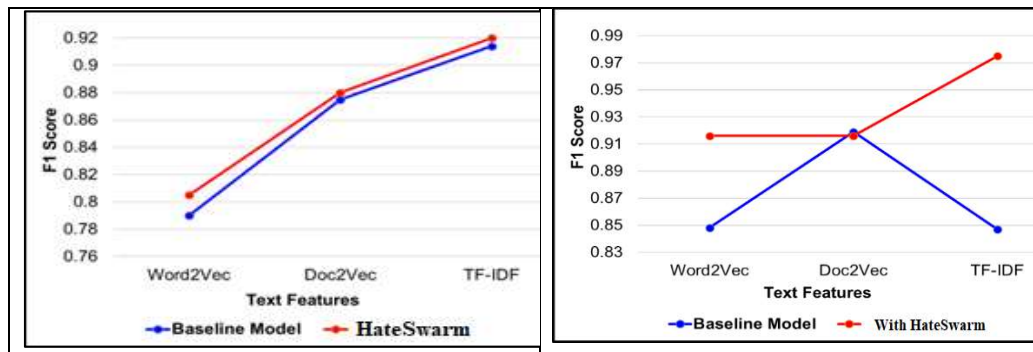
Dataset	Features	Model	Accuracy	Recall	Precision	F1 Score
<b>T.Davidson</b>	<b>Word2Vec</b>	SVC	0.788	0.787	0.787	0.787
		Decision Tree Classifier	0.747	0.747	0.747	0.747
		Logistic Regression	0.789	0.790	0.789	0.789
	<b>Doc2Vec</b>	Decision Tree Classifier	0.720	0.720	0.719	0.719
		Logistic Regression	0.874	0.875	0.874	0.874
	<b>TF-IDF</b>	SVC	0.741	0.741	0.741	0.741
		Decision Tree Classifier	0.856	0.856	0.856	0.856
		Logistic Regression	0.914	0.913	0.914	<b>0.914</b>
	<b>Combined Dataset</b>	<b>Word2Vec</b>	SVC	0.854	0.860	0.865
Decision Tree Classifier			0.867	0.866	0.867	0.867
Logistic Regression			0.847	0.848	0.847	0.847
<b>Doc2Vec</b>		Decision Tree Classifier	0.856	0.854	0.858	0.855
		Logistic Regression	0.919	0.917	0.921	0.919
<b>TF-IDF</b>		SVC	0.920	0.920	0.920	0.920
		Decision Tree Classifier	0.961	0.960	0.961	<b>0.961</b>
		Logistic Regression	0.848	0.845	0.852	0.847

**Table 3.2. Test results after applying the proposed Hate-Swarm algorithm on baseline models**

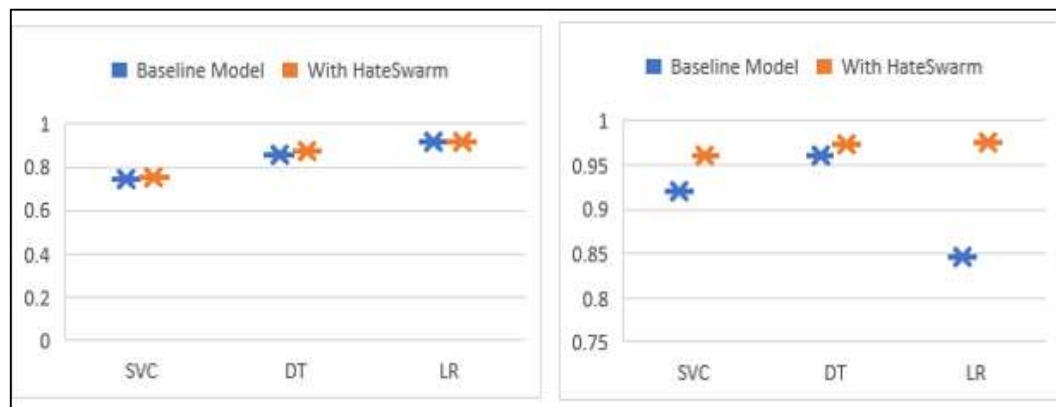
Data	Features	Model	Accuracy	Recall	Precision	F1 Score
T.Davidson	Word2Vec	SVC	0.810	0.810	0.810	0.810
		Decision Tree Classifier	0.782	0.782	0.781	0.781
		Logistic Regression	0.804	0.805	0.804	0.804
	Doc2Vec	Decision Tree Classifier	0.743	0.743	0.743	0.743
		Logistic Regression	0.880	0.880	0.879	0.880
	TF-IDF	SVC	0.752	0.752	0.752	0.752
		Decision Tree Classifier	0.876	0.876	0.876	0.876
		Logistic Regression	<b>0.920</b>	0.920	0.920	<b>0.920</b>
	Combined Dataset	Word2Vec	SVC	0.893	0.897	0.896
Decision Tree Classifier			0.902	0.902	0.902	0.902
Logistic Regression			0.916	0.916	0.916	0.916
Doc2Vec		Decision Tree Classifier	0.859	0.857	0.860	0.858
		Logistic Regression	0.917	0.914	0.921	0.916
TF-IDF		SVC	0.960	0.960	0.960	0.960
		Decision Tree Classifier	0.973	0.973	0.973	0.973
		Logistic Regression	<b>0.975</b>	0.975	0.975	<b>0.975</b>

Similarly, in Fig. 3.2 and 3.3, the comparison of the performance of each ML model trained with and without the proposed approach for feature selection using TF-IDF features is shown for each dataset. From these Figs., it was observed that LR performed better than SVC and DT.

In conclusion, these baseline ML models' overall performance was improved by using the proposed algorithm for feature selection. Out of all the combinations of ML models and feature extraction methods, LR and TF-IDF with HateSwarm algorithm performed the best in all datasets. Using the proposed algorithm for feature selection gave the following results, as shown in Table 3.2. In the T. Davidson hate speech dataset among all the models, the highest classification accuracy and F1 score of 92% and 0.92 were achieved by LR + TF-IDF in combination with HateSwarm. Similarly, in the Combined dataset, LR + TF-IDF + HateSwarm achieved an accuracy of 97.5% and an F1 score of 0.975. Most positive improvements of results were seen in the datasets after using the proposed algorithm for feature selection. In the combined dataset using DT + TF-IDF, LR + TF-IDF features, an increment in both the classification accuracy and F1 score was observed. The same observations can also be seen in Tables 3.1 and 3.2



**Fig. 3.2. Output of Baseline Approached**



**Fig. 3.3. Proposed Approach Output**

#### **3.4.2.1. Discussion**

Based on the results above, it can be concluded that the utilization of the proposed algorithm for feature selection was instrumental in improving the results and delivering quality solutions. The hypothesis that prioritizing data and constructing a data-centric model would enhance results was supported by the experimental findings. The computations carried out by the proposed algorithm facilitated the efficient formulation of high-quality data features, thereby improving the results of the binary classification task. The feature engineering process within our proposed methodology encompassed both feature extraction and feature selection. Through this methodology, the dimensionality of the text data was reduced, allowing for focus on the most appropriate features. This data-centric approach contributed to enhancing the performance of the ML model and decreasing both training time and computational requirements. A maximum classification accuracy of 92% and 97.5% was achieved on the T. Davidson dataset and combined dataset, respectively. Therefore, deploying appropriate feature engineering methods can significantly enhance the results of baseline ML models.

#### **3.5. Conclusion**

Over the past decade, there has been a notable increase in the time spent on social media and other online platforms, both of which have been linked to the propagation of toxic and hateful speech. Numerous studies utilizing advanced Machine Learning and Deep Learning models have been conducted by various researchers in the past, but our research focuses on enhancing data quality, rendering it more appropriate and meaningful, and employing basic ML models for predictions. By utilizing the proposed HateSwarm algorithm for feature selection, the feature optimization problem was efficiently addressed, leading to an enhancement in the model's performance. With our proposed methodology, a 92% accuracy was attained in classifying hate speech within the hate and non-hate dataset. Our proposed approach demonstrated competitiveness with other state-of-the-art Machine Learning models. Further refinement of our work could be achieved in subsequent studies by implementing a more tailored and advanced version of HateSwarm. Additionally, more extensive efforts could be dedicated to the feature engineering aspect. The amalgamation of appropriate feature extraction and selection techniques with advanced ML and Deep Learning models holds the potential for further enhancing results.

## **CHAPTER 4**

### **A Novel Method for Detection of Online Hate Speech using HateDetector**

#### **4.1 Introduction**

Online Hate Speech (OHS) refers to the utilization of offensive and discriminatory language directed at individuals based on attributes such as origin, religion, race, sexual orientation, or socioeconomic class, particularly in the creation of content, blogs, or targeted exploitation on social media. The [67] sheer volume of posts, comments, and messages on these platforms poses a significant challenge in monitoring shared information, despite their role as platforms for individuals to express and exchange ideas [85]. Furthermore, a considerable number of users resort to employing aggressive, unwarranted, and derogatory language when discussing various backgrounds, cultures, and other aspects [139]. This finally leads to inhumanity which may affect the social media users mentally.

People of all ages use social media, and they speak a variety of languages. These languages need to be translated into a common language, such as English, which is commonly referred to as “code.” This is because English is a lingua franca language; more specifically, it is a declarative programming language that is vast, flexible, and standard. As a result, the term “multilingualism” refers to the ability of an individual to speak many languages and the presence of diverse language groups in the same geographic area [158]. Accordingly, in contrast to the general belief, most of the population uses multi-language or two languages at a minimum [159]. The minority population of the World uses a single language which is said to be monolingual. Multilingualism uses several languages as well as code-mixing [60]. Code-mixing is the mashing up of words, sentences, and phrases from two different grammatical systems within the same speech (e.g., Tamil+ English= Tanglish or Hindi+ English = Hinglish)[160].

The code-mixing of Hindi, which is the most widely spoken language in South Asia, with English is known as “Hinglish,” a portmanteau derived from the two language’s names. Hinglish differs significantly from its parent languages in syntax, phonetics, grammar, and even punctuation. The accent and sentiments are Hindi, while the vocabulary is made up of several English (Roman) transliterations of Hindi words, as well as a few English terminologies was the mixing of words, phrases, and sentences from two distinct grammatical (sub)systems within the same speech event [161]. With the wide-reaching popularity of social media platforms, code-mixing has emerged as one of the significant linguistic phenomena among multilingual communities that switched languages [141]. Thus, to detect hate speech, the existing presentations used a meta-learning approach based on metric-based and optimization-based (MAML and

Proto-MAML) methods[161] , Sentiment reversal analysis, Small-sized Transformer model, used several layers of classifiers, RGWE method for sentiment analysis[100], Deep convolution neural network [143], Lateral semantics analysis [162]. Although many methods have been proposed in the past for Hate speech detection, however the major limitation of the existing methods are;

- The research work has been limited to spotting hate in the English Language and few pieces of research in Arabic, Indonesian, Italian, Turkish, Swedish, Albanian Language, and hate content in the rest of the languages like Hindi and Hinglish goes unfiltered.
- In order to furnish research in the field of OHS, a multimodal and multilingual dataset should be developed.
- One of the constraints is the lack of publicly available data towards the progress of online hate speech detection.
- Very few give high-accurate performance profanity check techniques have been introduced.
- Lastly, to reduce the complexity of words caused by incoming datasets while performing detection is not explored.

From the existing reviewed proposals, it was viewed that used many classifiers, concentrated only on a single analysis, hate speech detection was done only for one language, and required improved performance in those models.

In this chapter a novel ‘HateDetector: Multilingual Hate Speech Detection Technique’ has been proposed. In the proposed technique, Bidirectional Encoder Representations from Transformers (BERT) with Multi-Layer Perceptron (MLP) is developed to identify the nature of the tweets by performing the process of code conversion and similarity check that results in good vector values representing the tweet nature. Additionally, the exact sentiment or nature of a tweet, whether hate or non-hate, is identified using the Profanity Check Technique (PCT), composed of ReLu activation function with a logistic regression classifier that classifies the resultant vectors and its respective emoji’s to neutral or hate speech. This technique performs all analyses of a tweet. It also auto-detects and easily finds hate speech, even from poorly written and complex text. The proposed technique performed exceptionally well, with a classification accuracy of 97.9%. is better than the state-of-the-art models.

In this chapter, follow techniques for hate speech detection are proposed;

- Proposed a novel HateDetector that uses BERT-based word embedding with MLP in Multilingual Hate Speech Detection.
- A ‘Profanity Check Technique’ is proposed to extract the emotion of the tweet for a better classification of hate speech.



- The proposed model is trained and tested on three datasets against various models, which yields high performance under different experimentation settings.

## 4.2. Proposed Method

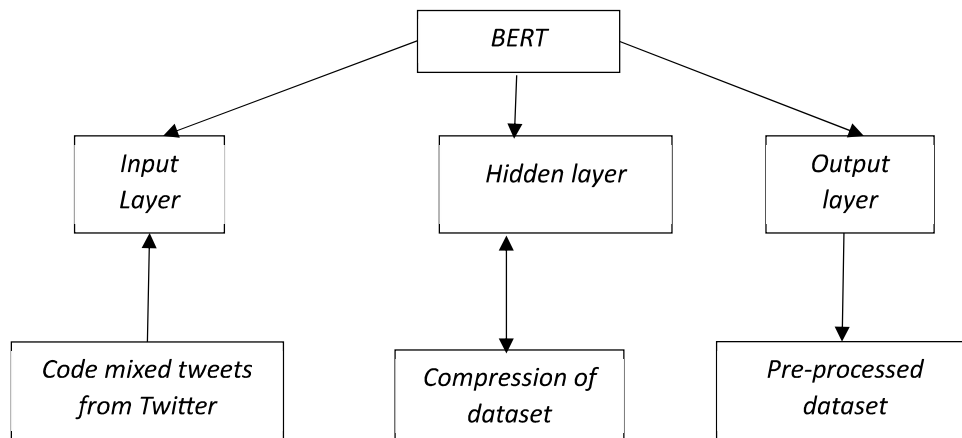
A novel technique was employed, utilizing Bidirectional Encoder Representations from Transformers with a multi-layer Perceptron, to identify the nature of tweets. This involved conducting code conversion and similarity checks, resulting in the generation of vector values that represent the nature of the tweet effectively. Furthermore, the sentiment or nature of the tweet, whether positive or hateful, was determined using PCT, which consists of a ReLu activation function with a logistic regression classifier. This classifier categorized the resultant vectors along with their respective emojis. Twitter has encountered various challenges in managing hate speech in its open space, making it difficult to handle the platform's database and identify instances of Hate Speech. Hence, the 'HateDetector: Multilingual Hate Speech Detection' was proposed, incorporating a Bidirectional Encoder Representation of Transformer (BERT) with a multi-layer perceptron-based word embedding. This pre-processes tweets from Twitter media, and the word embeddings are provided to an mBART model, which converts code-mixed tweets into English. The converted tweet then undergoes BOW with N-gram processing, splitting the sentence into words and assigning weights using numerical values through vector equations. Subsequently, the output is subjected to a similarity checker to verify spelling and grammar. Finally, a Log-likelihood test is conducted to assess positive samples, aiding in the detection of online hate speech and addressing challenges related to dataset utilization and the limitations of high-resource languages.

Additionally, to recognize the emotion conveyed by the sentence, the 'Profanity

Check Technique (PCT)' is utilized, employing a Rectified Linear Unit (ReLu) activation function. The results from the preceding steps are then passed through a sigmoid function, operating between zero and one, and logistic regression is applied to convey the final output, analyzing the nature of the sentence as either positive or negative based on the binary outcomes from the sigmoid function. Consequently, this approach enables the detection and analysis of hate speech across multiple languages and code-mixed language scenarios while preserving the exact emotions expressed within the sentence

### 4.2.1. Bidirectional Encoder Representation of Transformer

The proposed model consists of BERT with a 12-layer transformer network that pre-trains the data set and fine-tunes the tweet [161]. The pre-trained tokens are provided to each layer, a token from each layer is used as a token for the next layer, and this word embedding uses a neural network. Multi-layer Perceptron embedded within the BERT is a feed-forward artificial neural network with one input layer and 768 hidden layers and one output layer that processes the input data by backpropagation.

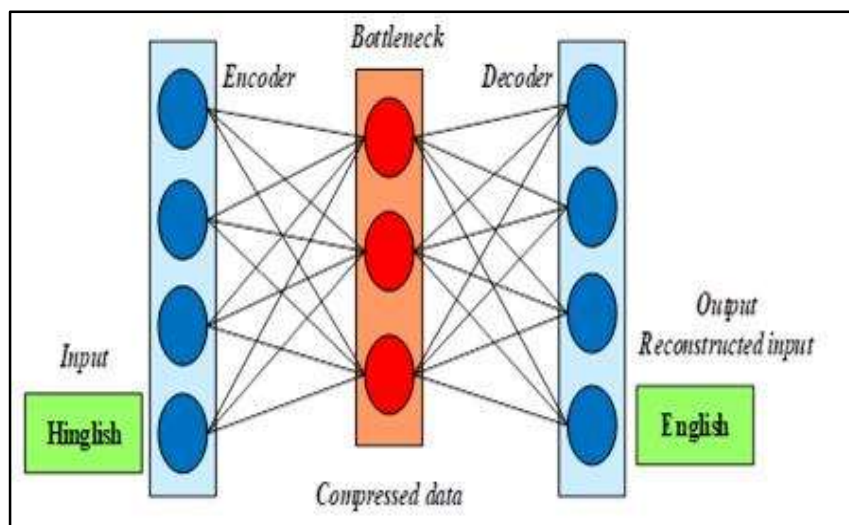


**Fig. 4.1. Pre-processing of tweets in Twitter datasets by BERT**

In Fig. 4.1, the data set assigned for the task is given into the input layer of the BERT, where the input data is fine-tuned and pre-trained in the hidden layers composed within the BERT, and the key phrases are given out through the output layer. Here, the input layer sends the data set, the upcoming hidden layer compresses the tweets fed into it, and the pre-processed key phrases are given out in the output layer. The BERT holds many pre-trained tasks that helps to perform well. Algorithm 1 shows the pre-processing flow where the tweet is cleansed by converting the lower to upper case and removing the stopping notations, URLs, tags. The tweet is split into each character, and the BERT model starts reading each character compared with the pre-trained data set.

#### 4.2.2 mBART: Multilingual Encoder-Decode

The pre-processed and pre-trained output from BERT is given as an input to mBART that denoises the multilingual text by encoding it sequence by sequence, Hinglish is converted into English since English is the code for the NLP, which fine-tune any of the supervised or unsupervised pair of languages without any specific modification in task with one trained set of parameters for all languages even if it is complex[158]. The process flow of mBART is shown in Fig. 4.2, an-autoencoder converting the output from BERT to the reconstructed code output by compressing the key phrases into their original language and then converting it into the code language understandable by the machine. Example: String: "Je déteste cette personne, elle est très méchante." The sentence "I hate this person, she is very mean." was translated from French to English. Subsequently, the mBART model was utilized to determine whether this sentence contains hate speech.



**Fig. 4.2. Process flow of mBART [37] for proposed technique**

Initially, preprocessing was conducted on the provided text to prepare it for input into the mBART model. This preprocessing encompassed tokenization of the text, conversion of each token into numerical sequences, and padding of the sequences to a predetermined length. Within this research, the text was tokenized into individual words, each word was converted into its corresponding numerical index within the mBART vocabulary, and the sequence was padded to a fixed length of 50 tokens.

Tokenized text: ['Je', 'déteste', 'cette', 'personne', ',', 'elle', 'est', 'très', 'méchante', '.']

Numerical sequence: Next, the tokenized sequence were converted into a numerical sequence by mapping each subword to its corresponding numerical index in the mBART vocabulary. This gives us the following numerical sequence: [27, 30888, 618, 4480, 18, 299, 51, 529, 31416].

Since mBART processes sequences of fixed length, therefore, padding the numerical sequence to a fixed length is required. In this experiment, the padding of 10 is used. Then, pad the sequence with the special padding token, which has a numerical index of 1, to get the following padded sequence: [27, 30888, 618, 4480, 18, 299, 51, 529, 31416, 1]. Now, we fed the padded numerical sequence into the mBART encoder network to get the fixed-length representation of the input sequence. The encoder network applies a series of self-attention and feedforward layers to the input sequence to compute a sequence of hidden states. The final hidden state, which summarizes the entire input sequence, as the fixed-length representation. The fixed-length representation was denoted as  $p$ .

$p = \text{Encoder}([27, 30888, 618, 4480, 18, 299, 51, 529, 31416, 1])$ .

Finally, the fixed-length representation  $z$  was fed into the mBART decoder network to generate the output sequence in the target language. A series of self-attention, encoder-decoder attention, and feedforward layers were applied by the decoder network to the input representation to compute a sequence of hidden states. These hidden states were then utilized to generate the output sequence token by token. As the input sentence was in French, the target language was set to English, which serves as the code language for the mBART model. The output sequence was denoted as  $y$ .

$$y = \text{Decoder}(z, [2]) \quad (4.1)$$

Here,  $[2]$  is the numerical index for the special start-of-sequence token in English. The decoder network generates the output sequence by predicting the next token in the sequence given the previous tokens and the input representation  $z$ . The output sequence is terminated by the special end-of-sequence token, which has a numerical index of 0 in English.

#### 4.2.3 BOW with N-gram

Code from mBART is given into BOW with N-gram that splits words from the sentence. Features are extracted by BOW that tokenizes the sentence into words and then these tokens are taken as an array of the frequency called it as bow vector; finally, these vectors are mapped at a fixed dimension in the space given. In simple words, a positive, negative or neutral sentence is converted to numbers by the following methodology of mapping the words.

Let us take all the sets of finite sequences of words as  $A$  and pre-trained dictionary labels as  $D$ . Mapping of finite set  $A$  to the dimensional feature space is done by

$$\emptyset: A \rightarrow R^M \quad (4.2)$$

The sentiment label set is taken as  $\gamma = \{1, \dots, K\}$ , assuming  $K = 2$  shows whether it is positive or negative. The labeled trained data set trains  $\gamma$  and the input dataset is given as  $x = (w_1, \dots, w_N)$  where  $N$  is the input sequence length. In bags of n-gram  $\emptyset(x)$  maps  $x$  to  $M = |\Gamma|$  where  $M$  is dimensional latent space where  $\Gamma$  is the vocabulary of n-grams and  $\Gamma$  is evaluated using  $|\Gamma| = O(|D|^n)$ .

The embedding of the  $\gamma_j$  is evaluated by,

$$P_{\gamma_j} = h(F \times z_{\gamma_j}) \quad (4.3)$$

where  $F$  : projection matrix

$$h(.) = \tanh(.)^3$$

$$\gamma_j = (w_j, w_{j+1}, \dots, w_{j+n-1})$$

$F$  : projection matrix

$z_{\gamma_j}$  is an operator string of trained set;

Finally, the vector representation of a document is represented as

$$\emptyset(x) \equiv d_x = 1/N \sum_{j=1}^N P_{\gamma_j} \quad (4.4)$$

where,

$$d_x \in \mathbb{R}^M \text{ and } x = (w_1, \dots, w_N)$$

The reason for formulating this evaluation is to fix the length  $n$  of the sentence with the number of phrases.

A simple  $N$ -gram calculation is done by

$$Ngram_K = X - (N - 1) \quad (4.5)$$

where,

$$Ngram: \text{Weightage}, K: \text{Given sentence}, X: \text{Number of words}$$

Equations 4.3 and 4.4 help in the easy and correct calculation of  $n$ -grams. The output of the  $n$ -gram is an  $n$ -gram dictionary with an actual valid  $n$ -gram id,  $n$ -gram phrases, document frequency, length of  $n$ -gram, global term frequency, and other information [108]. We use these details to find the output in the next step of our process.

#### 4.2.4. Similarity Checker and Log-Likelihood Test

The spelling mistakes in the respective input were detected, removed, and corrected by the similarity checker using the  $n$ -gram dictionary. A computer-programmed similarity checker, Grammarly, and plagiarism checker were utilized to compare the present vocabulary with the pre-existing one. Additionally, a Log-Likelihood Test was conducted to assess whether the output of the checker was satisfactory or not. If the log-likelihood value surpassed that of the dataset, it indicated a better fit for the model. The log-likelihood value ranges from negative infinity to positive infinity. Values are obtained by equation 4.6,

$$l(\theta) = \ln L(\theta) \quad (4.6)$$

where  $L(\theta)$ : the output of the similarity checker,  $l(\theta)$ : the output of the log-likelihood test

From equation (4.5), the validated vocabularies are given out as output, but it is not precise in its emotion analysis as the classification of words may fail to detect the

nature of the sentence correctly. Thus “Profanity Check Technique” is implemented which detects the correct emotion of the sentence, whether it is sad or happy, or angry.

#### 4.2.5 Profanity Check Technique

To overcome the expression issue, the PCT is proposed to verify with its library whether the string received has negativity or not by using Rectified Linear Unit (ReLU) that activates the works on the multi-layer neural network. This Deep Learning model is combined with sigmoidal function and logistic regression. Though the Sigmoidal function activates the vectors provided, it does not have vanish gradient issues. ReLU vanishes this gradient as it does not use any transforms in its calculation.

ReLU function  $f(x)$  is calculated by

$$f(x) = \max(0, x) \quad (4.7)$$

From equation (3.6) the output of the function is found. As if it returns 0 it is negative, or else it is positive. Though, complex functions cannot be learned using this ReLU function which may lead to regression problems. A generalized linear classifier model known as logistic regression is implemented to overcome this issue. That undergoes supervised learning with a labeled dataset and analyzes pre-trained binary data where the output is continuous with the parameters and summing of input and brings out the relation between independent and dependent variables. Logistic regression uses the sigmoid activation function to take back the label’s probability to code any value between 0 and 1. The S curve shows that the output is negative, where it cannot go behind 1. This is calculated using the equation 4.8 below.

$$\text{Loglikelihood test} = 2\log_e\{L_s(\theta) / L_g(\theta)\} \quad (4.8)$$

Where,

$s$ : pre-processed sample,  $g$ : pre-trained parameters

From equation (4.7) the output of the log-likelihood test is evaluated by giving the sample attained with the pre-trained dataset available.

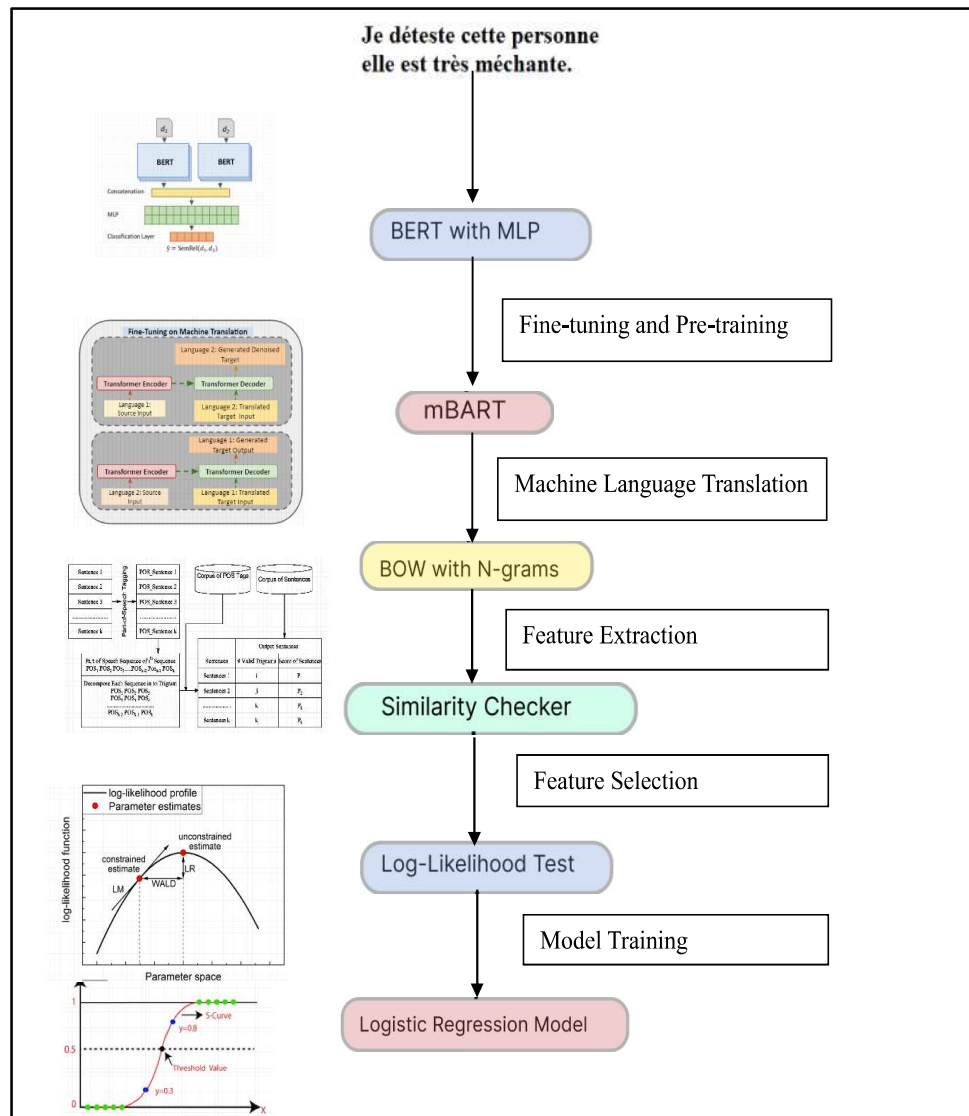
The Loglikelihood test, in conjunction with the Profanity Check Technique, was utilized to enhance the performance of the hate speech detection model. While the PCT served to identify profane words or phrases, it might not have been sufficient to accurately identify hate speech, as not all instances of hate speech contain profanity. The log-likelihood ratio test, a statistical method, was employed to ascertain the likelihood of a given text being hate speech based on the frequency of certain words or phrases in hate speech compared to non-hate speech. By merging the PCT with the log-likelihood ratio test, the model could potentially achieve a higher accuracy in detecting hate speech. The novel PCT algorithm 2 was proposed for addressing the OHS problem.

#### 4.2.6 Proposed Methodology

The "HateDetector: Multilingual Hate Speech Detection" model consists of several steps that help in detecting and analyzing hate speech. Firstly, the model uses a Bidirectional Encoder Representation of Transformer (BERT) to pre-process the tweets from Twitter media. BERT is a powerful natural language processing model that can understand the context of the text and produce high-quality embeddings for text. The word embeddings produced by BERT are then given to the mBART model, which converts the code-mixed tweet to English. This conversion helps in ensuring that the model can detect hate speech in multiple languages and code-mixed languages. After the tweet has been converted to English, it is given to Bag of Words (BOW) with N-gram. BOW is a technique that splits the given sentence into words and weighs each word using numbers by mapping the sentence provided using vector equations. This technique helps in detecting and identifying the frequency of the occurrence of specific words that may be associated with hate speech. The output of BOW is then fed to the similarity checker to check the spelling and grammar of the tweet. This step ensures that the model can detect and analyze hate speech that may contain incorrect spelling or grammar. Finally, the model employs the Profanity Check Technique (PCT) with Rectified Linear Unit (ReLU) activation function to recognize the emotion of the sentence. PCT is a technique that uses a set of profane words to identify the emotion associated with a sentence. The output of the PCT layer is then given to a sigmoid function that works between the zeros and ones. The sigmoid function provides binary outcomes that indicate whether the sentence is positive or negative. Then finally the logistic regression layer transmits the final output by analyzing the nature of the sentence, whether it is positive or negative, and identifying any hate speech in the sentence. The proposed model provides an effective solution to the challenges faced by Twitter in managing and detecting hate speech in the open space. The model employs various techniques such as BERT, Multilayer Perceptron-based word embedding, and Profanity Check Technique (PCT) to detect and analyze hate speech in multiple languages, code-mixed language, and identify the emotions associated with the sentence. The model's ability to detect and analyze hate speech in various languages and code-mixed language makes it a valuable tool in promoting a safe and healthy social media environment. A multilingual hate speech detection that holds up a BERT with multi-layer perceptron-based word embedding. It detects hate speech of tweets with non-linguistic concepts. This is done with the help of several tests and functions performed within the multilingual hate speech detection technique. The classification performance is divided into a trained, test, and validation set. This training set is used as a reference for the pre-processed dataset, the test set is to optimize the proposal's approach, and the validation set is to evaluate the proposed technique. Fig. 3.4 shows the process flow of the proposed model. Initially, a code-mixed tweet is given to the BERT with Multilayer Perceptron (MLP), which consists

of several encoders and decoders to pre-process the mixed tweets and remove all the unnecessary stopping words and tags. All the hidden layers are used to compress both text and image datasets. The pre-processed data is entered into the mBART (multilingual encoder-decode), which de-noise the key-phrased dataset again. The BOW with n-gram maps the vectors in the dimensional feature space and splits according to the gram. A similarity checker is used to gamble the vocabulary that collects the data, analyses the meaning, confirms it with the trained data available, and investigates the speech. The logarithmic value of the log-likelihood test performs the process of sensing the Positive sentiments of the words. However, it could not reach the correct sense of the sentence, so PCT with activation function and classifier is performed with the sigmoidal flow of zeros and ones, which precisely identifies the nature of the sentence and detects the hate speech. Fig. 4.3 shows the process flow of the proposed model. Initially, a code-mixed tweet is given to the BERT with Multilayer Perceptron (MLP), which consists of several encoders and decoders to pre-process the mixed tweets and remove all the unnecessary stopping words and tags. All the hidden layers are used to compress both text and image datasets. The pre-processed data is entered into the mBART (multilingual encoder-decode), which de-noise the key-phrased dataset again. The BOW with n-gram maps the vectors in the dimensional feature space and splits according to the gram. A similarity checker is used to gamble the vocabulary that collects the data, analyses the meaning confirms it with the trained data available and investigates the speech. The logarithmic value of the log-likelihood test performs the process of sensing the Positive sentiments of the words. However, it could not reach the correct sense of the sentence, so PCT with activation function and classifier is performed with the sigmoidal flow of zeros and ones, which precisely identifies the nature of the sentence and detects the hate speech. In the next section, we discussed our proposed methodology in a detailed fashion.





**Fig. 4.3. Block Diagram of the proposed technique to detect hate speech**

In the pre-processing section, the given data set is processed before it is given as input to the mBART. The dataset consists of tweets which further consist of Emojis to express emotions, tags, stopping notations, uppercase letters, and URLs are cleansed initially from each tweet by the algorithm 4.1.

---

**Algorithm 4.1. Pre-processing of Tweets for the Hate speech detection**


---

**Input: text:** Annotated dataset

**Output: text:** Pre-processed dataset

*def clean\_text(text):*

*## convert upper case words to lower case*

*text = text.lower ().split ()*

*## Remove stop words*

*stops = set (stopwords.words('english'))*

*text = [w for w in text if not w in stops and len(w) >= 3*

*text = " ".join(text)*

---

*## Clean the text*

*text = re.twt (r' https?:// [A - Za - z0 - 9./] + ', 'url', text)*

*text = re.twt (r" [^A - Za - z0 - 9^,!. \ /' + - =]", " ", text)*

*text = re.twt (r"what's", "what is ", text)*

*text = re.twt (r" \'s", " ", text)*

*text = re.twt (r" \'ve", " have ", text)*

*text = re.twt (r" n't", "not ", text)*

*text = re.twt (r" i'm", "i am ", text)*

*text = re.twt (r" \'re", "are ", text)*

*text = re.twt (r" \'d", "would ", text)*

*text = re.twt (r" \'ll", "will ", text)*

*text = re.twt (r" ,", " ", text)*

*text = re.twt (r" \.", " ", text)*

*text = re.twt (r"!", "!", text)*

---

*for (\/, ^, +, -, =, ', :) loop \_ do*

*text = re.twt (r" (\d+)(k)", r"\g < 1 > 000", text)*

*text = re.twt (r" e g", "eg ", text)*

*text = re.twt (r" b g", " bg ", text)*

*text = re.twt (r"u s", " american ", text)*

*text = re.twt (r" \0s", " 0", text)*

*text = re.twt (r" 9 11", "911 ", text)*

*text = re.twt (r" \e - mail", " email", text)*

*text = re.twt (r" j k", "jk ", text)*

*text = re.twt (r" \s{2,}", " ", text)*

*text = re.twt (r'@[A - Za - z0 - 9] + ', ', text)*

*text = re.twt (r" (\w)\1{2,}', r'\1\1', text)*

*text = re.twt (r" \w(\w)\{2}', ' ', text)*

```

text = EliminateStop(text)
return text

```

---

```

def del_NonAlphaWords(sentence):
    return ''.join ([word for word in sentence.split

```

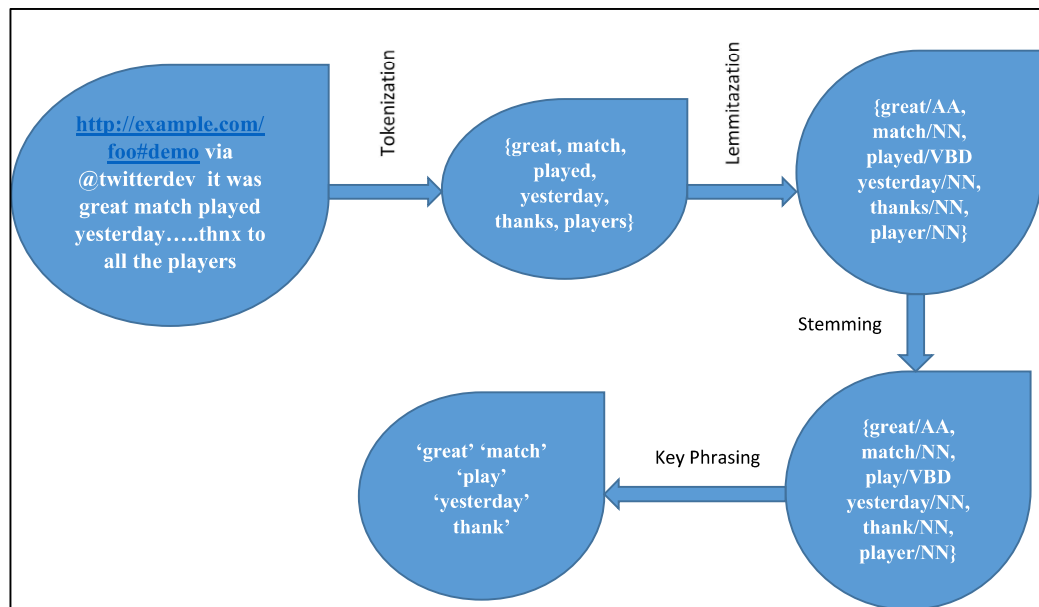
**Algorithm 4.2.** Short -Pre-processing of Tweets for the Hate speech detection

Input: Tweet text

Output: Cleaned tweet text

1. Convert all characters to lowercase
2. Remove URLs and mentions (words that start with '@')
3. Remove hashtags but keep the words after them
4. Replace contractions (e.g., "can't" -> "cannot")
5. Replace slang and informal words with their standard equivalents (e.g., "u" -> "you")
6. Remove non-alphabetic characters and punctuation, except for exclamation and question marks
7. Tokenize the text into individual words
8. Remove stop words (common words that don't carry much meaning, such as "the" and "and")
9. Perform stemming or lemmatization to reduce words to their base form
10. Combine the remaining words back into a single string and return the cleaned text.

In the pre-processing section, the given data set is processed before it is given as input to the mBART. The dataset consists of tweets which further consist of Emojis to express emotions, tags, stopping notations, uppercase letters, and URLs are cleansed initially from each tweet by the above algorithm 4.1. Whether the text contains hate or not, these embellishments add nothing to the text. The next step of pre-processing the tweet is tokenization and lemmatization, where another specific symbol replaces the sensitive data without giving up the security level and the tokens are lemmatized and finally, the output is stemmed and key-phrased where plurals are removed and forms as a key-phrase sentence that is stored and used when it is needed.



**Fig. 4.4. Systematic representation of Preprocessing of the Twitter Dataset**

The text contains hate or not, these embellishments add nothing to the text. The next step of pre-processing the tweet is tokenization and lemmatization, where another specific symbol replaces the sensitive data without giving up the security level and the tokens are lemmatized and finally, the output is stemmed and key-phrased where plurals are removed and forms as a key-phrase sentence that is stored and used when it is needed. Algorithm 4.2 is the internal flow of logistic regression where the classifier finally identifies the nature of the sentence by splitting the tweets into the matrix and augmented matrix. On checking its mode, it identifies whether the tweet is positive or negative and the output is printed.

#### **Algorithm 4.3. PCT for OHS**

**Input:** *twt*: A text string

**Output:** *decision*: A binary label indicating whether the text contains profanity or not

**Begin**

**1. Input:** A string of text

**2. Perform pre-processing on the input text:**

- a. Convert the text to lowercase
- b. Remove all punctuation marks
- c. Tokenize the text into individual words
- d. Remove stop words
- e. Perform stemming and lemmatization

3. Use a pre-defined list of profanity words and compare each word in the pre-processed text to the list of profanity words.
  4. If any profanity words are found, assign a score of 1 for each profanity word.
  5. Calculate the loglikelihood score for the input text using a pre-trained language model:
    - a. Tokenize the pre-processed text into a sequence of tokens
    - b. Use the language model to predict the probability distribution of the next token given the previous tokens.
    - c. Calculate the log-likelihood score of the input text based on the predicted probability distribution
  6. Combine the PCT score and the loglikelihood score to classify the input text:
    - a. If the PCT score is greater than a pre-defined threshold AND the loglikelihood score is negative,
      - classify the input text as hate speech
    - b. If the PCT score is greater than a pre-defined threshold but the loglikelihood score is positive,
      - classify the input text as profanity
    - c. If the PCT score is below the pre-defined threshold, classify the input text as non-offensive
- 

### 7. Output the classification of the input text.

Example: for the above Algorithm

Input: "I hate this person, she is very mean"

Pre-processed text: "hate person mean"

Profanity words: ["hate", "mean"]

PCT score: 2

Loglikelihood score: -7.5

Threshold: 1.5

Since the PCT score is greater than the threshold and the loglikelihood score is negative, the input text is classified as hate speech.

A brief illustration of the implementation of BERT with MLP is given below.

- Preprocess the input data: We have tokenized the input into individual words or subwords, converted it into numerical sequences, and padded it to a fixed length. It involves breaking the input text into individual words or subwords. We have used a tokenizer library such as Tokenizer from the Transformers library in Python. We have created an instance of the tokenizer class and used its encode method to tokenize the input text into a list of tokens.
- Load the pre-trained BERT encoder: Then we used TensorFlow to load the pre-trained BERT model.

- Extract features using BERT encoder: In this step, we have passed the preprocessed input sequences through the BERT encoder to obtain the output features for each token. BERT outputs a sequence of vectors of dimension  $d$ , where  $d$  is the hidden size of the BERT model.
- Feed features into an MLP classifier: We flatten the sequence of output features to obtain a fixed-size feature vector for the entire input sequence and feed this vector into an MLP classifier. The MLP consists of one or more hidden layers, each with a nonlinear activation function such as ReLU, and a final output layer with the appropriate number of units for the task-specific prediction.
- Train and evaluate the BERT-MLP model: We have trained the model on the labeled training data using an optimization algorithm such as Adam and evaluated its performance on a held-out validation set. Then model is fine-tuned by adjusting the hyperparameters such as learning rate, number of hidden layers, and batch size.

The final output from the above algorithm is then passes to the logistic regression algorithm. It is a machine learning algorithm that is commonly used for binary classification tasks. It takes a set of input features and learned a set of weights for those features that can predict the probability of a given example belonging to a particular class (in this case, the class of hate speech vs. non-hate speech). Once the input text has been preprocessed and features have been extracted using Bag-of-Words with Ngram, logistic regression is trained on these features to learn the weights that best predict the probability of a given example being classified as hate speech. During training, the logistic regression algorithm iteratively adjusted the weights of the features until it achieves the highest possible accuracy on the training data. Once the model has been trained, it is used to predict the probability of new examples belonging to each class. In the case of hate speech detection, the logistic regression model would output a probability score for each input text indicating the likelihood that it is hate speech. The output of the PCT with Loglikelihood test is used as input features for the logistic regression algorithm, which can then learn to predict the probability of a given example being classified as hate speech.

So, it is clear that the proposed technique utilizes high resource language for hate speech detection and a much more pre-trained data set is placed in the dataspace with the help of a multi-layer perceptron embedded with BERT. All the forms of text, images, and emojis are taken into account to detect the OHS, and finally the emotion is efficiently predicted with the help of the above profanity check technique.

#### **4.2.7. Data Set**

The hate speech detection uses the Multilingual Hate Speech Technique, which uses BERT with a multi-layer neural network that is further encoded with PCT to hold the emotion of the data set. The methodology carries tweets on Twitter media as the dataset for performing the methodology.

- T. Davidson [19]: The dataset consists of 24,783 tweets divided into three categories: Hate speech (5.8%), Offensive Language (77.4%) and Neither (16.8%). This open-sourced dataset was collected from GitHub. The tweets labeled as Offensive or Neither were included in the Non-Hate Category. Furthermore, the data set column labeled as having Hate Speech was included in Hate Category.
- Hinglish Hate Speech Dataset: The first set of data was drawn from the paper [163]. Two linguists, one fluent in Hindi and the other fluent in English, annotated the tweets that were gathered through the Twitter API. Data from the research classification of Offensive Tweets in the Hinglish Language was used for the second batch of data. As a result of our current task, the dataset has been relabeled into two classes: Not-Abusive and Hate-inducing categories have been labelled as hate, while the non-offensive category has been relabeled as not hate. The final piece of data comes from the HASOC task, which was a collaborative effort.
- Hot (Hate Offensive Text) : It was collected by using the Twitter Streaming API, in which the tweets had more than three Hinglish words. These tweets were then manually annotated. This collection of tweets covered the period spanning November 2017 and February 2018. In order to limit the scope of our data mining to a particular geographic region, we only considered tweets that originated from the Indian subcontinent. It has been shown that in a Hindi offensive Text (HOT) data set with many numbers, nearly 24779 tweets are taken for pre-processing. The tweet has been divided into Hate speech, offensive language, and other classes. Then the given tweets are pre-processed by the BERT with a multi-layer perceptron in the Multilingual Hate-speech Detection Technique.

### 4.3. Experimental Set up and Implementation

BERT with a multi-layer perceptron-based word embedded system has been recently used by most researchers who undergo research on hate-speech detection. The technique used in this study was a Multilingual hate speech detection technique with word embedding and detected hate speech on the Internet. The proposed technique is used to detect the pre-processed dataset's vocabulary. Good words have been evaluated, and hate speeches have been detected by sensing the exact emotion of the sentence formulation and reporting an individual or a group. Python programming language (version 3.6) and its machine learning (ML) module Scikit-Learn was utilized for all experiments and model construction. All calculations were performed on a system with an Intel(R) Core i7 8750H CPU 2.20 GHz processor, 8GB RAM, and a 4GB NVIDIA GeForce 1050 TI Graphics Processing Unit running Windows 10 Home edition. Google Collaboratory Notebook was utilized for certain intensive computations and model training. Further in this section, we have discussed the four datasets collected from the open-sourced repositories, the experimentation results we obtained after training the proposed technique on these Datasets, and a brief discussion

and comparison report on the results obtained. The tweets are pre-processed using BERT, as the upper-case letters are converted into lower-case letters, removal of tags, URLs, and stopping notations makes the machine easy to understand and validate the given tweet. The first part of the processed tweet holds the tweets with the capitalization of letters and the second part comprises pre-processed tweets without capitalization and has no tags like #, @, and so on. This simulation result has been obtained by applying the proposed technique. In the above simulation, the tweets have been pre-processed using BERT. Followingly, the Hinglish comment has been given to the mBART to convert them into English. Then forwarded to the similarity checker where the incomplete or misspelled words are vocabulary and spell-checked using an online checker forwarded to the log-likelihood test where the tweet's negative, positive or neutral nature is verified.

Additionally, the sentiment of the tweet has been analyzed by the Profanity check technique. Here, with ReLu, sigmoid function, and logistic regression, the last validated words are compared with the nearby words in the sentence. The emotion has been found out whether it's negative, positive, or neutral. Finally using the above simulation, the input tweet has been classified into three sub-categories. When we come to Hate speech, 4744 words are optimized with 38419 reference words, and from those, 4259 words are validated.

#### 4.3.1 Performance Evaluation Metrics

The performance of the proposed technique has been evaluated in this subsection with parameters [164] such as Precision, Recall, Accuracy, and F1-score. Then this performance evaluation graph comprises the result with other techniques such as Bidirectional Encoder Representation from Transformer- Convolutional Neutral Network (BERT-CNN), Convolutional Neutral Network (CNN), Bidirectional Long-Short Term Memory (Bi-LSTM), Ternary Trans-CNN, Deep Learning (DL) Ensemble Stacked, Embeddings from Language Model(ELMO). Equations (4.9-4.12) show the mathematical formulas of these evaluation metrics.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4.9)$$

$$Precision = \frac{TP}{TP+FP} \quad (4.10)$$

$$Recall = \frac{TP}{TP+FN} \quad (4.11)$$

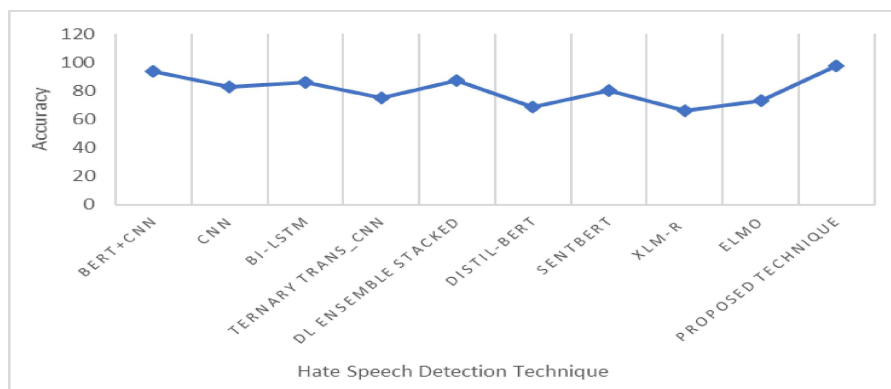
$$F1\ SCORE = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4.12)$$

Where, TP: True Positive, FP: False Positive, TN: True Negative, FN: False Negative.

##### 4.3.2.1 Accuracy of Hate-Detector technique with previous Implemented techniques



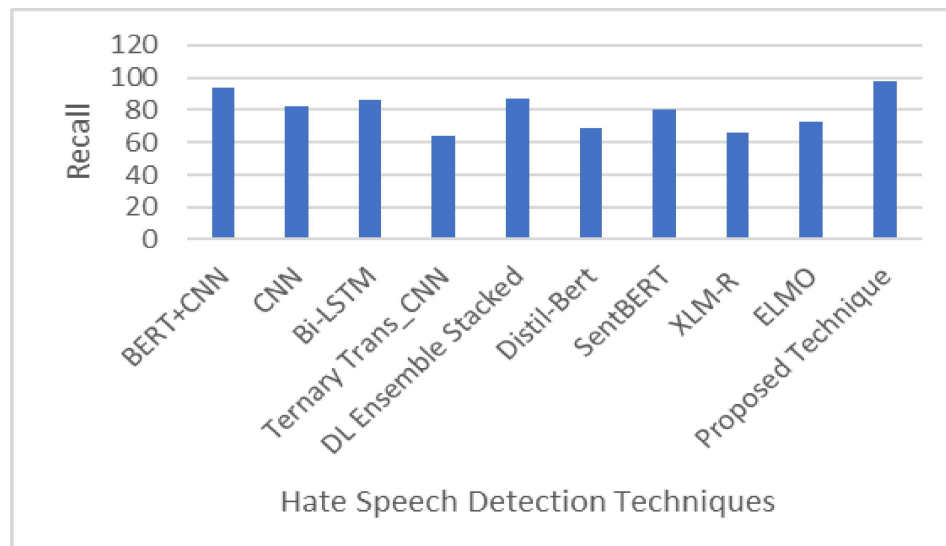
The degree of achieving the standard level of the calculation is accuracy by using Equation 4.8. From the result, it has been clear that the accuracy attained to 97.6% when it has been analyzed with the existing technique like BERT-CNN has 94% accuracy, CNN has 82.7% accuracy, Bi-LSTM has 86.2% accuracy, Ternary Trans-CNN has 75% accuracy, DL Ensemble Stacked has 87.3% accuracy, Disttil-BERT has 69%, SentiBERT has 80%, XLR-R has 66% and ELMO has 73% accuracy. On over-viewing it, the proposed technique reaches high in its accuracy and ELMO Technique was low in its accuracy. In Fig. 4.5, the performance of the proposed technique based on accuracy is shown.



**Fig.4.5. Accuracy result of the Hate-Detetctor technique with previous Implemented techniques**

#### **4.3.2.2 Recall of Hate-Detetctor technique with previous Implemented techniques**

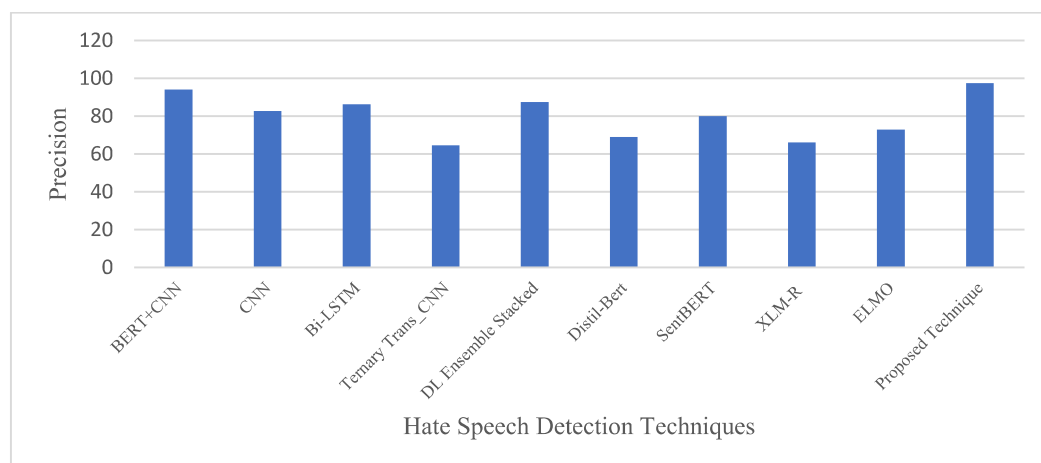
The capability to identify the positive tweets is recall. From the result, the recall value by using Eqn. 10 of the proposed technique attained is 97.6% when it has been analyzed with the existing technique, BERT-CNN has 94%, CNN has 82.7%, Bi-LSTM has 86.2%, Ternary Trans-CNN has 64.4%, DL Ensemble Stacked has 87.3%, Disttil-BERT has 69%, SentiBERT has 80%, XLR-R has 66%and ELMO has 73%. On over-viewing it, the proposed technique performs high in recall and the Ternary Trans-CNN Technique is low in its recall.



**Fig. 4.6. Recall of the HateDetector technique with previous Implemented techniques**

#### 4.3.2.3 Precision of Hate-Detector technique with previous Implemented techniques

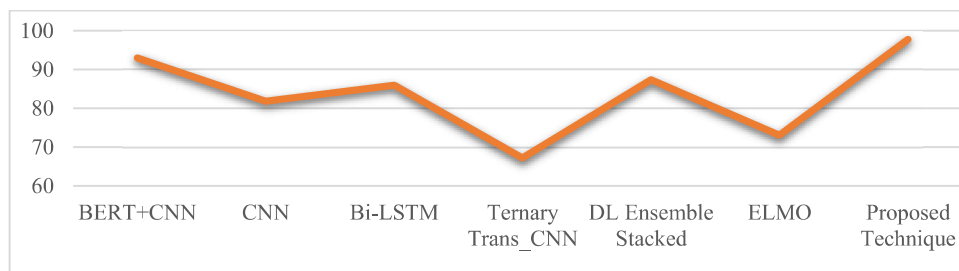
The quality measure of positive samples is called precision. From the result, the precision value by using Eqn. 4.9 of the proposed technique has been attained to 97.5% when it has been analyzed with the existing technique, BERT-CNN has 94%, CNN has 82.7%, Bi-LSTM has 86.2%, Ternary Trans-CNN has 64%, DL Ensemble Stacked has 87.4%, Distil-BERT has 69%, SentiBERT has 80%, XLR-R has 66% and ELMO has 73%. The results show that the proposed technique gives high value of precision and the Ternary Trans-CNN Technique was low in its precision.



**Fig. 4.7. Precision result of the HateDetector technique with previous Implemented techniques**

#### 4.3.2.4 F1 score of Hate-Detector technique with previous Implemented techniques

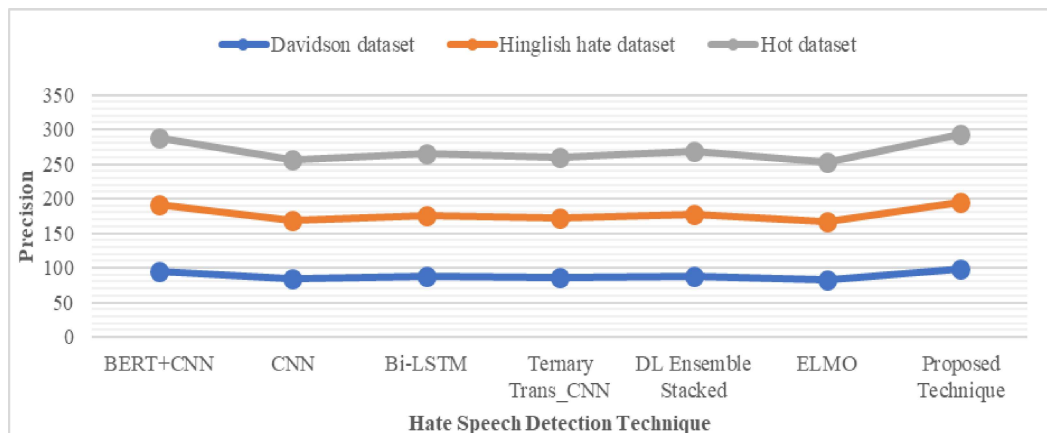
The proposed technique has achieved 97.8% when it has been analyzed with the existing technique. The other technique like BERT-CNN has 94%, CNN has 82%, Bi-LSTM has 86%, Ternary Trans-CNN has 71%, DL Ensemble Stacked has 88%, Disttil-BERT has 69%, SentiBERT has 80%, XLR-R has 66% and ELMO has 73% on over-viewing it, proposed achieved high in F1-Score performance and Ternary Trans-CNN Technique is low in its F1-Score.



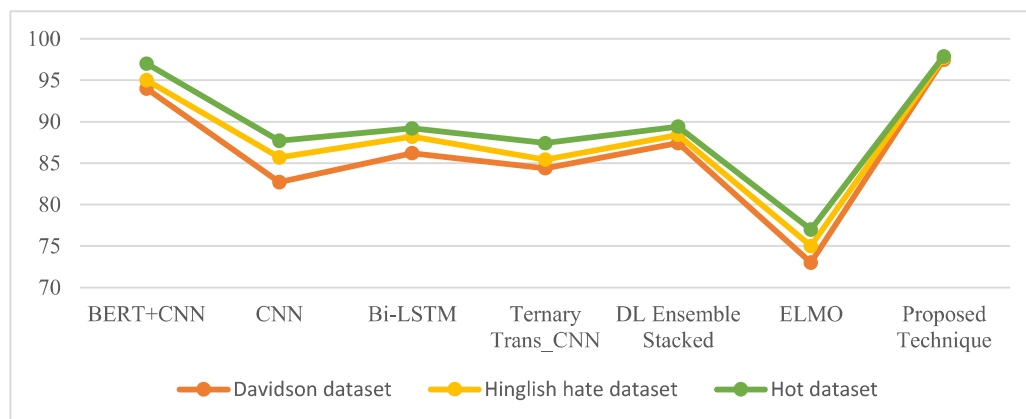
**Fig.4.8. F1-Score result of the HateDetector technique with previous Implemented techniques**

#### 4.3.3. Comparative Analysis with the state-of-the-Art-Methods

This subsection compares three different types of data sets with proposed and existing Techniques BERT-CNN, CNN, Bi-LSTM, Ternary Trans-CNN, DL Ensemble Stacked, and ELMO. The comparative analysis of proposed and existing techniques based on three different types of data sets: Davidson Dataset, the Hinglish Hate Dataset, and the HOT dataset are shown in Fig 4.10. The three types of datasets maintain the same level of accuracy in the proposed technique when compared with the other existing technique. Each technique taken for comparison has undergone certain deviations from its accuracy value for all three data sets. Here too, the accuracy does not change. The proposed technique has achieved high in its accuracy without any such deviations and the Ternary Trans-CNN and ELMO techniques have undergone more deviations when compared with others.

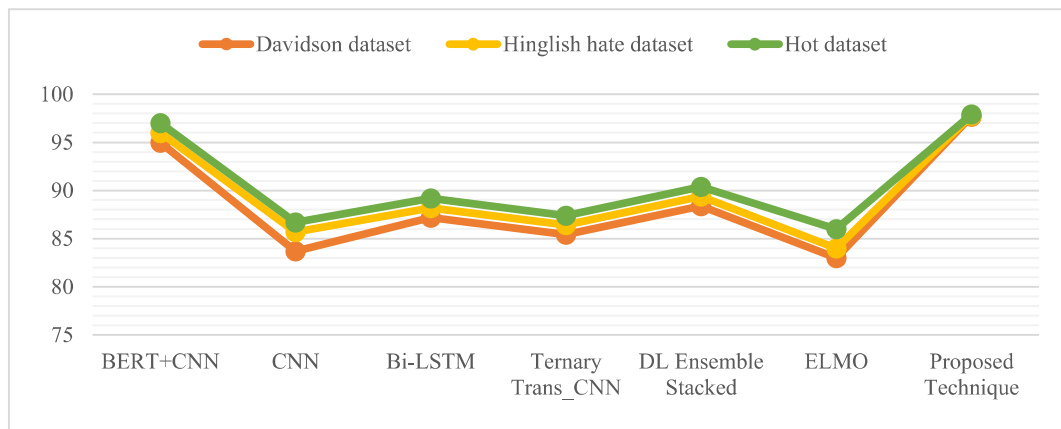


**Fig. 4.9. Comparison Analysis of Accuracy of HateDetetctor technique with previous Implemented techniques**



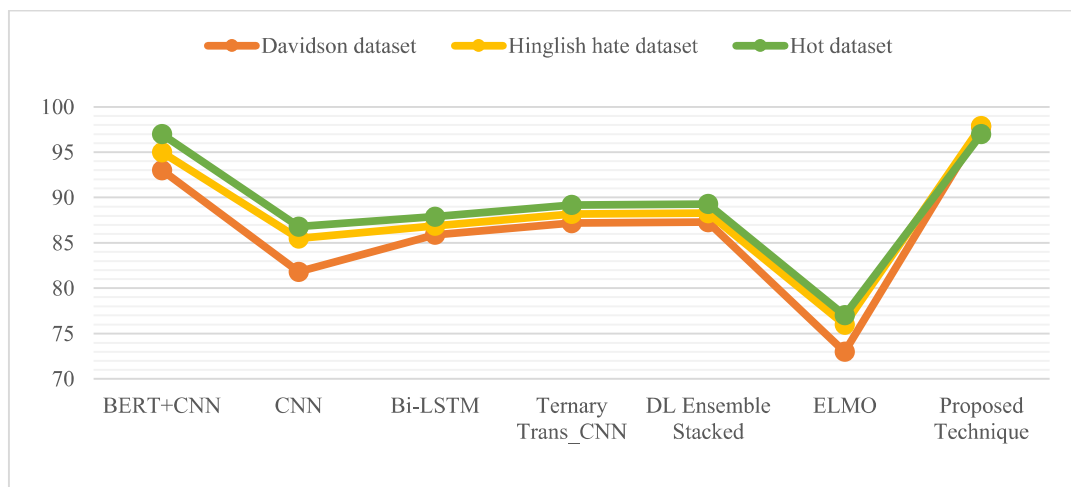
**Fig.4.10. Comparison Analysis of Recall of HateDetetctor technique with previous Implemented techniques**

In Fig. 4.10 the comparison analysis of proposed and existing technique was viewed based on the above said three data set. The three types of datasets maintain the same level of recall in the proposed when compared with the other existing technique. The proposed has achieved high recall without any such deviations and the CNN, and Bi-LSTM techniques have undergone more deviations when compared with others.



**Fig.4.11. Comparison Analysis of Precision of HateDetetctor technique with previous Implemented techniques**

In Fig. 4.11 the proposed and existing technique is compared with three datasets. The proposed technique has achieved high precision too and all other techniques have highly deviated for all three data sets.



**Fig 4.12. Comparison Analysis of F1-Score of HateDetetctor technique with previous Implemented techniques**

From Fig. 4.12 it has been clear that the proposed technique does not vary for any dataset given. The output result shows that all the three data set that works under this proposed brings out the output with a high-performance range. All technique has high deviations and Bi-LSTM, Ternary Trans-CNN, and DL Ensemble Stacked technique has equal deviations.

**Table 4.1. The comparative analysis of proposed and existing techniques**

<b>Hate Speech Models</b>	<b>Davidson dataset</b>	<b>Hinglish Hate dataset</b>	<b>Hot dataset</b>
<b>BERT+CNN</b> [165]	94	94.6	95.6
<b>CNN</b> [166]	84.7	85.7	86.7
<b>Bi-LSTM</b> [167]	86.2	87.2	88.2
<b>Ternary Trans_CNN</b> [168]	84.4	86.4	87.4
<b>DL Ensemble Stacked</b> [169]	87.4	85.4	86.4
<b>Distil-Bert</b> [170]	69	69.1	69.5
<b>SentBERT</b> [171]	80	80.3	80.2
<b>XLM-R</b> [172]	66	66.2	66
<b>ELMO</b> [173]	73	75	78
<b>Proposed Technique: HateDetector</b>	<b>97.5</b>	<b>97.3</b>	<b>97.7</b>

From Table 4.1, it shows the comparative analysis of proposed and existing techniques based on three different types of data sets: the Davidson Dataset, the Hinglish Hate Dataset, and the HOT dataset. The three types of datasets maintain the same level of accuracy in the proposed technique when compared with the other existing technique. Each technique taken for comparison has undergone certain deviations from its accuracy value for all three data sets. Here too, the accuracy does not change. The proposed technique has achieved high in its accuracy without any such deviations and the Ternary Trans-CNN and ELMO techniques have undergone more deviations when compared with others. Thus, comparing all the parameters of the proposed technique for all three datasets, it attained the utmost high performance. The proposed technique shows an accuracy level of 97.9%, precision 97.9%, recall 98% and F1-score 97.8%. On comparing with the existing technique, namely Bidirectional Encoder Representation from Transformer- Convolutional Neural Network (BERT-CNN), Convolutional Neural Network (CNN), Bidirectional Long-Short Term Memory (Bi-LSTM), Ternary Trans-CNN, Deep Learning (DL) Ensemble Stacked, Embeddings from Language Model (ELMO) the proposed technique gives higher and efficient performance.

#### **4.4. Conclusion**

In this chapter, an innovative technique for detecting and analyzing Multilingual Offensive Hate Speech (OHS) on Twitter using Artificial Intelligence was proposed. The approach involved a series of comprehensive steps, particularly in addressing Hinglish Tweets. The tweets were preprocessed using BERT, encoded using mBART, weight applied through N-gram and Bag of Words (BOW), spell-checked, and emotions analyzed using the Profanity Check technique. This technique employed ReLu, sigmoid function, and logistic regression for detecting hate speech. In

comparison to existing techniques such as BERT-CNN, CNN, Bi-LSTM, Ternary Trans-CNN, DL Ensemble Stacked, and ELMO, the proposed methods exhibited superior performance. The evaluation metrics, including accuracy, precision, F1-Score (97.9%), and recall (98%), demonstrated the effectiveness of the approach. Notably, the techniques excelled in eliminating code-mixing, offered fast processing capabilities, were easily trainable, and exhibited high reliability. In the next chapter, an exploration of unlabeled data for unsupervised machine learning models was undertaken, recognizing the significance of such data given the time-consuming nature of data labeling. Consequently, an in-depth investigation into deep learning models was deemed imperative and advantageous for effectively addressing hate speech issues.

## **CHAPTER 5**

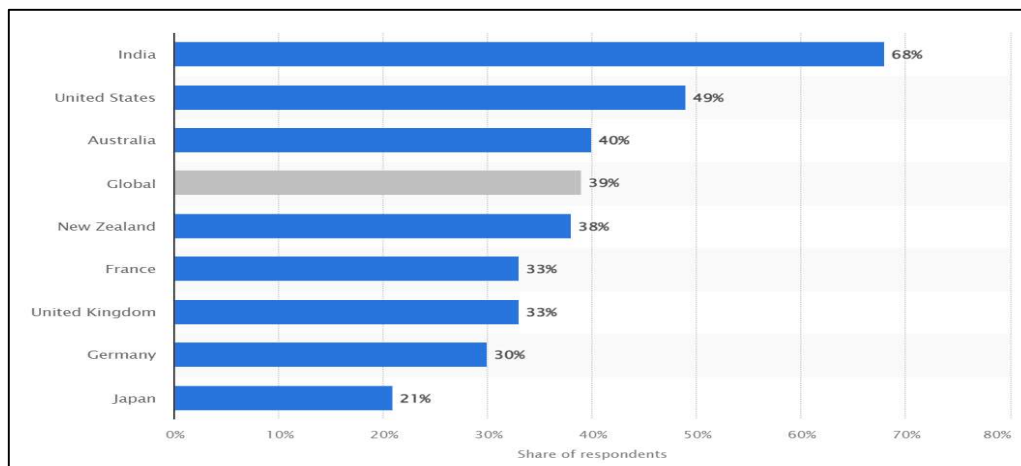
### **Creation of Generalized multilingual dataset for Hate Speech Analysis**

The issue of hate speech, particularly in the context of increasing online interactions on social media platforms, remains a significant challenge in contemporary society. Despite advancements in algorithms to address hate speech, research progress is hindered by the lack of suitable datasets in languages other than English. While various hate speech datasets in diverse languages exist, there is a notable absence of a standardized or universal approach for developing multilingual datasets. To bridge this gap, in this chapter we have introduced an innovative methodology that leverages RAKE and the Twitter web API to construct a standardized multilingual dataset. The dataset includes 3620 tweets in five different languages: Arabic, German, French, English, and Hindi-English Mix, with two labels indicating hatred and non-hatred. The dataset proves to be valuable in addressing challenges such as out-of-vocabulary words.

#### **5.1 Introduction**

The widespread adoption of social computing, particularly through platforms like social media and chat forums, has significantly enhanced human connectivity. Microblogging applications, in particular, empower individuals worldwide to quickly and extensively share their thoughts. These platforms offer convenient access and a level of anonymity, encouraging user engagement in discussions, debates, and the expression of opinions. However, some individuals misuse these platforms to dominate conversations, advocate for extreme perspectives, and occasionally pursue personal agendas. This interplay of factors has created an environment where aggressive and harmful content, commonly known as hate speech, can rapidly spread [174], [175]. Hate speech encompasses expressions that target individuals or groups based on attributes such as gender, race, ethnicity, skin color, nationality, political affiliations, sexual orientation, or regional characteristics. A significant challenge arises from the dissemination of hate speech through social media networks (SMNs), which function as one of the fastest communication platforms. In Fig. 5.1., instances of cyberbullying, over the different countries in year 2022. As the prevalence of cyberbullying cases on prominent online platforms such as Twitter and Facebook continue to rise, there is an increasing imperative for these platforms to assume accountability in detecting and isolating toxic content within their networks.





**Fig 5.1. Cyberbullying Instances over the different countries in year 2022<sup>23</sup>**

Given the vast volume of content on the internet, developing an automated system for detecting hate speech must prioritize scalability, reliability, and robustness. Traditional lexical detection methods often lack precision because they do not consider the context in which messages are conveyed during classification. To tackle the challenge of identifying Online Hate Speech (OHS), various machine learning models have been explored in the literature. These encompass algorithms such as support vector machines (SVM), random forests, neural networks, and other deep learning architectures [176], [177]. To enhance hate speech detection efforts, several curated datasets specifically tailored for this purpose have been created to support the research community. These datasets play a crucial role in training and evaluating classification models. For instance, Devansh et al. recently developed a dataset for the English language. In comparison, Sreelakshmi et al. focused on curating tweets in the Hindi language [178]. In previous studies, various approaches have been employed to extract and formalize data by employing varied methodologies for diverse languages [179]. In the recent work, Authors build a curated and annotated a dataset for target-based hate speech in the Hindi language, denoted as TABHATE. It consists of 2,020 tweets; the dataset undergoes annotation by three independent annotators. Employing a multiclass labeling approach, each tweet is categorized as either individual targeting, community targeting, or none. The dataset has 182 individual target text, 243 community target text and 1595 text has none category . The dataset was highly imbalanced and twitter API was used to extract the dataset. In another research, a dataset for Hindi-English was built. The tweets were extracted from the twitter blogs, were cleaned and labeled into two classes, 0 for Non hate and 1 Hate speech. Dataset comprised of 4579 tweets, out of which 1662 tweets were labelled as Hate speech

<sup>23</sup> <https://www.statista.com/statistics/272014/global-social-networks-cyberbullying-ranked-by-country/>

whereas 2919 were labelled as non-hate speech [180]. In another work, a dataset for hate speech detection using social media was curated which comprised of English sentences and was categorized into two classes: hateful content and non-hateful content. In total, the dataset includes 451,709 sentences, with 371,452 classified as hate speech and 80,250 as non-hate speech. The total number of bad words used in hateful content was 377. The dataset was built from different web sources like Kaggle, GitHub, and other websites [178]. A dataset named DIALOCONAN, compiled of 3,000 multi-turn fictitious dialogues crafted through human-machine collaboration was built, consisting of 3,059 dialogues. The dialogues encompassed six primary targets of hate, specifically JEWS, LGBT, MIGRANTS, MUSLIMS, PEOPLE OF COLOR (POC), and WOMEN [181]. The dataset was meticulously curated by combing human expert intervention with machine-generated dialogues derived from 19 distinct strategies. Knowledge-grounded Hate Countering dataset, comprising 195 pairs of hate speech and corresponding counter-narratives, along with the back-ground knowledge utilized in constructing the counter-narrative was built. Encompassing multiple hate targets such as islamophobia, misogyny, antisemitism, racism, and homophobia, this dataset offers a comprehensive collection of Hate Speech-Counter Narrative (HS-CN) pairs [182]. The counter-narratives were authored by an expert assigned to craft an appropriate response to a given hate speech, leveraging the relevant knowledge extensively. DeTox dataset encompassed 10,278 annotated German social media comments and offers a broader scope with twelve distinct annotation categories, extending beyond conventional hate speech detection. The labels focus on toxicity, criminal relevance, and various forms of discriminatory comments. The dataset employs six annotators to categorize and assign labels to different comment types. In the past numerous datasets on hate speech in various languages have been curated through diverse methods [183].

In the past, various datasets on hate speech in different languages have been assembled using diverse methodologies. However, the absence of a standardized or universally applicable approach for developing multilingual datasets has been evident. To fill this void, the authors introduce an innovative methodology for constructing a standard multilingual dataset. The proposed approach utilizes RAKE and the Twitter web API, with the detailed methodology elaborated in the following section.

To fill the Gap, in this chapter, the authors proposed an automatic method for keyword identification, leveraging this approach to extract data from diverse web platforms. The authors employed Rapid Automatic Keyword Extraction (RAKE)-based key-word extraction to identify high-scoring keywords. These selected keywords were then employed to search various web platforms for dataset creation. The proposed methodology is implemented across four distinct languages and is adaptable to others. To validate the results, two annotators categorized the text into labeled data [184],

[185]. Thus, in this work, the authors devised a versatile multilingual dataset for the analysis of hate speech, utilizing web platforms as the basis.

## **5.2. Proposed Approach**

In this chapter, a methodology is proposed based on RAKE and twitter API for building multi-lingual dataset. various components of the proposed model namely, RAKE Model and Twitter API.

### **5.2.1. Text Preprocessing**

To assign a newly reported OHS issue, the textual description undergoes several pre-processing steps. These steps include, Token Identification, removal of stop words followed by stemming. Since OHS text may consist of multiple sentences, the first step in token identification involves separating each sentence using a delimiter to extract them individually. Subsequently, the text is divided into smaller units, which can be words, symbols, or phrases. Then, commonly used words, such as articles, prepositions, and conjunctions which do not convey significant semantic information, are identified as stop words. To reduce noise in the data and emphasize more meaningful words, stop words are removed from the tokenized text. It is followed by reducing words to their root or base form by eliminating prefixes and suffixes. Stemming simplifies the vocabulary, facilitating the analysis and categorization of text data.

### **5.2.2 Feature Extraction**

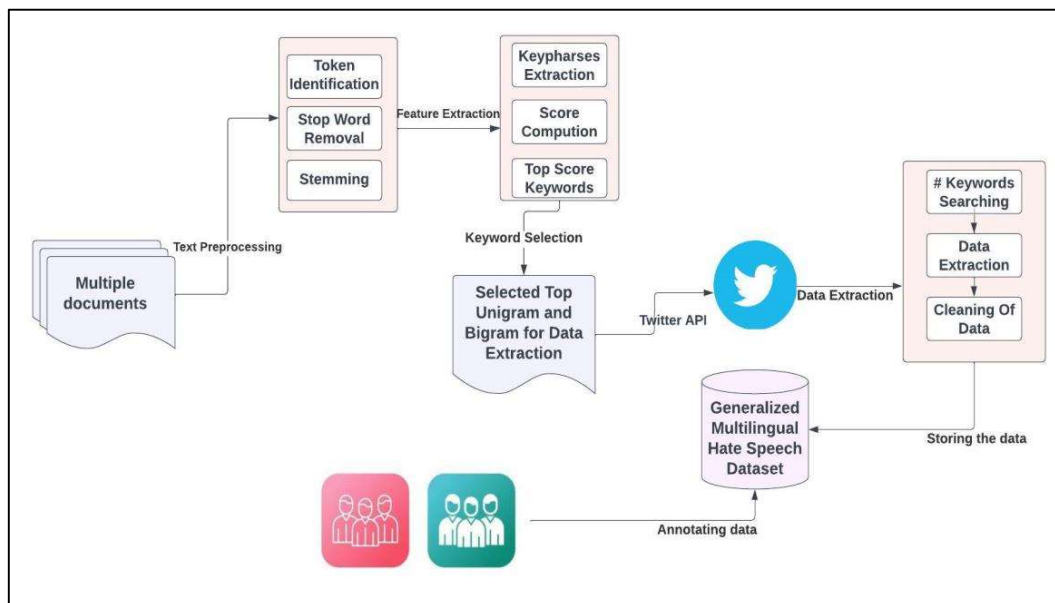
For the feature extraction authors has used the RAKE method to identify the most important unigram and bigram from the text. RAKE has been employed to identify the most relevant keywords within the text [186]. The first step involves Candidate Keyword Identification, where RAKE divides the text into sequences of one or more content words, creating candidate keywords. Subsequently, Co-occurrence Analysis is conducted to generate candidate keywords by evaluating the frequency of content words co-occurring within them. Following this, Keyword Scoring is applied, assigning a score to each candidate keyword based on the cumulative scores of its individual content words. This score is determined by considering both the frequency and significance of each content word. Finally, in the last stage of Final Selection, keywords with higher scores are selected as representative phrases or terms that effectively capture the essence of the content. From the set of identified key-phrases, authors have opted to choose unigrams and bigrams for keyword selection. This decision stems from the observation as, typically on the web, more than bigrams do not yield relevant search results.

### **5.2.3. Twitter API**

Twitter is a real-time microblogging site where users worldwide share their thoughts in concise tweets of up to 280 characters. It serves as a platform for networking, engagement, and interaction, enabling individuals, businesses, and organizations to

connect, share multimedia content, and participate in conversations. It also plays a crucial role in trend monitoring, offering insights into ongoing discussions and popular topics, while also serving as a significant platform for the dissemination of hate speech. Historically, 80% of hate speech text has been observed on Twitter.

The Twitter API (Application Programming Interface) facilitates developers in programmatically interacting with Twitter's platform. It allows for the retrieval and posting of data, accessing user information, and performing various other operations. The Twitter API offers different endpoints for various functionalities, with one of them being the Twitter Standard Search API, utilized for searching and retrieving tweets. It is employed to further refine the text collected in the initial step. The identified unigrams and bigrams are utilized as hashtags for searching tweets. Some of the hashtag words include "opfers" German word meaning "Victim" in English, and "missbraucht\_haben" which translates to "have\_abused". Subsequently, 500 texts are extracted for each language, employing different keywords. The tweets are cleaned by removing special characters, URLs, mentions, and emojis, lowercase conversion, and person\_name removal. Notably, stop words are retained, and stemming is omitted to preserve text integrity for annotation analysis. Proposed methodology is shown in Fig 5.2.



**Fig. 5.2. Proposed methodology**

### 5.3. Data Collection

Data has been gathered from various websites known for hosting substantial amounts of hate speech content. These sources include the dark web forum, Islamic text repositories, and hate speech dataset catalogs. The collected data spans five different

languages: Arabic, German, French, English, and Hindi. The collected dataset comprises a minimum of one thousand lines of text, characterized by substantial content related to online hate speech. After that these sentences are passed for the text processing step.

### **5.3.1. Annotating Dataset**

The tweets within the dataset underwent manual annotation by three separate and independent annotators. Three annotators annotated the dataset, comprised of a language expert, a psychologist, and a research scholar experienced in hate speech text. The dataset's appropriateness was assessed by calculating standard measures of inter-annotator agreement. Annotators were given ample time for the annotation process. Before commencing the task, they were clearly briefed on the annotation goal and how to categorize tweets based on the defined classes. Annotators were aware that the task involved reviewing content that could be hateful or offensive. Tweets deemed as hateful were those targeting individuals based on race, religion, gender, psychological beliefs, community/religion/organization affiliations, or political parties, spreading hate or false beliefs. Conversely, tweets not falling into the category of hate were labeled as non-hateful. Annotators assigned class labels after carefully evaluating each tweet. Since the dataset involves multiple languages, an initial generalized approach was adopted, using two labels: 0 for non-hate and 1 for hate. This simplified approach may evolve in the future to incorporate multiple categories for more nuanced analysis. In the last and final step, the annotated data with text file is stored in csv format for the further usage. The data on hate speech text has been collected for 5 different languages. To test the validation of the dataset authors will implement deep learning and machine learning models in future.

## **5.4. Implementation and Result**

To validate the proposed approach, multiple data sources were utilized by the authors. The data for multiple languages with a high content of hate speech was obtained from the dark web forum. Similarly, data on Hate Speech in different languages was available in the Hate Speech Dataset Catalogue. Once the data sources had been identified, the text data for various languages was stored in different folders. The implementation of the code was carried out using the Python programming language. The Google Colab environment was employed for Python programming. For preprocessing the data, the nltk, pandas, and numpy libraries were utilized for reading and text cleaning purposes. The Rake and multi-rake library were used for implementing the Rake module for keyphrase identification. The tweepy library was employed for the extraction and cleaning of Twitter tweets. To validate the research, the German language text data is presented by the authors. The text data, featuring a high content of hate speech, was collected from different websites discussed earlier. A sample is provided in Fig.5.3. Fig. 5.4. Depicts the English translation for the same.

"Es sind so abscheuliche, widerwärtige Vorwürfe, dass selbst hartgesottene Ermittlern der Atem stockte: Ein Jugendbetreuer aus Rinteln soll kleine Jungen auf brutalste Art vergewaltigt haben! Seit November sitzt Sascha B. in Untersuchungshaft in der JVA Hannover an der Schulenburger Landstraße, schweigt zu den Vorwürfen. Die Ermittler sind sicher, dass der 29-Jährige seine Opfer mit einer ganz perfiden Masche krderte: Er war Jugendbetreuer bei der Feuerwehr, erlangte so das Vertrauen der Kinder, die ihn arglos auch zu Hause besuchten. Er hatte ihnen, so erzählte ein Opfer der Polizei, seinen „niedlichen Hund“ zeigen wollen. Dann soll der Feuerwehrmann über seine Opfer hergefallen sein, sie missbraucht haben. Die Mutter eines Opfers wirft ihm in der Deister- und Weserzeitung vor: „Er hat meinen Sohn gepackt, aufs Bett geworfen, brutal vergewaltigt.“ Als der Junge vor Schmerzen aufschrie, soll Sascha B. ihm ein dreckiges Handtuch in den Mund gestopft haben".

**Fig. 5.3. German Language Hate Speech Text**

These are such disgusting, disgusting allegations that even hardened investigators took their breath away: A youth worker (29) from Rinteln is said to have raped small boys in the most brutal way! Since November, Sascha B. has been in custody in the Hanover prison onschulenburger Landstrasse and has remained silent about the allegations. The investigators are certain that the 29-year-old lured his victims with a very perfidious scam: He was a youth worker for the fire department and thus gained the trust of the children, who also unsuspectingly visited him at home. One victim told the police that he wanted to show them his "cute dog."

**Fig. 5.4. English translation for the above Text**

In next step, pre-processing is done using nltk library. After pre-processing, these sentences are passed into the multi-rake library. The keywords are extracted as depicted by Eq.(5.1)-(5.3)

$$rake = Rake(maxwords_{lang} = 3) \quad (5.1)$$

$$rake = Rake() \quad (5.2)$$

$$keywords = rake.apply(text) \quad (5.3)$$

After that score computation is performed on these keywords as depicted in Table 5.1. Similarly, unigram and bi grams are also extracted to search relevant text as depicted in Table 5.2.

**Table 5.1. Keywords and its Score Value**











German Keywords and its Score	Translated in English
('selbst hartgesottene ermittelern', 9.0), ('seinen „niedlichen hund', 9.0), ('aufs bett geworfen', 9.0), ('fast alle stammen', 9.0), ('oberstaatsanwalt bodo becker', 9.0),	('even hardened investigators', 9.0), ('his "cute dog", 9.0), ('thrown on the bed', 9.0), ('almost all of them come from', 9.0), ('chief public prosecutor Bodo Becker', 9.0)

Table 5.2. Bigram and Unigram Score Value

Bigrams and Unigrams and its score	Translated in English
('atem stockte', 4.0), ('kleine jungen', 4.0), ('jva hannover', 4.0), ('schulenburger landstra <sup>ك</sup> e', 4.0), ('erlangte so', 4.0), ('ihn arglos', 4.0), ('hause besuchten', 4.0), ('jugendbetreuer', 1.5), ('sicher', 1.5), ('junge', 1.5), ('opfer', 1.25), ('untersuchungshaft', 1.0), ('schweig', 1.0), ('vorwürfen', 1.0), ('ermittler', 1.0), ('29-jährige', 1.0), ('feuerwehr', 1.0), ('vertrauen', 1.0),	('breath stopped', 4.0), ('little boys', 4.0), ('jva hannover', 4.0), ('schulenburger landstra <sup>ك</sup> e', 4.0), ('achieved so', 4.0), ('him unsuspectingly', 4.0), ('visited home', 4.0), ('so told', 4.0), ('want to show', 4.0), ('youth worker', 1.5), ('safe', 1.5), ('boy', 1.5), ('victim', 1.25), ('remand', 1.0), ('silent', 1.0), ('accusations', 1.0), ('investigator', 1.0), ('trust', 1.0), ('children', 1.0), ('them', 1.0), ('police', 1.0)

The top 10 keywords for searching on Twitter, including "opfer," "hause besuchten," "weserzeitung," and "wurde," were selected from these unigrams and bigrams. The authors conducted a search for 35 top keywords in each language. Table 5.3 displays the tweets extracted using these keywords.

Table 5.3. Extracted Tweets from Twitter

Keywords	German Text search on twitter	English Translation
<b>Opfer</b>	<p>1. Es wird endlich Zeit   auch alle Cov19 Impf Opfer in DE mit solchen Summen zu entschädigen, da ihnen schweres Leid zugefügt wurde. Die Frau  in Dänemark erhielt 1,47 Millionen € für ihre schweren Cov19 Impf Schade</p> <p>2.   Ich hatte im ersten Moment gedacht, dass sind Opfer, die ab 2015, von den Kriminellen Migranten begangen wurden, Vergewaltigung,</p>	<p>1. It's finally time   to compensate all Cov19 vaccination victims in DE with such sums because they suffered severe suffering. The woman  in Denmark received €1.47 million for her severe Cov19 vaccination damage.</p> <p>2.   At first I thought that these were victims who were committed by the criminals of migrants from 2015 onwards,</p>

	Mörder, usw. gut das ich mich geirrt habe	rape, murderers, etc. Good thing I was wrong.
'hause besuchten'	Die Zeit der #Narren und #Jecken, der Höhepunkt der fünften Jahreszeit ist gekommen. Auch bei uns im Hause ist #Karnevalsstimmung ausgebrochen! Unsere Mitarbeiter haben sich verkleidet, besuchten Kindergärten der Umgebung und bescherten diese mit Krapfen! #Karneval #karneval2020	The time of #fools and #jerks, the climax of the fifth season has come. #Carnival atmosphere has broken out in our house too! Our employees dressed up, visited kindergartens in the area and gave them donuts! #Carnival #Carnival2020
'नफरत'	Nafrat tu kar mat, hum ekta me maante...	Don't hate, we believe in unity
'गिरफ्तार'	गाजियाबाद पुलिस ने दोनों "शांतिप्रिय नौजवानों" यश त्यागी और तुषार चौधरी को गिरफ्तार कर लिया है। यश त्यागी पेशेवर क्रिमिनल है। एक साल पहले बैंक लूट में जेल गया था।	Ghaziabad Police has arrested both "peaceful youths" Yash Tyagi and Tushar Chaudhary. Yash Tyagi is a professional criminal. A year ago he was jailed for bank robbery





Likewise, tweets were sought using various keywords and across diverse languages. The tweets presented in Table 5.4. are based on the extracted keywords for different languages.

**Table 5.4 Tweets extracted for different Languages**

Language	Number of Tweets
German	570
Arabic	700
French	650
English	1000
Hindi-English Mix code	700
<b>Total</b>	<b>3620</b>



Table 5.5 Sample Tweets and Its Label

Tweets	Annotator_1 Labelled	Annotator_2 Labeled	Annotator_3 Label	Final Label
“Es wird endlich Zeit   auch alle Cov19 Impf Opfer in DE mit solchen Summen...”	0	0	0	0
“   Ich hatte im ersten Moment gedacht, dass sind Opfer,die ab 2015,von den Kriminellen..”	1	1	1	1
“Die Zeit der #Narren und #Jecken, der Höhepunkt der fünften Jahreszeit ist gekommen. Auch bei uns im Hause ist #Karnevalsstimmung”	0	0	1	0
“Nafrat tu kar mat, hum ekta me maante..”.	0	0	0	0
“गाजियाबाद पुलिस ने दोनों "शांतिप्रिय नौजवानों" यश त्यागी और तुषार चौधरी को गिरफ्तार कर लिया है। यश त्यागी पेशेवर क्रिमिनल है..”	1	1	0	0

Following that, annotation was conducted for distinct labels, with 0 denoting non-hated content and 1 representing hated content, as illustrated in Table 5.5 and Table 5.6. In the Table 5.2 shows the sample example of tweets, here the annotators labeled them as; Final Label is considered as label for the tweets. Final labels are assigned as two of three annotator is agreed with the same label. So according to the final labeled for different tweets are shown in Table 5.6.

Table 5.6. Annotation for different text data

Language	Total Tweets	Non-Hated Tweets	Hated Tweets
German	570	134	436
Arabic	700	103	597
French	650	178	472
English	1000	245	755

<b>Hindi-English Mix code</b>	700	154	546
<b>Total</b>	<b>3620</b>	<b>814</b>	<b>2806</b>

### 5.5. Conclusion

The addressing of hate speech is recognized as a multifaceted challenge requiring a multidisciplinary approach for effective resolution. Collaboration with experts in linguistics, sociology, psychology, and the technological domain is deemed essential to garner a well-rounded perspective on both the problem and potential solutions. Staying attuned to the rapidly evolving online landscape, marked by the emergence of new platforms and trends, is also acknowledged as crucial. Despite the compilation of various hate speech datasets in diverse languages in the past using different methodologies, a standardized or universal approach has been lacking. To fill this gap, an innovative methodology for constructing a standardized multilingual dataset is pro-posed by the authors, leveraging RAKE and the Twitter web API. The dataset comprises 3,620 tweets across five languages: Arabic, German, French, English, and Hindi-English Mix, with two labels indicating hatred and non-hatred. Adopting a comprehensive research approach, a diverse dataset of Twitter posts in multiple languages was built, aiming to encompass a broad spectrum of content. Utilizing advanced Natural Language Processing (NLP) techniques, the content is meticulously analysed and categorized, with a focus on identifying instances of hate speech, offensive language, and potential targets. This inclusive framework, spanning multiple languages, is deemed essential for the development of a holistic understanding of the issue, considering the substantial variations in hate speech across languages and cultural contexts.

## **CHAPTER 6**

### **A Versatile Framework for Hate Speech Detection for Multilingual Datasets based on BERTopic**

#### **6.1 Introduction**

This chapter introduces a second approach for Online hate speech detection for Multilingual datasets. The proposed approach is innovative and results in a more accurate approach compared to our initial approach outlined in Chapter 5 of this thesis. This novel approach aimed at addressing the limitations of existing research in combating Online Hate Speech (OHS). Despite significant efforts documented in the literature, several key shortcomings persist, impeding the effectiveness of hate speech classification. Firstly, the exploration of methods to simplify complex language expressions and improve the detection process, especially in the context of diverse and evolving datasets, has been insufficiently addressed. Additionally, the utilization of highly precise techniques such as Discriminative Linguistic Feature Analysis remains relatively scarce, warranting further investigation to enhance their effectiveness.

Furthermore, there is a absence of highly accurate techniques for cross-platform hate speech detection, indicating a critical need for the development of improved methods in this domain. The predominance of research focused on detecting hate speech in English has left numerous languages, including Arabic, Indonesian, Italian, Turkish, Swedish, and Albanian, with limited investigations. Consequently, hate content in languages such as Hindi and Hinglish remains inadequately filtered, highlighting a significant gap in current research efforts.

To overcome these challenges, the proposed approach leverages Bidirectional Encoder Representations from Transformers (BERT) and topic modeling. BERT encoding is chosen for its ability to capture contextual information and semantic relationships within text, making it well-suited for understanding and processing hate speech text. Complementarily, topic modeling is applied to extract additional insights and uncover underlying topics or themes within the hate speech text.

This hybrid approach enables the discovery of hidden topics, facilitates text categorization, and supports topic-driven analysis, ultimately enhancing hate speech detection and management. By combining advanced techniques in natural language processing and machine learning, the proposed approach offers a promising avenue for addressing the complex challenges associated with combatting online hate speech across diverse languages and platforms.

The major contributions of the proposed work are as:

- BERT embeddings are integrated with clustering techniques to enhance topic extraction from text data and addresses the high-dimensional embedding issues.
- Uniform Manifold Approximation and Projection (UMAP) and Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) are used which reduces the computational load of BERT embeddings and improves its effectiveness in uncovering intricate topic structures.
- Further, to improve topic interpretation and simplifies large documents, exemplar documents are selected based on reduced embeddings.
- The hybrid approach of BERTopic and SVM for hate speech classification creatively merges topic modeling and classification.

## 6.2. Algorithms used for Online Hate speech detection

In this section, various algorithms used to detect hate speech are explained.

### 6.2.1. Text Pre-processing

To assign a newly reported OHS, textual description is first pre-processed. It includes various steps such as:

**Segmentation:** OHS text may contain multiple sentences. Segmentation involves separating each sentence by a delimiter (such as a period or newline character) to extract them separately. This step helps in analyzing and processing each sentence individually.

**Tokenization:** The divides text into smaller units. These tokens can be words, symbols, or phrases. For example, the sentence "I live in India" can be tokenized as: ["I", "live", "in", "India"].

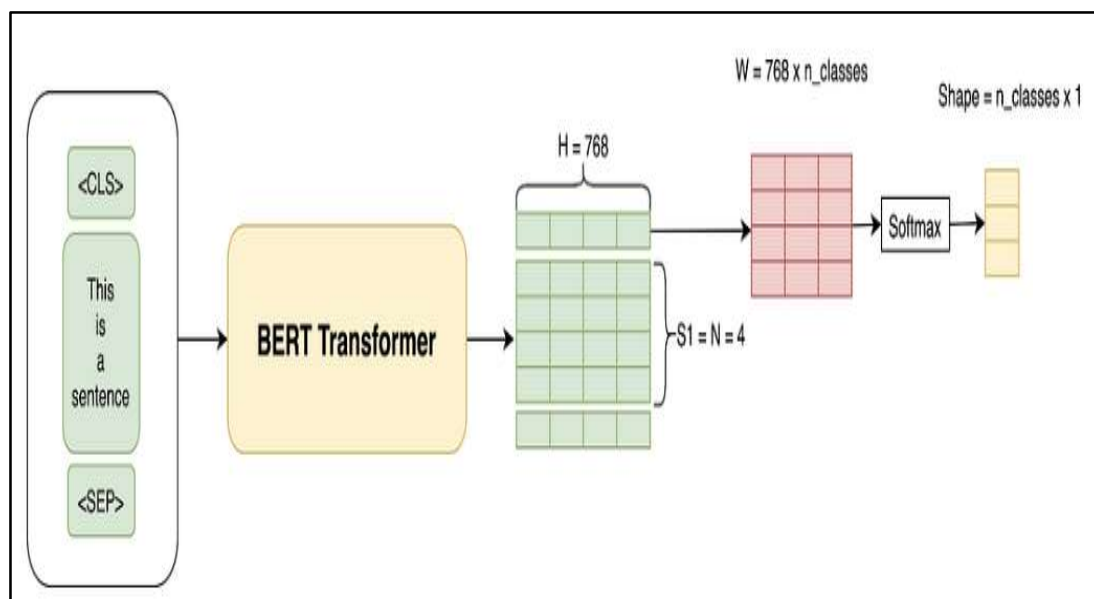
**Removal of Stopwords:** They are commonly used words that do not carry much semantic information, such as articles, prepositions, and conjunctions ("the," "a," "and," "this," etc.). It reduces noise in the data and focus on more meaningful words. This step is performed by removing the identified stopwords from the tokenized text.

**Removal of punctuation:** Punctuation marks and special characters like periods, commas, question marks, and exclamation marks are often irrelevant. Removing them simplifies the text and reduces noise.

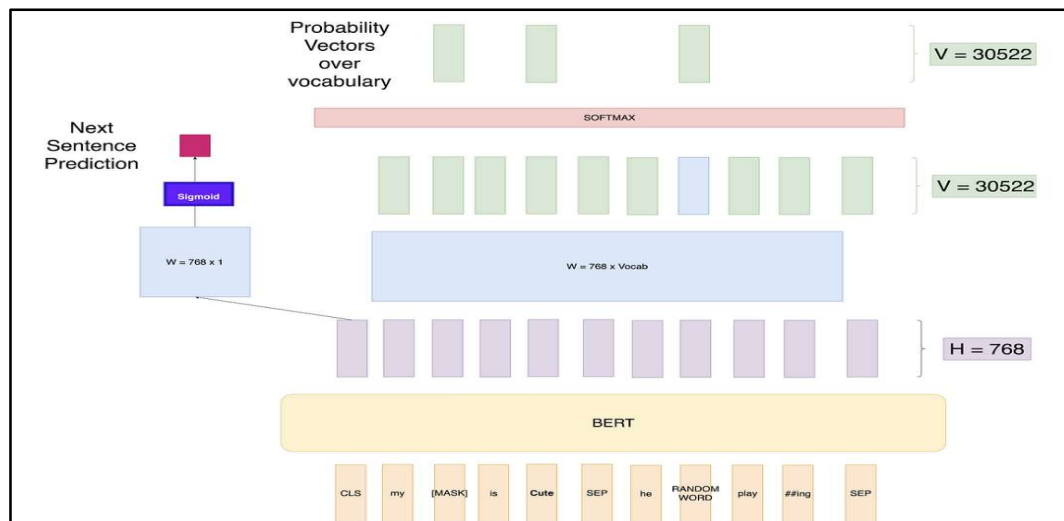
**Stemming:** It reduces words to their root or base form by removing prefixes and suffixes. This process helps simplify the vocabulary, making it easier to analyze and categorize text data. For instance, the word "presentation" can be stemmed to "present." By consolidating words with the same root, stemming reduces the dimensionality of the data [187].

### 6.2.2. Bidirectional Encoder Representations from Transformers (BERT)

BERT achieves human-like text comprehension and generation by grasping contextual word relationships within sentences. It employs a transformer architecture, leveraging self-attention mechanisms to consider a word's entire context, both left and right neighboring words. This bidirectional approach grants BERT a profound understanding of word context and meaning. During training, BERT utilizes extensive datasets such as Wikipedia articles and books. To generate word embeddings using the BERT tokenizer, the initial step involves dividing the input text into its constituent words or segments. Following this tokenization process, the processed input is fed through the BERT model to generate a series of hidden states. These hidden states are instrumental in crafting word embeddings for every word within the input text as depicted in Fig. 6.1. The training process involves two core tasks: masked language modeling and next sentence prediction. In masked language modeling, BERT learns to predict masked words within sentences, encouraging a deep understanding of word relationships. Next sentence prediction involves discerning whether a sentence follows another in a given text. After pre-training on this vast corpus, BERT can be fine-tuned for specific tasks. Fine-tuning entails adding task-specific layers atop the pre-trained BERT model and training it on labeled task-specific data as depicted in Fig. 6.2. Pre-training imparts general language understanding, while fine-tuning tailors BERT's capabilities for precise performance on distinct tasks [127], [132], [188].



**Fig.6.1. Working of BERT Transformer for embedding**



**Fig.6.2. Prediction with BERT**

### 6.2.3. Topic Modelling

Topic modelling algorithm that is often used to discover underlying topics in a collection of documents. Although BERT and topic model serve different purposes and operate at different levels of language understanding, it is possible to use topic model on the output generated by BERT for software bug reports. When BERT processes text(tweets), it captures the contextual relationships and representations of the tokens in the bug reports. The output of BERT can be a dense vector representation, which encodes the semantic meaning of the bug report. Topic modeling operates at a more abstract level, seeking to unveil hidden themes within a set of documents. It operates under the assumption that each document comprises a blend of various topics, and each topic can be defined by a distribution of words [189], [190]. By examining how words tend to appear together in the documents, topic modeling deduces the latent subjects or themes that underlie the collection.

## 6.3. Proposed Framework

In this section, the framework of the proposed approach as depicted in Fig. 6.3. and the core algorithm used to detect hate speech is explained.

### 6.3.1. Pre-trained BERT Encoder

First, the multilingual datasets are pre-processed using standard text pre-processing as explained in section 6.2.1. After pre-processing of textual data, the processed text data is converted into contextualized embeddings using the Pre-trained BERT encoder. Contextual embeddings are crafted for every token in the input sequence, capturing not only the individual word meanings but also their nuanced changes in various sentence contexts. BERT stands out as a cutting-edge language representation model explicitly created to grasp contextual information from both the left and right contexts

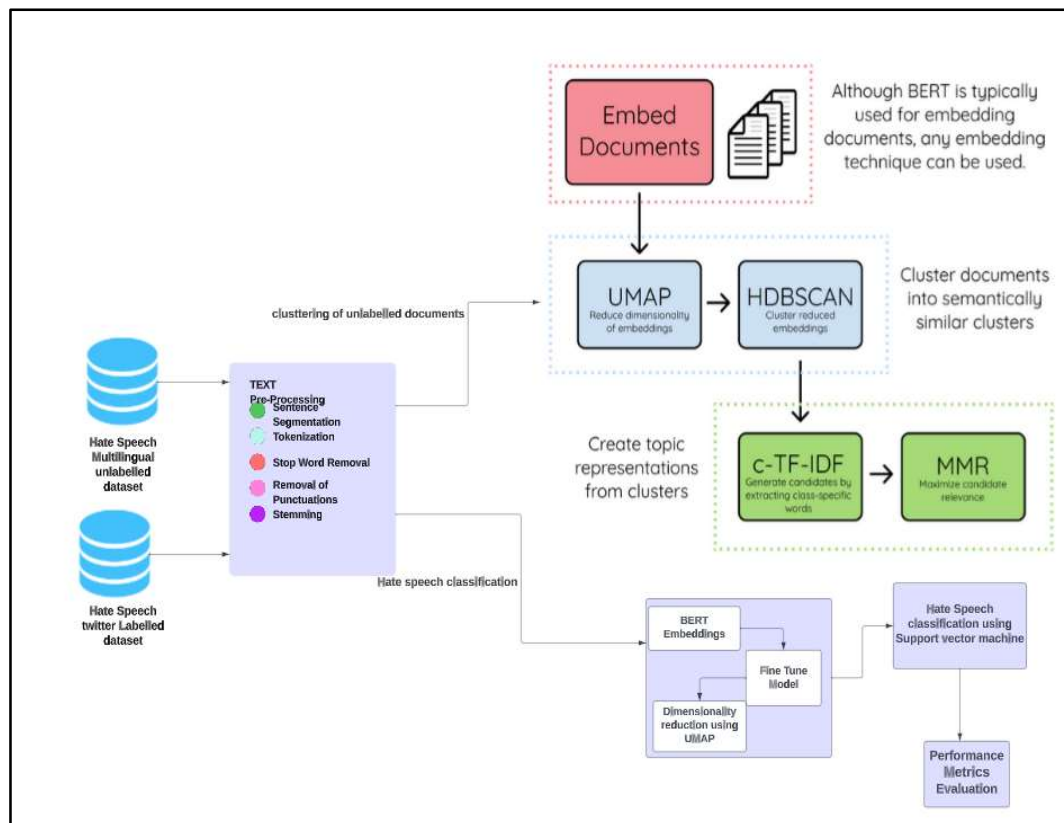
within a text. This capability is attained through extensive pre-training on a vast corpus of unlabeled text, empowering BERT to acquire profound bidirectional representations.

$$\text{BERT}_{\text{input}(x)} = \text{BPEE}(x) + \text{SE}(x) + \text{PE}(x) \quad (6.1)$$

As shown in Eq. 6.1 the BERT model takes three important components as input: byte pair encodings embedding (BPEE), segment embedding (SE), and positional embedding (PE). These components help BERT effectively process and understand the input text. Byte Pair Encodings Embedding (BPEE) represents the text at the subword level, allowing BERT to handle out-of-vocabulary words effectively. It breaks down words into subword units, which helps capture the meaning of complex and rare words. BPEE enables BERT to handle a wide range of vocabulary and improve the model's ability to understand and generate text. Segment Embedding (SE) is used to differentiate between different segments of text, such as sentences or paragraphs. In tasks that involve multiple text inputs, SE helps BERT distinguish between different parts of the text and understand the relationships and interactions between them. By incorporating segment embedding, BERT can process and comprehend the context and connections between various segments of the input text. Positional Embedding (PE) provides positional information to help BERT understand the sequential order of words within a sentence. By including positional embeddings, BERT can capture the relative positions of words and consider the sentence structure. This positional information enables BERT to understand the context and meaning of words within the context of the entire sentence. It's aims to enhance computers' understanding of ambiguous words in the text by leveraging contextual information from the surrounding text. By taking into account the context and relationships between words, BERT can better grasp the meaning and nuances of the text. In the proposed system, BERT is utilized as a powerful tool to extract contextualized information from the input text. By considering the surrounding context, BERT can capture rich semantic information and improve the model's ability to understand and process text data effectively. This enables the model to capture the subtleties and nuances of language, leading to improved performance in various NLP tasks, including hate speech detection.

### **6.3.2. Hate speech Clustering using BERTopic for Unlabelled Multilingual Datasets**

The embeddings obtained from BERT are used to cluster similar documents of unlabelled multilingual data of Hate speech. BERTopic is a technique in natural language processing that leverages the strength of BERT (Bidirectional Encoder Representations from Transformers) embeddings and clustering algorithms.



**Fig. 6.3. Framework for the proposed approach**

Its purpose is to identify and extract topics or clusters from a set of text documents. BERT, being a transformer-based model, can generate highly contextualized word embeddings. These embeddings capture rich semantic information from the text. Each word in a document is transformed into a vector representation based on its context within the sentence and document.

$$\text{Word\_embedding} = \text{BERT}(\text{word}) \quad (6.2)$$

To enhance the manageability and efficiency of BERT embeddings, dimensionality reduction techniques are employed. One such method is Uniform Manifold Approximation and Projection (UMAP), which reduces the dimensionality of embeddings while preserving essential data structures. UMAP transforms high-dimensional vectors into a lower-dimensional space, maintaining inherent patterns and relationships. The reduced-dimensional embeddings are then subjected to a clustering algorithm in BERTopic. Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) is chosen for this purpose. HDBSCAN excels at identifying clusters of various shapes and densities, making it adept at capturing intricate topic structures in text data. Through this clustering process, distinct clusters are formed, each representing a potential topic.



$$\text{Reachability} - \text{distance}(p, q) = \max(\text{core} - \text{distance}(p), \text{distance}(p, q)) \quad (6.3)$$

Where:

- $\text{core-distance}(p)$  is the core distance of point  $p$ .
- $\text{distance}(p, q)$  is the distance between points  $p$  and  $q$

Each document belongs to one of these clusters, indicating its primary topic. These clusters can correspond to a broad range of topics present in the document collection. To summarize each topic cluster, a representative document is selected. This representative document is chosen as the one closest to the centroid of the cluster in the reduced-dimensional embedding space. This document serves as an exemplar of the documents within that cluster, offering insights into the primary topic of that cluster. For further enhancement of the interpretability of topics, BERTopic identifies keywords associated with each topic using TF-IDF (Term Frequency-Inverse Document Frequency).

$$\text{TF} - \text{IDF}(\text{term}, \text{document}) = \text{TF}(\text{term}, \text{document}) * \text{IDF}(\text{term}) \quad (6.4)$$

This is done by analyzing the most important terms within the documents belonging to a cluster. Once the topics are identified, meaningful labels are assigned for interpretation. Fig 6.3 depicts the framework of the proposed approach.

### 6.3.3. Hate Speech classification using BERTopic and Support Vector machine for Labelled Datasets

After clustering Unlabelled data of Hate speech into appropriate clusters, further Classification is done for labelled data of hate speech using BERTopic in combination with Support Vector Machines (SVM). It is a two-step process that involves first extracting topics from the text data using BERTopic and then using SVM for binary classification (hate speech or not hate speech) based on the identified topics. After extraction of topics, feature engineering is performed first. Each document is associated with a feature vector that represents the likelihood or strength of its association with each topic cluster.

$$\text{Feature\_vector}(\text{document}) = [P(\text{Topic}_1), P(\text{Topic}_2), \dots, P(\text{Topic}_n)] \quad (6.5)$$

Where:

- $P(\text{Topic}_i)$  represents the probability or strength of association of the document with  $\text{Topic}_i$ .

These feature vectors serve as the input for the subsequent classification step, with each feature corresponding to a topic. In next step, data splitting is performed in which dataset is split into training and testing sets for SVM model training and evaluation.

$$\text{Training\_set}, \text{Testing\_set} = \text{Split}(\text{Data}, \text{Train\_percentage}) \quad (6.6)$$

Where:

- Data is the entire dataset.
- Train\_percentage is the percentage of data allocated for training.

After splitting, model is trained on a binary SVM classifier on the training data using the topic-based feature vectors as input and the hate speech labels as the target variable. The proposed approach is novel and significant as BERTopic often extracts complex and non-linear relationships between documents and topics. SVM, with the use of kernel functions can capture these intricate decision boundaries, making it well-suited for classifying text data with non-linear topic structures. Also, BERTopic create feature vectors for text documents based on their associations with topics. These feature vectors can capture the semantic meaning of the text more effectively than traditional bag-of-words representations. SVM can work well with high-dimensional feature spaces, making it a suitable choice for utilizing the topic-based features extracted by BERTopic. After model training, model is evaluated on the testing data using performance metrics such as accuracy, precision, recall, F1-score, and ROC AUC.

#### **6.4. Pseudocode of the proposed approach**

The pseudocode of the proposed approach is shown below:

#### **6.5. Implementation**

The proposed approach is evaluated of three distinct datasets described in section 6.5.1. Also, the various steps of implementations are presented.

##### **6.5.1 Dataset Used**

The proposed approach has been applied on three distinct datasets namely, Islamic dataset; Tumblr dataset and Twitter dataset.

- a. The Islamic dataset was collected from 2004-2010 comprising of 78,304 tweets. In this work, tweets from 2008-2010 is considered comprising of 5757 tweets. The Tumblr dataset comprises of 1,005 text descriptions. These two datasets, Islamic and Tumblr datasets are multilingual un-labeled data consisting of hate speech tweets.

**Input:**  $D_{Islamic}$ ,  $D_{Tumblr}$ , and  $D_{Twitter} \in D$  be the datasets

**ED** – Represent the contextual embeddings for each token in  $TD$ ,

**CD** – Set of clusters for dataset  $D$ , where each cluster  $c$ ,  
 $c \in CD$  represents a potential topic

**RD** – Set of representative documents for dataset  $D$

**KD**– Set of keywords associated with each cluster in  $CD$  for dataset  $D$

**FD** – Set of feature vectors for dataset  $D_{Twitter}$

**MD** – Represents the trained model

- For each dataset  $D$ , apply text preprocessing steps:

**// Text Preprocessing**

- Split  $D$  into sentences:  $S_D \leftarrow \text{split\_sentences}(D)$ .

- Tokenize each sentence  $s$  in  $S_D \leftarrow T_D - \text{tokenize\_sentences}(S_D)$ .

- $T_D \leftarrow T_D - \text{remove\_stopwords}(T_D)$ .

- $T_D \leftarrow T_D - \text{remove\_punctuation}(T_D)$ .

- $TD \leftarrow T_D - \text{stem\_words}(T_D)$ .

- $ED \leftarrow \text{Bert\_Encoder}(TD)$

- $E'D \leftarrow \text{UMAP\_reduce}(ED, n\_components)$

- $CD \leftarrow \text{HDBSCAN}(\text{Reduced Embeddings } E'D)$

- $KD \leftarrow \text{TF\_IDF}(\text{Clusters } CD)$

**//Topic Labelling**

Assign labels to the identified topics based on representative documents and Keywords

**// Hate Speech Classification with SVM**

- For  $D_{twitter}$ , do Feature Engineering:

- $FD_{train} \leftarrow \text{split } FD$

- $FD_{test} \leftarrow \text{split } FD$

- $MD \leftarrow \text{Train\_SVM}(FD_{train})$

- $\text{Accuracy} \leftarrow MD(FD_{test})$

- $\text{Precision} \leftarrow MD(FD_{test})$

- b. The Twitter dataset comprises of 24,784 tweets categorized in 3 different classes i.e. class 0,1 and 2, where class 0 represents hate speech tweets, class 1 represents offensive tweets and class 2 represents neither of them.
- c. The third dataset used is twitter dataset which consists tweets in Hindi language. It is a labelled dataset comprising of 4579 tweets. Out of 4579 tweets, 1662 tweets are labelled as Hate speech whereas 2919 are labelled as non-hate speech.

In this work, Twitter blogs in both English and Hindi are considered. The data is categorized into different classes, with English Tweets falling into three classes (0, 1, and 2), where class 0 represents hate speech tweets, class 1 represents offensive tweets, and class 2 represents neither category. For Hindi tweets, they are labelled in two classes: yes or no. Furthermore, the proposed framework is extended to unlabelled data, where offensive text is categorized into different categories, such as racism, colour bias, murder & hatred, and women related speech. In this case, two datasets have been utilized, one from Tumblr and the other from the Islamic community gathered from the dark web. Table 6.1 depicts the sample tweets and the text used in this work.

**Table 6.1. Sample tweets from distinct datasets**

S.No	Dataset Name	Text/Tweet	Class/ Category
1	Twitter Dataset[191]	<i>“ My grandma used to call me .....monkey all the time... she did refer to a broken bottle as a nigger_knife</i>	0 –Hate Speech
2	Twitter Dataset[192]	<i>You can call me Fireman booke cause I turn the hoes on.</i>	1- offensive
3	Twitter Dataset[193]	<i>Your teeth are like the stars.” “Aww thanks!” “Yeah... yellow...and white &amp; far near/ away from each other.”</i>	2- No offensive
4	Twitter Hindi Dataset[194]	<i>Doctor sab sahi me ke PhD (in hate politics) wale. Bhai padhe likhe ho fir kyu ye sab baate karte ho. Tum bas bowling khelo, aur maje lo.</i>	NO
5.	Twitter Hindi Dataset[195]	<i>Sarkar banne ke bad Hindu hit me ek bhi faisla Jo bjp ke dwara liya gaya ho, bjp ko gay, gobar, mandir,masjid aur nafrat faila kar vot chahiye</i>	Yes

6	Islamic dataset [60] [49]	<i>May Allaah preserve him and all the other Muslimeen..Aameen-Many people were wondering where brother Living went, alHamdulillah good to know he is alright and running about busy-</i>	Religion
7	Tumblr Dataset[52]	<i>How To Lose Your Mind To ISIS And Then Fight To Get It _ Sinjar leading a platoon of ISIS jihadis.</i>	Hatred

### 6.5.2. Evaluation Metrics

The performance of the proposed approach has been evaluated on various performance metrics namely; Precision, Recall, Accuracy, and F1-score. The mathematical formulas of various evaluation metrics are presented in eq. (6.7)-(6.9).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6.7)$$

$$Precision = \frac{TP}{TP+FP} \quad (6.8)$$

$$Recall = \frac{TP}{TP+FN} \quad (6.9)$$

Where,  $TP$  – True Positives,  $FP$  – False Positives,  $TN$  – True Negatives,  $FN$  False Negatives

Further, to evaluate semi-supervised approach, coherence score is computed through, different methods such as pointwise mutual information (PMI), normalized pointwise mutual information (NPMI), or cosine similarity. These methods quantify the semantic similarity between word pairs within the topic. In this research NPMI value has been used to compute the coherence, inter-topic distance and similarity graph for various bugs. NPMI calculates the coherence of a topic by measuring the degree of association between the top words within the topic as in eq. (2.10). A higher NPMI score indicates better coherence and interpretability of the topics.

$$f(w_i, w_j) = \frac{\log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}}{-\log P(w_i, w_j)} \quad (6.10)$$

$P(w_i, w_j)$  is the probability of words  $w_i$  and  $w_j$  co-occurring in a document and  $P(w_i)$  is the marginal probability of word  $w_j$ .

### 6.5.3. Process Illustration

The process of implementation of both semi-supervised and supervised approaches on Unlabelled and labelled datasets respectively is explained in this section.

### **6.5.3.1. System Configuration**

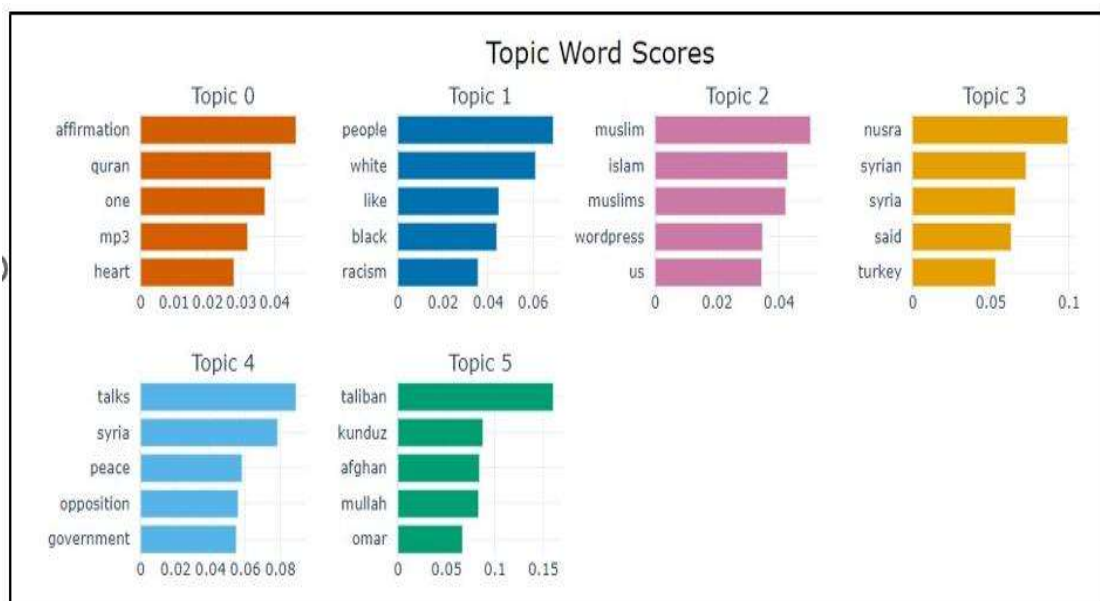
Python programming language and BERTopic package [196] was utilized for all experiments and model construction. All calculations were performed on a system with configurations Intel(R) Core i7 8750H CPU 2.20 GHz processor, 8GB RAM, and a 4GB NVIDIA GeForce 1050 TI Graphics Processing Unit running Windows 10 Home edition. Google Collaboratory Notebook was used for certain intensive computations and model training. Various libraries such as BERTopic, numpy, pandas, os, Nltk and UMAP, sklearn for embedding, classification and topic assignment are used.

### **6.5.3.2. Clustering of hate speech tweets for unlabelled datasets**

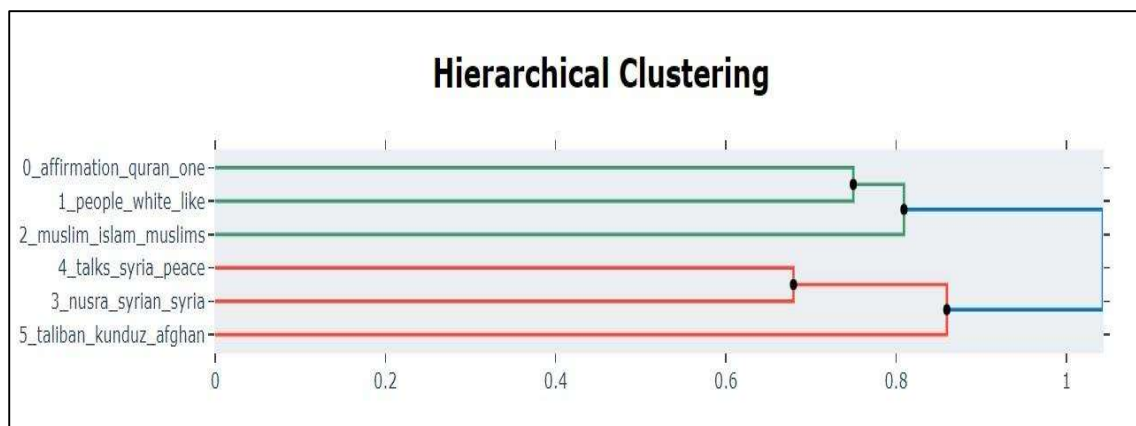
- For the Semi-Supervised learning approach, Islamic and Tumblr data were utilized. These datasets contained hate speech content, albeit without categorization into various labels.
- Tweets were loaded and stand pre-processing steps such as Tokenization, stop word removal, stemming and lemmatization were performed.
- After pre-processing of data, embeddings are computed using BERT.
- After computing sentence embeddings, dimensionality reduction is performed using Uniform Manifold Approximation and Projection (UMAP), which reduces the dimensionality of embeddings while preserving essential data structures. UMAP transforms high-dimensional vectors into a lower-dimensional space, maintaining inherent patterns and relationships.
- The reduced-dimensional embeddings are then subjected to a clustering algorithm in BERTopic. Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) is chosen for this purpose. HDBSCAN excels at identifying clusters of various shapes and densities, making it adept at capturing intricate topic structures in text data. Through this clustering process, distinct clusters are formed, each representing a potential topic.
- Various keywords and their probabilities for Islamic text can be observed in Fig 6.4. Furthermore, top keywords for each topic were identified and illustrated in Fig.6.5, followed by hierarchical clustering of various topics in Fig. 6.6.

<p>1: [('people', 0.06562247307029964), ('white', 0.05878585055746079), ('like', 0.04302949138223344), ('black', 0.04243572110125395), ('racism', 0.034423295288042145), ('it', 0.027357772854090245), ('racist', 0.02715523702223439), ('theyre', 0.02600544432990209), ('racial', 0.025249724600791897), ('say', 0.02406427455644119)],</p>	<p>2: [('muslim', 0.052936867496691374), ('attack', 0.029622579866950292), ('islam', 0.029439169027521795), ('muslims', 0.02885828433252691), ('islamic', 0.028515853425823027), ('us', 0.0248651321151447), ('ramadan', 0.02476166234813707), ('piece', 0.02476166234813707), ('view', 0.024400459810047227), ('wordpress', 0.023686396460793256)]</p>	<p>5: [('trump', 0.057542019307058344), ('kelly', 0.05403884142374713), ('sanders', 0.04601805959358869), ('bernie', 0.03971624186452528), ('class', 0.03706849240689026), ('ruling', 0.03541349615688516), ('political', 0.02569643952176791), ('clinton', 0.025227614039002783), ('khan', 0.025227614039002783),</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Fig.6.4. Keywords and its probabilities for Topic 0 and 1 for Ismalic Text**

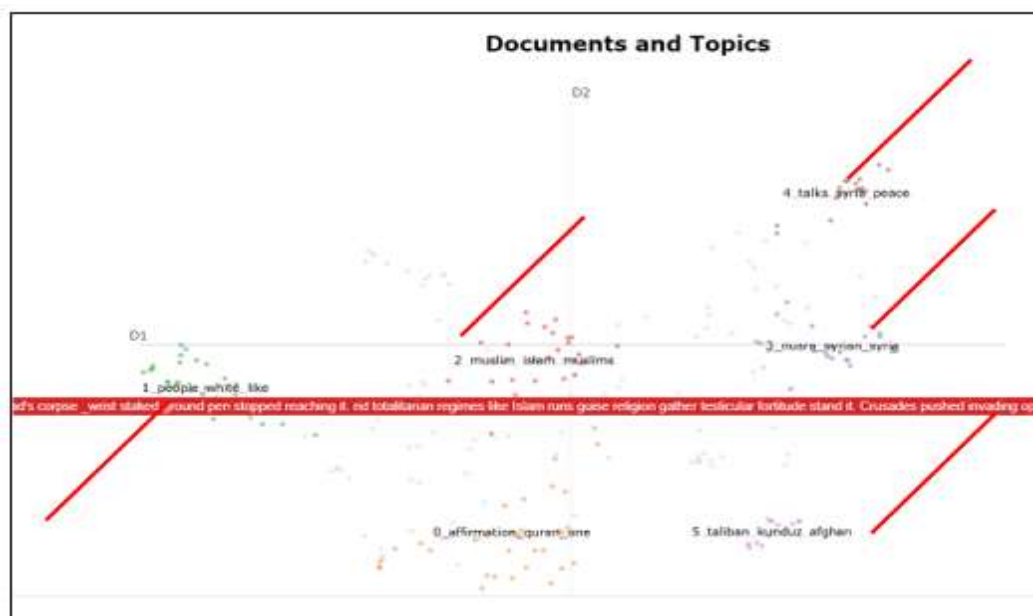


**Fig. 6.5. Top keywords for distinct topics for Islamic Text**



**Fig 6.6. Topics and Clusters for the Islamic Text**

Different clusters and their related documents were represented in Fig. 6.7. In this representation, clusters were denoted by lines, and documents were denoted by dots.



**Fig. 6.7. Documents are grouped in different Topics based on keywords**

### 6.5.3.3. Classification of hate speech tweets for Labelled Datasets

- For this approach, English and Hindi datasets were utilized. The English tweet dataset encompassed a total of 27,645 tweets, while the Hindi tweets dataset contained 4,759 tweets.
- Initially, the first data was loaded, and the "tweets" feature was utilized for the training and assignment process. Subsequently, the "tweets" were converted into a data frame, and basic text mining processes were implemented, including stop word removal, stemming, and lemmatizing.
- Following this, the sentence embedding method "paraphrase-MiniLM-L12-v2" and Principal Component Analysis method for dimension reduction were employed.
- Afterward, the UMAP library was utilized for clustering, and the final tf-idf method was employed for further processing. The values utilized for various hyperparameters were as follows: Min\_df was set at 10 for the smaller dataset and 20 for the larger dataset, while other parameters included number\_of\_components=5, minimum\_distance=0.0, and cosine distance for similarity computation.
- Subsequently, the model underwent training with various steps discussed above such as embedding, dimension reduction, clustering, and vectorization. Based on distinct features and keywords, all the documents were classified into



distinct topics. Finally, an SVM classifier was employed for the classification of tweets, utilizing the Sklearn library. SVM was chosen due to its effectiveness in classifying high-dimensional data.

- For illustration, “*I am Muhajir Aur mere lye sab se Pehly Pakistan he. agr 10 lakh Altaf Jese leaders bh is zameen ki behurmati kren un sbko sar e aam phansi Deni chahye.*” belongs to the cluster in topic 0. Furthermore, the model is trained on class labels as 1 or 0.
- Different keywords identified for Hindi Tweets for topic 1 and topic 3 were illustrated

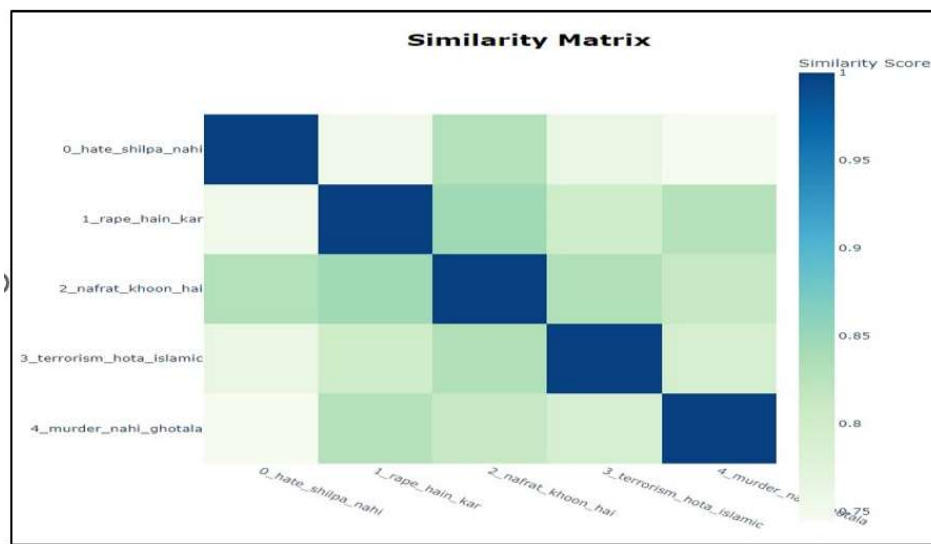
In Fig. 6.8., while Fig. 6.9. depicted the clustering of distinct documents into different topics followed by similarity matrix in Fig. 6.10.

<pre>( 'hate', 0.06959430223388244), ( 'nafrat', 0.03642317497013495) ( 'khoon', 0.03642317497013495), ( 'nahi', 0.03572949984353123), ( 'hai', 0.03227571007754056), ( 'aap', 0.027244447055535203), ( 'ne', 0.026264321003145564), ( 'wo', 0.025883210220456573), ( 'nhi', 0.02504826795528056), ( 'kar', 0.023787828573488008)]</pre>	<pre>( 'rape', 0.15555697589410222), ( 'murder', 0.0850239375151555) ( 'hain', 0.04829562489854222), ( 'kar', 0.043151923689099606), ( 'kya', 0.03177562778609578), ( 'nahi', 0.030092454930936416) ( 'party', 0.02889217174363587) ( 'hote', 0.02719192822780027), ( 'par', 0.02304384939181734), ( 'ne', 0.02242087742076537)]</pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Fig. 6.8. Keywords and its probabilities for Topic 0 and 1**

Document	Topic	Representation_ doc	Top Words	Probability
I am Muhajir .. Aur mere lye sab se Pehly Paki...	0	[@TheKaranPatel Yaar hadd ho gayi Karan Bua k...	hai - ko - khoon se - ki - nafrat - ke	0.97
KYa timepass kar rahi hai....sadak pe kisi se ...	2	[Humidity ka kam tu mat kar rape hua he bhi ki...	rape - he - ke - hai - ki - hain - ka - kar	1.000

**Fig. 6.9. classification of documents as per topics**



**Fig. 6.10. Similarity matrix for various topics**

- After the model had been trained with various steps as mentioned above, including embedding, dimension reduction, clustering, and vectorization, all the documents were classified into distinct topics based on their distinct features and keywords. Subsequently, an SVM classifier was utilized for the classification of tweets, utilizing the Sklearn library. SVM was chosen due to its effectiveness in classifying high-dimensional data. Finally, labels were assigned for each class for both Hindi and English Tweets, as depicted in Fig. 6.11.

Topic	Count	Name	Representation	Representative_Docs	pre_Class
0	202	0_bitch_bitches_hoes_gussy	[bitch, bitches, hoes, pussy, fuck, love, like...	["@Thornton25: &#8220;@KantKeepAhBYTCH: love e...	1
1	31	1_trash_blah_bird_ho	[trash, blah, bird, ho, hahaha, teach, teejone...	["@sleepy_yongguk: "I saw ajumma walk beginnin...	1
2	7	2_least_shoes_128166_glitter	[least, shoes, 128166, glitter, youll, hallowe...	["@RTNBA: Drakes new shoes released Nike/Jorda...	1

**Fig. 6.11. Classification of Tweets in different classes**

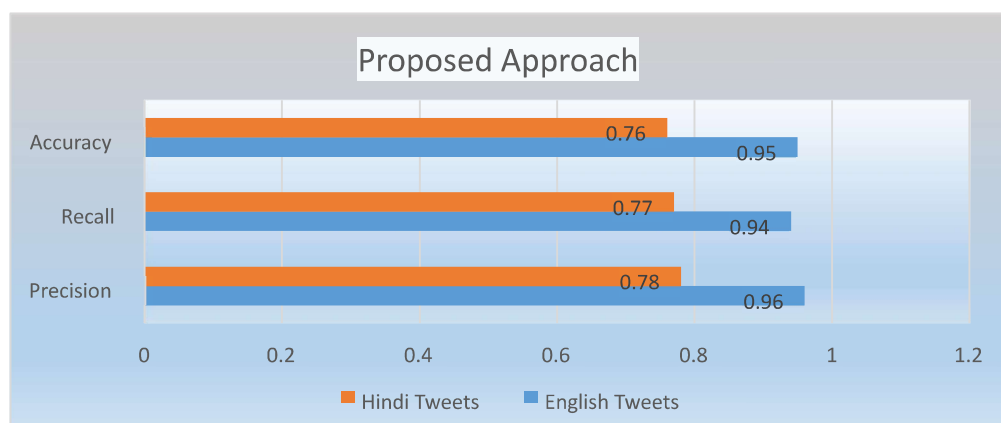
## 6.6. Results and Discussions

Addressing hate speech is a multifaceted challenge that necessitates a multidisciplinary approach for effective resolution. Collaborating with experts in linguistics, sociology, psychology, and technological domain is essential to gain a well-rounded perspective on both the problem and potential solutions. Also, it is crucial to stay attuned to the rapidly evolving online landscape, with the emergence of new platforms and trends. Therefore, the proposed framework is validated on multilingual data comprising of Islamic, English and Hindi languages. Two distinct

approaches were employed for hate speech detection. In the first approach, labelled data was utilized for classification. The documents were categorized based on these assigned labels. In the second approach, on unlabelled data, documents were grouped into specific topics. Two annotators were tasked with defining cluster names based on these topics. For instance, when addressing hate speech related to color discrimination, keywords such as “black,” “white,” “racism,” and “effect” were used to define clusters. The model is trained for approximate 45k tweets/text statements. The performance of the proposed approach has been evaluated on various performance metrics namely; Precision, Recall, Accuracy, and F1-score. The various performance metrics evaluated for the proposed approach are depicted in Table 6.2. and Fig 6.12.

**Table 6.2. Classification Accuracy by the Proposed Approach**

Proposed Approach	Precision	Recall	Accuracy
English Tweets	0.96	0.94	0.95
Hindi Tweets	0.78	0.77	0.76



**Fig. 6.12. Classification Accuracy of the proposed approach**

The proposed approach was evaluated on a mix of Hindi-English tweets text, comprising tweet ids and their corresponding annotations. In previous research, several features were utilized for hate speech classification [197]. The accuracy achieved with these previous features was compared with the current approach for Hindi tweets, as depicted in Table 6.3. Similarly, for English tweets in past research, comparisons were made regarding various parameters such as the social media platform, features, and accuracy with the current approach, as shown in Table 6.4. Further, the accuracy of proposed approach is compared with state-of-the-art techniques in Fig. 6.13.

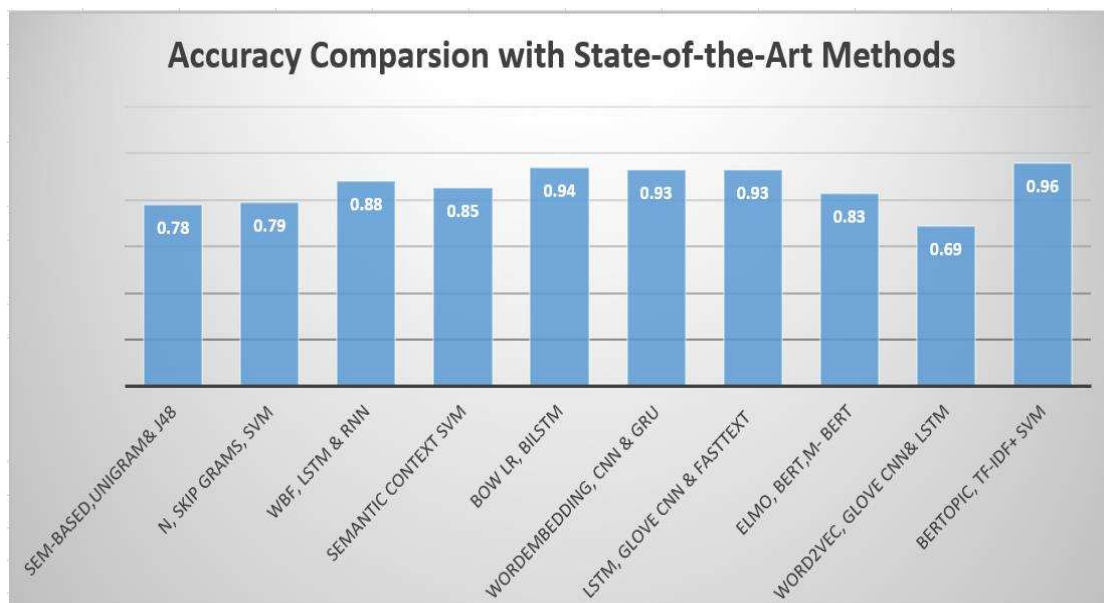
Table 6.3. Comparison with the other techniques for the Hindi tweets

Features	Accuracy
Character	66.8
Word-n-gram	69.9
Punctuation	63.2
Lexicon	63.8
All Features	66.7
All Features	71.7
Proposed Approach- keyword, topics and tf-idf	76.7

Table 6.4. Comparison with the other techniques for English Tweets

Author and Year	Platform	Type	Features Representation	Accuracy
Watanabe et al. 2018 [49]	Twitter	Hate, Offensive	Sentiment-Based, Semantic, Unigram-J48graft	0.78
Malmasi and Zampieri 2018 [198]	Twitter	Hate, Offensive	N-grams, Skip-grams, hierarchical, word clusters RBF kernel, SVM	0.79
Pitsilis et al. 2018 [91]	Twitter	Racism or Sexism	Word-based frequency and LSTM, vectorization RNN	0.88
Fernandez and Alani 2018 [199]	Twitter	Radicalization	Semantic Context SVM	0.85
Ousidhoum et al. 2019 [159]	Twitter	Sexual orientation, Religion, Disability	BOW LR, biLSTM	0.94
Zhang and Luo 2019 [85]	Twitter	Racism, Sexism	Word embeddings CNN + GRU	0.93
Badjatiya et al. 2017 [106]	Twitter	Hate Speech	Random embedding, LSTM, GBDT GloVe CNN, FastText,	0.93

<b>Dowlagar and Mamidi 2021, [200]</b>	Twitter	Hate Speech	ELMO BERT, Multilingual BERT	0.83
<b>Rizos et al. 2019 [201]</b>	Twitter	Hate Speech	Word2Vec, CNN, LSTM, GRU GloVe	0.69
<b>Proposed Approach</b>	<b>Twitter</b>	<b>Hate Speech</b>	<b>BERTopic, Tf-idf+ SVM</b>	<b>0.96</b>

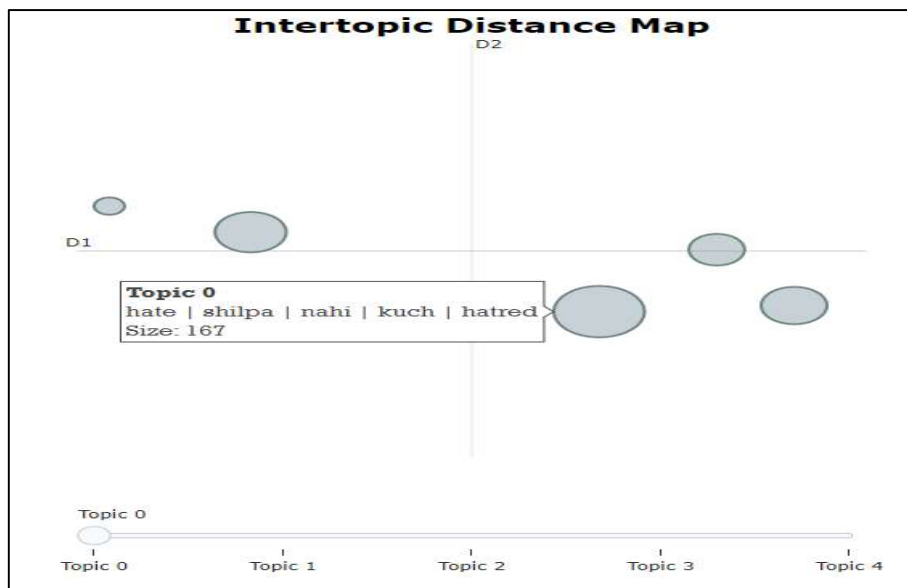


**Fig. 6.13. Comparisons with the state-of-the-art methods in terms of accuracy**

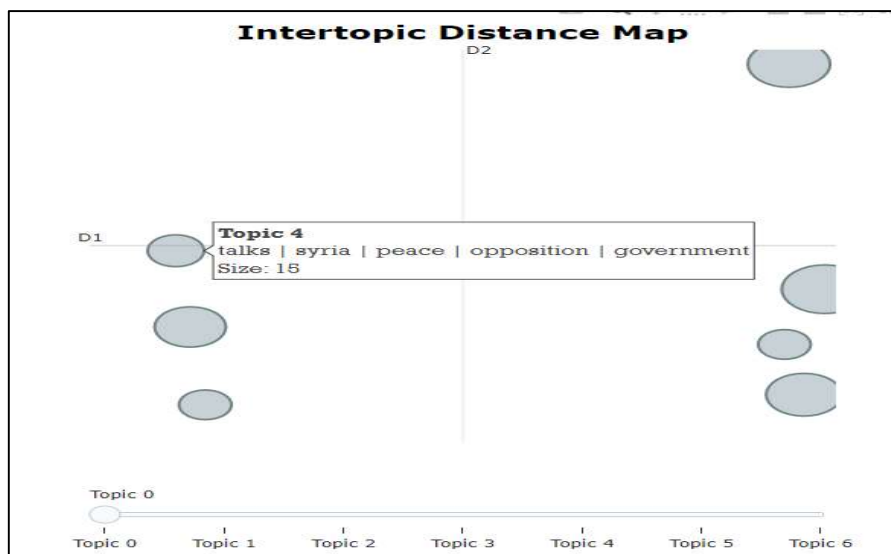
To evaluate the performance of the proposed approach on unlabeled datasets namely; Islamic Text and Tumbler Dataset, Coherence score is used. It is widely used in topic modelling to evaluate the quality and interpretability of the generated topics. The coherence score measures the semantic similarity between the top N words that represent a particular topic. NPMI for the different dataset are shown in Table 6.5 using UDAP method gives improved results as compared to the old vectorization method and Intertopic distance Map is shown in Fig 6.14 & Fig. 6.15 [159].

**Table 6.5. PMI for the proposed Approach**

Features Used	Datasets	NPMI using UDAP Method
<b>BERTopic+ Fine Tuning</b>	Tumbler	0.67
<b>BERTopic+ Fine Tuning</b>	Islamic	0.73
<b>BERTopic+ Fine Tuning</b>	Twitter English	0.78
<b>BERTopic+ Fine Tuning</b>	Twitter Hindi	0.73



**Fig. 6.14. Intertopic Distance Map for Hindi tweets**



**Fig. 6.15. Intertopic Distance**

## 6.7. Conclusion

The need for a fresh perspective on Hate Speech (HS) detection and addressing negative content arises from recognizing its significant societal impact, particularly in online environments. Detecting HS is a complex issue that demands not only advanced computational tools but also a deep understanding of its intricacies. The solutions provided should be comprehensive and beneficial for society as a whole. A proactive, prevention-oriented approach is crucial, necessitating collaboration between academia, social platforms, and public institutions. It is essential to raise awareness

about these challenges and provide a comprehensive overview of the progress made so far. To ensure a comprehensive research approach, a diverse dataset of Twitter posts in both English and Hindi is gathered, ensuring it represents a wide range of content. Advanced Natural Language Processing (NLP) techniques are utilized to thoroughly analyze and categorize the content, identifying instances of hate speech, offensive language, and potential targets. This comprehensive framework, covering multiple languages, is crucial for gaining a holistic understanding of the issue due to the significant variations in hate speech across languages and cultural contexts. Categorizing hate speech into distinct types further enhances our ability to analyze and comprehend its various manifestations, surpassing current state-of-the-art techniques. The proposed approach achieves a 96% accuracy for labeled text and 72% for unlabeled text. It is effective for multilingual text data, with plans to expand its application to languages such as Arabic and German in the future, which are often used in online hate communities. Additionally, we aim to work on community detection based on text mining.

The proposed approach has provided researchers with profound insights, enabling them to understand the implications related to language and culture. It supports targeted efforts to combat HS while addressing underlying biases and stereotypes. Creating a collaborative environment and fostering deeper understanding among stakeholders is pivotal. This approach effectively tackles HS, nurturing a more inclusive and respectful digital space for everyone.

## CHAPTER 7

# A Hybrid T5-LSTM Hate Speech Classification Framework for Multilingual Content

### 7.1. Introduction

This chapter introduces a third approach for Online hate speech detection for Multilingual datasets. The proposed approach is innovative and results in a more accurate approach compared to our initial approaches outlined in Chapter 5 and 6 of this thesis. This novel approach aimed at addressing the limitations of existing research in combating Online Hate Speech (OHS). Despite significant efforts documented in the literature, several key shortcomings persist, impeding the effectiveness of hate speech classification. One key limitation is the insufficient attention given to methods for simplifying complex language expressions and improving the detection process. Hate speech often manifests in various forms and evolves rapidly, making it challenging to detect using conventional techniques. Additionally, the reliance on individual highly precise techniques, such as Text to Text Transfer and Convolutional Neural Networks, without exploring hybrid approaches, further hinders effective hate speech classification. Hate speech detection requires nuanced understanding and context, which may not be adequately captured by singular techniques.

Moreover, the lack of highly accurate techniques for cross-platform hate speech detection exacerbates the problem. Hate speech proliferates across different online platforms and mediums, necessitating robust detection methods adaptable to diverse environments. Furthermore, the predominance of research focused solely on English neglects the prevalence of hate speech in other languages, leaving a significant portion of online content unchecked. This limitation underscores the importance of addressing multilingual challenges in hate speech detection to ensure comprehensive coverage across various linguistic communities. Therefore, in this work, to establish a robust methodology, a diverse dataset comprising Twitter posts in both English and Hindi and multilingual dataset (comprising tweets in five distinct languages) is analysed, ensuring inclusivity across a broad spectrum of content. The proposed framework named, Hybrid T5-LSTM Hate Speech Classification Framework (T5-LSTM HSCF) is designed to accommodate multiple languages, plays a pivotal role in facilitating this issue, thereby recognizing the substantial variations in hate speech across diverse linguistic and cultural contexts. Several researchers have utilized various methodologies for hate speech classification [113], [122], [135], [177], [202], [203], [204], [205], [206], [207]. Transfer Learning have also been used as a key strategy for hate speech classification [208], [209]. Transfer learning empowers models to generalize knowledge gleaned from diverse datasets and apply it to specific tasks, enhancing their adaptability and performance. Among these approaches, the Text-to-Text Transfer Transformer (T5 model) has emerged as a significant approach,



specifically for its text-to-text framework for training and encoding textual data. To address the various challenges encountered in hate speech classification, a hybrid methodology is proposed that integrates the capabilities of both the T5 and LSTM models. Initially, T5 algorithm is employed to train the data for hate speech detection, leveraging its text-to-text framework to encode and process textual information effectively. Subsequently, to refine the classification outcomes, LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Network) models were introduced into the framework. This hybrid approach facilitates a comprehensive analysis of hate speech by categorizing it into distinct types, thereby enhancing the ability to comprehend and address its diverse manifestations as compared to state-of-the-art techniques. The proposed hybrid approach attains improved accuracy rates of 96%, 76% and 71 % for labelled text in English, Hindi and multilingual dataset, respectively. By leveraging the strengths of both the T5 and LSTM models within a unified framework, this approach represents a significant advancement in hate speech classification, enabling more nuanced analysis and understanding of this complex and pervasive issue in online communication.

The major contribution of the proposed work is:

- Pre-trained models are applied on large datasets based on the principle of transfer learning and the proposed model is fine-tuned to improve classification accuracy.
- A hybrid methodology is proposed that integrates transfer learning and text to text transformer model in which the transformer architecture is applied for encoding and processing of textual data.
- Further to improve, Long Short-Term Memory and Convolutional Neural Network are incorporated that capture long-range dependencies in sequential data and retain information from past observations.
- The proposed hybrid approach achieves improved accuracy rates, with 96% accuracy for English and 76% accuracy for Hindi datasets as compared to state-of-the-art techniques.

## **7.2. Algorithms used for Hate Speech Detection**

In this section, various algorithms used for multilingual hate speech detection are explored.

### **7.2.1. Text Pre-processing**

To assign a newly reported OHS, multiple textual description is first pre-processed. It includes various steps such as:

**Token Identification-** OHS text may contain multiple sentences. In this, first separating each sentence by a delimiter (such as a period or newline character) to

extract them separately. After that text is into smaller units. These tokens can be words, symbols, or phrases.

**Removal of Stopwords:** They are commonly used words that do not carry much semantic information, such as articles, prepositions, and conjunctions ("the," "a," "and," "this," etc.). It reduces noise in the data. This step is performed by removing the Stemming identified stop words from the tokenized text.

**Stemming:** It reduces words to their root or base form by removing prefixes and suffixes. This process helps simplify the vocabulary, making it easier to analyse and categorize text data[210].

### 7.2.2. Transfer Learning using T5

In the realm of transfer learning, models are equipped with a sophisticated comprehension of language sources, contextual nuances, and stylistic variations. This empowers them to transcend the limitations of task-specific training and effectively tackle the intricacies present across diverse domains [208]. Adding to the efficacy of transfer learning is the T5 model, which has garnered significant attention for its innovative text-to-text framework utilized in both training and encoding textual data. Initially, the T5 model undergoes pre-training on vast corpora, allowing it to capture the inherent patterns and complexities of language structures. Subsequently, fine-tuning is performed to tailor the model for specific task objectives. Numerous studies have delved into various models built upon the T5 architecture, as documented in [211], [212], [213]. The architecture of T5 model is depicted in Fig.7.1.

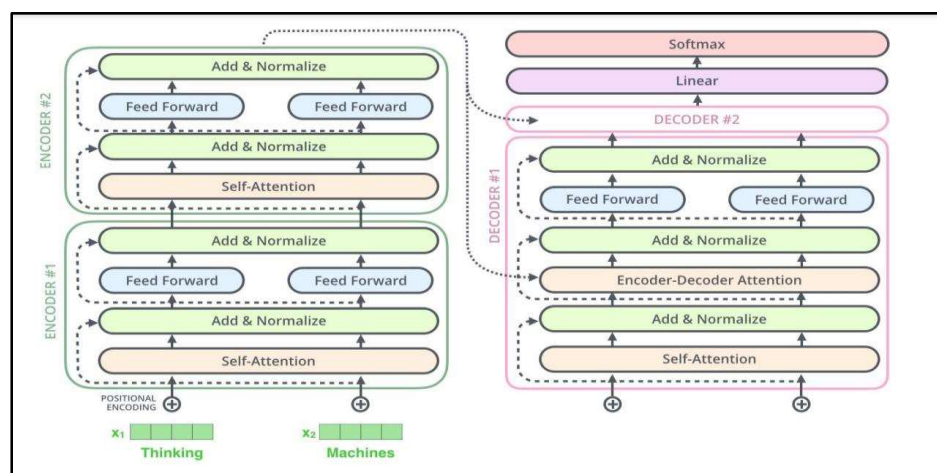


Fig.7.1. Architecture of T5 model

### 7.2.3 Long Short – term memory and convolution neural network

Conventional RNNs encounter difficulties in maintaining long-term dependencies due to the vanishing gradient problem. To overcome this challenge, LSTMs were

introduced. Memory cells and gating mechanisms were incorporated that allow them to selectively retain or forget information over time. These memory cells are equipped with three gates: the forget gate, input gate, and output gate, in addition to the memory cell itself. Each gate is responsible for regulating the flow of information into and out of the memory cell, enabling LSTMs to effectively capture long-range dependencies in sequential data. Specifically, the forget gate determines which information from the previous time step should be discarded, while the input gate controls which new information should be stored in the memory cell. The memory cell maintains the current state of the network, and the output gate determines which information from the memory cell should be passed on to the next time step [61]. By integrating these gating mechanisms, LSTMs are capable of learning and retaining information over extended sequences, making them well-suited for tasks such as natural language processing, time series prediction, and speech recognition. The four gates are represented mathematically as:

The new cell memory,  $C_t$ , can be computed, given an old memory,  $C_{t-1}$  as:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (7.1)$$

Forget Gate:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (7.2)$$

Memory Gate:

$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (7.3)$$

Input Gate:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (7.4)$$

Output Gate:

$$o_t = \sigma(W_o x_t + U_o h_{t-1}) \quad (7.5)$$

Moreover, LSTMs can be effectively combined Convolutional Neural Networks. LSTMs excel at learning and capturing long-term dependencies in sequential data, making them particularly well-suited for tasks such as language translation, speech recognition, and text classification. Fig. 7.2 presents the combined approach of CNN and LSTM architecture.

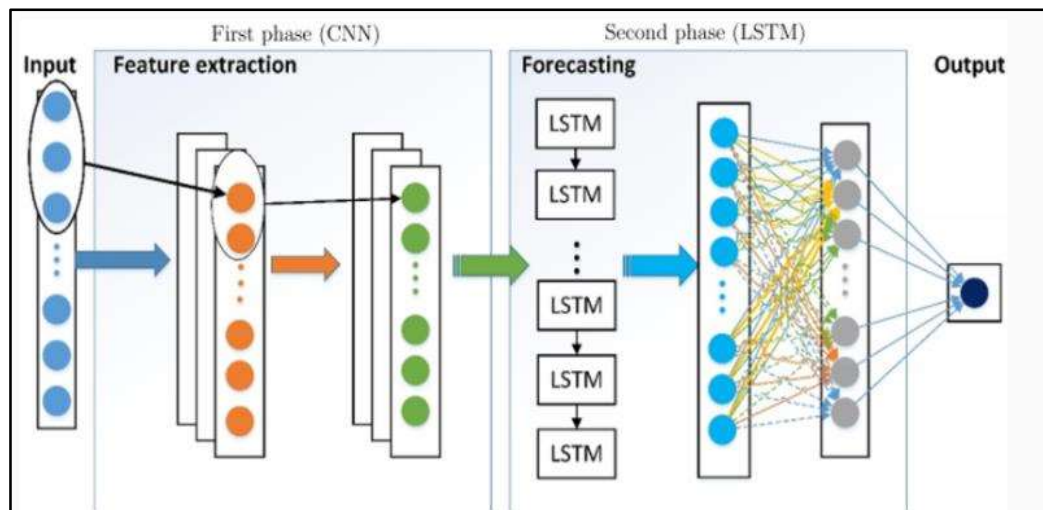


Fig.7.2. Architecture of CNN-LSTM model for feature enhancement

### 7.3. Proposed Framework Hybrid T5-LSTM Hate Speech Classification Framework (T5-LSTM HSCF)

This section provides an overview of the proposed framework and elaborates on the central algorithm utilized for hate speech detection. The proposed framework is depicted in Fig.7.3.

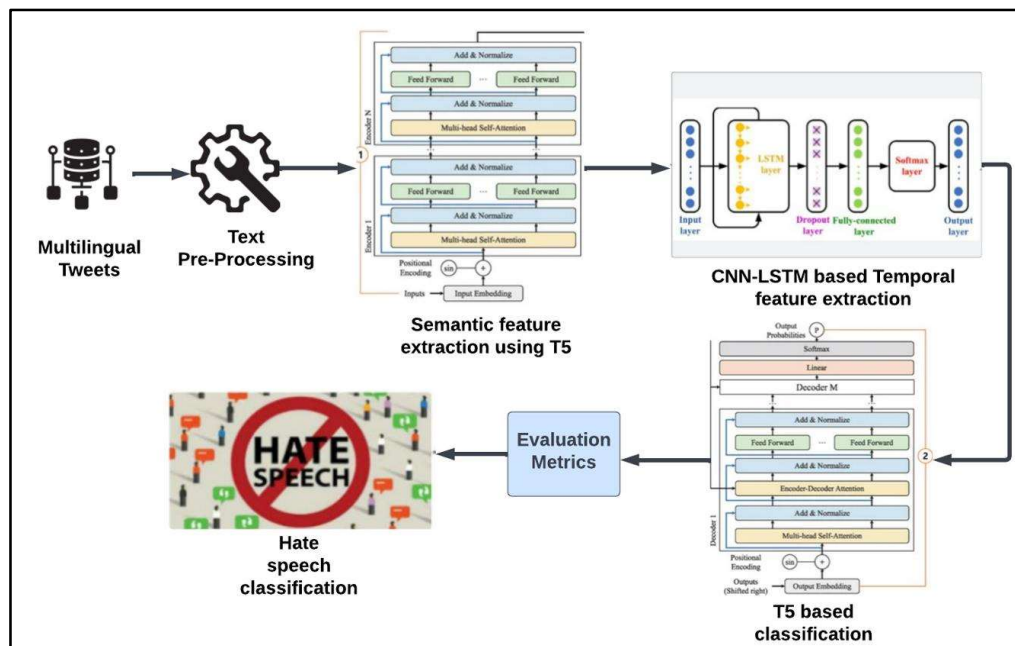


Fig. 7.3. Proposed Framework T5-LSTM HSCF

### 7.3.1 T5-LSTM hybrid model

The Text-to-Text Transfer Transformer (T5) model is a state-of-the-art language model that excels in various natural language processing tasks, including hate speech classification. T5 is based on the transformer architecture and employs a text-to-text framework, where both the input and output are text sequences. The transfer architecture consists of multiple layers of self-attention mechanisms and feedforward neural networks.

First, T5 represents the input text as a sequence of tokens

$$X = \{x_1, x_2, \dots, x_n\} \quad (7.6)$$

where  $x_i$  represents the  $i$ -th token in the input sequence.

These tokens are obtained through tokenization and is assigned a unique token ID.

- The input tokens  $X$  are passed through multiple transformer layers to encode the contextual information of the input sequence.

Let's denote the output of the transformer layer as  $H_L$ .

- The output of the final transformer layer  $H_L$  will contain rich contextual representations of the input tokens, capturing their semantic meaning and relationships within the context of the entire input sequence.
- The contextual representations  $H_L$  are then fed as an input to architecture of CNN-LSTM model for further feature refinement.
- The LSTM model processes the output representation of T5 model  $H_L$  over a sequence of time steps. At each time step, the LSTM unit performs a series of operations to capture the temporal dependencies and extract features from the input representation.
- The output of the LSTM model is a sequence of hidden states, denoted as  $LSTM_{Output} = \{h_1, h_2, \dots, h_n\}$

where  $n$  represents the number of time steps or tokens in the input sequence.

- Then, the  $LSTM_{Output}$  is fed into a classification unit of T5 model which consists of a linear layer followed by a softmax activation function.
- Let, the logits  $Z$  for each class  $c$  are computed as:

$$Z_c = W_c \cdot H_L + b_c \quad (7.7)$$

where,

$W_c$  and  $b_c$  are the weight matrix and bias vector for the  $c$ -th class, respectively.

T5 model is trained and uses a cross-entropy loss function to calculate the loss between the predicted logits and the ground truth labels. The loss function is defined as:

$$L = -N \sum_i \sum_c C_{y^i, c} \cdot \log(y^i, c) \quad (7.8)$$

where  $N$  is the number of samples,  $C$  is the number of classes,  $y^i, c$  is the ground truth label for sample  $i$  and class  $c$ , and  $y^i, c$  is the predicted probability of sample  $i$  belonging to class  $c$ .

The T5 model is trained using backpropagation and gradient descent optimization to minimize the cross-entropy loss.

- The model is then fine-tuned for hate speech classification, the model's weights are updated based on hate speech-related data, enabling it to learn and adapt specifically to this task.
- The output of the model is then evaluated using performance metrics such as precision, recall, f-1 score and support.

### 7.3.2. Proposed Pseudocode of the approach

The algorithm of the proposed approach is as follows:

### 7.4. Implementation

In this section, the proposed approach is evaluated on Multilingual datasets.

#### 7.4.1. Datasets Used

**English Languages Tweets-** This dataset comprises of 24,784 tweets categorized in 3 different classes i.e. class 0,1 and 2, where class 0 represents hate speech tweets, class 1 represents offensive tweets and class 2 represents neither of them. These two datasets, Islamic and Tumblr datasets are multilingual unlabelled data consisting of hate speech tweets [214]

**Hindi Languages Tweets-** This dataset having tweets in Hindi/English mixed language tweets. It is a labelled dataset comprising of 4579 tweets. Out of 4579 tweets, 1662 tweets are labelled as Hate speech whereas 2919 are labelled as non-hate speech [215].

**Multilingual Datasets** – A standardized multilingual dataset is constructed based on RAKE and the Twitter web API. The dataset consists 3620 tweets for 5 different languages such as Arabic, German French, English and Hindi-English Mix having two

labels hatred and non-hatred. The dataset was extracted from twitter. For feature, extraction RAKE method was used to extract significant unigrams and bigrams

**Algorithm: T5-LSTM HSCF**

*Input: Multilingual tweets*

*Output: Classified Hate or Non – Hate Speech*

*Step 1: Initialize T5 – CNN – LSTM Model*

*initialize T5 model weights*

*initialize CNN – LSTM model weights*

*Step 2: Set Hyperparameters:*

*SELF.MODEL = T5ForConditionalGeneration.from\_pretrained('t5 – base')*

*SELF.TOKENIZER = T5Tokenizer.from\_pretrained('t5 – base')*

*SELF.CNN = nn.Conv1d(in\_channels = 768, out\_channels = 64, kernel\_size = 3)*

*SELF.LSTM = nn.LSTM(input\_size = 64, hidden\_size = 32, batch\_first = True)*

*SELF.FC = nn.Linear(32, 3) # As output has 3 classes*

*Step 3: For iteration in range(max\_iter):*

*For each tweet in Multilingual tweets:*

*H\_L = T5\_model(tweet)*

*logits = CNN\_LSTM\_model(H\_L)*

*loss = calculate\_loss(logits, ground\_truth\_labels)*

*train\_model(tweet, ground\_truth\_label)*

*fine\_tune\_for\_hate\_speech(tweet, hate\_speech\_label)*

*end for*

*end for*

*Step 4: For each tweet in Multilingual tweets:*

*H\_L = T5\_model(T5\_input)*

*logits = CNN\_LSTM\_model(H\_L)*

*metrics = calculate\_metrics(logits, ground\_truth\_labels)*

*end for*

*return metrics*

By prioritizing words that occurred frequently and were close to each other, RAKE generated composite scores for each candidate word and most significant keyword phrases were extracted based on highest score. Further, the Twitter API was employed to further refine the text collected in the initial step, utilizing the identified unigrams and bigrams as hashtags for searching tweets. Some of the hashtag words included “opfers,” a German word meaning “Victim” in English, and “missbraucht\_haben,” which translates to “have\_abused.” Subsequently, 500 texts were extracted for each language using different keywords. The tweets underwent cleaning processes, which included the removal of special characters, URLs, mentions, and emojis, as well as lowercase conversion and the removal of person\_names. Stop words were retained to preserve text integrity for annotation analysis. Furthermore, three annotators with domain expertise such as language expert, a psychologist, and a research scholar labelled the extracted dataset. Since the dataset involved multiple languages, an initial generalized approach was adopted, using two labels: 0 for non-hate and 1 for hate. In the final step, the annotated data with text files was stored in CSV format with the data for 5 different languages. The complete method is explained in Chapter 4. The sample tweets for English and Hindi Languages are presented in Table 7.1. Further, Table 7.2. presents tweets for 5 different languages such as Arabic, German, French, English and Hindi-English Mix having two labels hatred and non-hatred.

**Table 7.1. Tweets for English and Hindi Languages**

S.No	Dataset Name	Text/Tweet	Class/Category
1	Twitter Dataset [216]	<i>“ My grandma used to call me .monkey all the time... she did refer to a broken bottle as a nigger_knife</i>	0 –Hate Speech
2	Twitter Dataset [217]	<i>You can call me Fireman booke cause I turn the hoes on.</i>	3- offensive
3	Twitter Dataset [217]	<i>Your teeth are like the stars.” “Aww thanks!” “Yeah... yellow...and white &amp; far near/ away from each other.”</i>	4- No offensive
4	Twitter Hindi Dataset [218]	<i>Doctor sab sahi me ke PhD (in hate politics) wale. Bhai padhe likhe ho fir kyu ye sab baate karte ho. Tum bas bowling khelo, aur maje lo.</i>	No



5.	Twitter Dataset [218]	Hindi	<i>Sarkar banne ke bad Hindu hit me ek bhi faisla Jo bjp ke dwara liya gaya ho, bjp ko gay, gobar, mandir, masjid aur nafrat faila kar vot chahiye</i>	Yes
----	-----------------------	-------	--------------------------------------------------------------------------------------------------------------------------------------------------------	-----

**Table 7.2. Tweets for different Languages**

S.No	Dataset Name	Text/Tweet	Class/Category
1	Multilingual Dataset German Text	<i>“Ich hatte im ersten Moment gedacht, dass sind Opfer, die ab 2015, von den Kriminellen Migranten begangen wurden, Vergewaltigung, Mörder, usw. gut das ich mich geirrt habe”</i>	0 –Hatred
2	Multilingual Dataset German Text	“Die Zeit der #Narren und #Jecken, der Höhepunkt der fünften Jahreszeit ist gekommen. Auch bei 140ies e Hause ist #Karnevalsstimmung ausgebrochen! Unsere Mitarbeiter haben sich verkleidet, besuchten Kindergärten der Umgebung und bescherten 140ies emit Krapfen! #Karneval #karneval2020”	1- Non Hatred
3	Multilingual Dataset Arabic Text	-- المصدر / صفحة (صوت الجهاد) في 12/1/2010 موقع رسمي لإمارة أفغانستان الإسلامية - طالبان	0 –Hatred
4	Multilingual Dataset Arabic Text	طالبان قاري محمد يوسف (احمدی) للمناطق الجنوب الغربية والشمال الغربية في البلاد هاتف	1-Non Hatred

#### 7.4.2. Steps for Implementation

##### Step 1: Text Preprocessing

For the demonstration, only English language tweets were considered. Following the basic data cleaning process, Fig. 7.4 displays the first five tweets.

```

0 RT mayasolovely As woman shouldnt complain cl...
1 RT mleeew17 boy dats coldtyga dwn bad cuffin d...
2 RT UrKindOfBrand Dawg RT 80sbaby4life You eve...
3 RT CGAnderson vivabased look like tranny
4 RT ShenikaRoberts The shit hear might true mi...

```

**Fig 7.4. First Five Tweets after Cleaning**

The tweets have been categorized and organized into distinct groups, as illustrated in Table 7.3, After removing of duplicate samples, left with 19,151 instances for offensive language, 4,160 instances, and 1,429 instances, respectively. Total sample of each category are shown in Fig. 7.5.

**Table 7.3. Number of each sample into different categories**

Category	Number of Samples
Offensive_language	19190
Neither	4163
Hate_speech	1430

```

Examples:
Total: 24740
hate: 1429 (5.78% of total)

Examples:
Total: 24740
Ofensive: 19151 (77.41% of total)

Examples:
Total: 24740
Neither: 4160 (16.81% of total)

```

**Fig. 7.5. Total percent of each sample Category**

## Step 2: Semantic feature extraction using T5 and Classification using LSTM-CNN

The dataset displayed a significant imbalance, necessitating careful handling. Subsequently, the dataset was partitioned into training and testing sets, with 70% assigned for training and 30% reserved for testing. As discussed, a transfer learning approach was utilized by the primary authors for model development, involving the

T5 model. Additionally, LSTM and CNN architectures were employed for classification purposes. The training process consisted of applying these models to the dataset. Fig. 7.6 illustrates the training steps.

```
SELF.TOKENIZER = T5Tokenizer.from_pretrained()
SELF.CNN = nn.Conv1d(in_channels=768, out_channels=64, kernel_size=3)
SELF.LSTM = nn.LSTM(input_size=64, hidden_size=32, batch_first=True)
SELF.FC = nn.Linear(32, 3) # AS OUTPUT HAS 3 CLASSES
SELF.VALIDATION_STEP_OUTPUTS = []
```

**Fig. 7.6. Various steps involved in training the data**

The model is trained using 10-fold validation based on these parameters, and the various parameters are set as depicted in eq. (7.9).

```
(model_name_or_path='t5-base', tokenizer_name_or_path='t5-base',
max_seq_length=512, learning_rate=3e-4, weight_decay=0.0,
adam_epsilon=1e-8, warmup_steps=0,
train_batch_size=8, eval_batch_size=8, num_train_epochs=2, gradient_acc
umulation_steps=1, n_gpu=1, early_stop_callback=False, seed=42)
```

(7.9)

Following that, the model was trained by incorporating various parameters, specifically employing the T5, LSTM, and CNN architectures. This comprehensive approach ensured that the model was equipped with a diverse set of features and capabilities for effective training and balancing the data skewness. Fig. 7.7 depicted the model fitting and parameter assignment for training the dataset.

*trainer.fit(model)* (7.10)

pytorch_lightning.callbacks.model_summary:			
	Name	Type	Params
0	model	T5ForConditionalGeneration	222 M
1	cnn	Conv1d	147 K
2	lstm	LSTM	12.5 K
3	fc	Linear	99
-----			
223 M		Trainable params	
0		Non-trainable params	
223 M		Total params	
892.255		Total estimated model params size (MB)	

**Fig. 7.7. Model fitting with T5, LSTM and CNN**

After the model had been fitted with the specified parameters and diverse architectures, including T5, LSTM, and CNN, it was tested using the designated test data. The

evaluation metrics were then employed to assess the model's performance across multilingual datasets.

### 7.3.1. Evaluation Metrics

The performance of the proposed approach has been evaluated on various performance metrics namely; Precision, Recall, Accuracy, and F1-score. The mathematical formulas of various evaluation metrics are presented in eq.(7.11)-(7.13).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (7.11)$$

$$Precision = \frac{TP}{TP+FP} \quad (7.12)$$

$$Recall = \frac{TP}{TP+FN} \quad (7.13)$$

Where,  $TP$  – True Positives,  $FP$  – False Positives,  $TN$  – True Negatives,  $FN$  – False Negatives

## 7.5. Results and Discussions

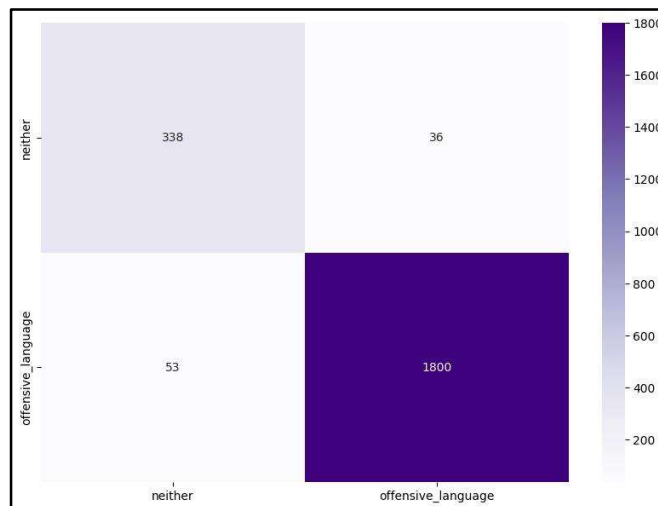
Addressing hate speech necessitates a comprehensive approach that leverages expertise from various disciplines, including linguistics, sociology, psychology, and technology. Collaboration with specialists in these fields is crucial to gain diverse perspectives on the issue and to devise effective solutions. Additionally, it is essential to maintain vigilance in monitoring the constantly evolving online landscape, which continually introduces new platforms and trends. Therefore, the proposed framework was validated using multilingual data encompassing English, Hindi, and other languages such as Arabic, German, and French. The model underwent training using approximately 29,634 tweets for different languages and was subsequently tested on the provided sample text data shown in Table 7.4.

**Table 7.4. Sample Test Results for different tweets in different languages**

Text Data	Actual Category	Predicted category
Bands make dance stamps make twerk If Romney becomes president hoes work	offensive_language	offensive_language
This new shit im workin on yea nigguh	offensive_language	offensive_language
RT ItsChee Whipped httpstcouCVHrjatv	Neither	Neither
RT BleacherReport VIDEO Dont hurt yourself kobebryant hits jumper over rookie proceeds talk trash httpcomlnvAim ht	Neither	Neither

Sarkar banne ke bad Hindu hit me ek bhi faisla Jo bjp ke dwara liya gaya ho,bjp ko gay,gobar,mandir,masjid aur nafrat faila kar vot chahiye	Yes	Yes
Se Cancer K Mareez K Elaj Karwane Lahor Jane Wale Mutadid Sheri Panjab Poliec K Hate Chad Gai.	No	NO
<i>Ich hatte im ersten Moment gedacht, dass sind Opfer,die ab 2015,von den Kriminellen Migranten begangen wurden, Vergewaltigung, Mörder,usw.gut das ich mich geirrt habe</i>	Hatred	Hatred
طالبان قاري محمد يوسف (احمدی) للمناطق الجنوب الغربية والشمال الغربية في البلاد هاتف	Non-Hatred	Hatred
-- المصدر / صفحة (صوت الجهاد) في 12/1/2010 موقع رسمي لإمارة أفغانستان الإسلامية - طالبان	Hatred	Hatred

A similarity matrix was utilized to represent the pairwise similarities or distances between data points in a given dataset. It depicted the relationships among instances within the feature space. The similarity matrix for the English tweets for the different classes was displayed in Fig. 7.8.



**Fig.7.8. Similarity Matrix for English Tweets**

The "loss over learning epoch for different cycles" described the tracking of the variation in the model's loss function during training across multiple epochs and cycles. It aided in assessing the convergence and performance dynamics of the model,

with a decreasing loss indicating learning progress and an increasing or stagnant loss suggesting potential issues. Monitoring this trend assisted in optimizing the model's performance. Fig. 7.9 displayed the loss over learning epoch for different cycles for the applied models.

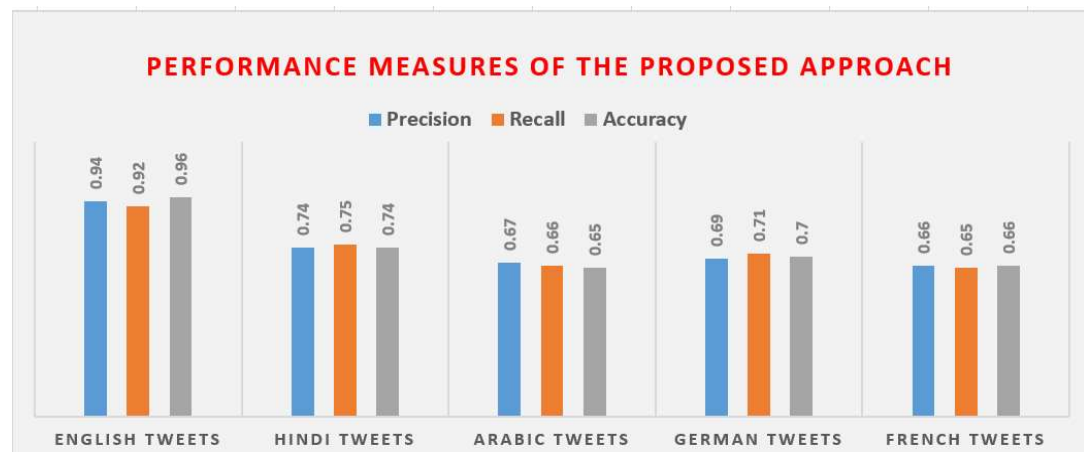


**Fig. 7.9.** Shows the loss over learning epoch on different cycles.

The various performance metrics evaluated for the proposed approach are depicted in Table 7.5 and Fig. 7.10.

**Table 7.5. Classification Accuracy by the Proposed Approach**

Proposed Approach		Precision	Recall	F1-score	Support
<b>English Tweets</b>	Neither	0.86	0.90	0.88	374
	Offensive_language	0.98	0.97	0.98	1853
	Accuracy	0.96			2227
<b>Hindi Tweets</b>	Yes	0.78	0.76	0.76	245
	No	0.67	0.68	0.68	347
	Accuracy	0.74			592
<b>Arabic Tweets</b>	Hatred	0.68	0.64	0.67	132
	Non-Hatred	0.67	0.63	0.65	53
	Accuracy	0.65			185
<b>German Tweets</b>	Hatred	0.70	0.72	0.71	73
	Non-Hatred	0.65	0.64	0.65	56
	Accuracy	0.70			119
<b>French Tweets</b>	Hatred	0.69	0.65	0.68	98
	Non-Hatred	0.62	0.64	0.63	44
	Accuracy	0.66			142



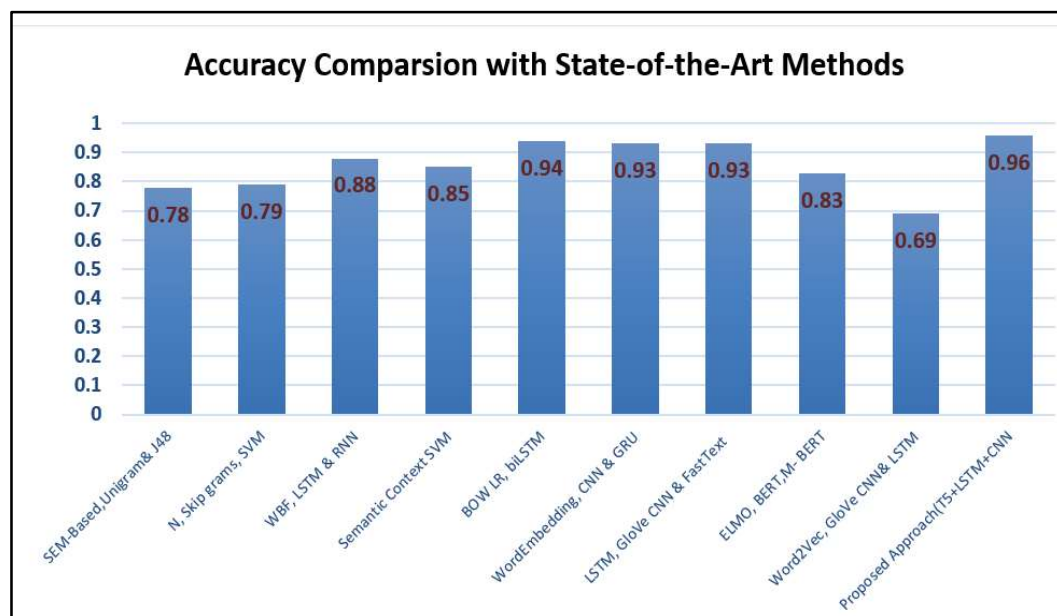
**Fig. 7.10. Performance Measures of the proposed approach**

Table 7.6 and Fig. 7.11 displayed the previous research on English tweets, which was compared with multiple parameters, including social media platforms, features, and accuracy, with the current proposed approach.

**Table 7.6. Comparison with the other techniques for English Tweets**

Author and Year	Platform	Type	Features Representation	Accuracy
<b>Watanabe et al. (2018)</b> [49]	Twitter	Hate, Offensive	Sentiment-Based, Semantic, Unigram-J48graft	0.78
<b>Malmasi &amp; Zampieri (2018)</b> [198]	Twitter	Hate, Offensive	N-grams, Skip-grams, hierarchical, word clusters, RBF kernel, SVM	0.79
<b>Pitsilis et al. (2018)</b> [91]	Twitter	Racism or Sexism	Word-based frequency and LSTM, vectorization RNN	0.88
<b>Fernandez and Alani (2018)</b> [199]	Twitter	Radicalization	Semantic Context SVM	0.85
<b>Ousidhoum et al. (2019)</b> [159]	Twitter	Sexual orientation,	BOW LR, bi-LSTM	0.94

		Religion, Disability		
<b>Zhang and Luo (2019)</b> [85]	Twitter	Racism, Sexism	Word- Embeddings CNN + GRU	0.93
<b>Badjatiya et al. (2017)</b> [106]	Twitter	Hate Speech	Random embedding, LSTM, GBDT, GloVe CNN, FastText,	0.93
<b>Dowlagar and Mamidi (2021)</b> [200]	Twitter	Hate Speech	ELMO BERT, Multilingual BERT	0.83
<b>Rizos et al. (2019)</b> [201]	Twitter	Hate Speech	Word2Vec, GloVe CNN, LSTM, GRU	0.69
<b>Proposed Approach(T5+LSTM+CNN)</b>	<b>Twitter</b>	<b>Hate Speech</b>	<b>T5+LSTM+CNN</b>	<b>0.96</b>



**Fig. 7.11. Comparisons with the state-of-the-art methods for English language Dataset**

The proposed approach is also evaluated on “Hindi-English mix tweets” text consisting of tweet ids and the equivalent annotations. In previous researches, several



features are used for hate speech classification. The accuracy attained on previous features are compared with the current approach for Hindi tweets is depicted in Table 7.7.

**Table 7.7. Comparison with the other techniques for the Hindi tweets**

<b>Features</b>	<b>Accuracy</b>
<b>Character</b>	66.8
<b>Word-n-gram</b>	69.9
<b>Punctuation</b>	63.2
<b>Lexicon</b>	63.8
<b>All Features</b>	71.7
<b>Proposed Approach- T5 +LSTM+CNN</b>	<b>74.7</b>

The recently curated dataset consisted of tweets in five distinct languages, namely Arabic, German, French, English, and Hindi-English Mix, with two designated labels: hatred and non-hatred. This dataset provided advantages in addressing out-of-vocabulary words. At present, this multilingual hate speech dataset was considered a valuable asset for the research community. Since this dataset had not been utilized in prior studies, comparisons were not included in the current work.

## **7.6. Conclusion and Future Work**

The pervasive nature of social computing platforms has led to the widespread dissemination of hate speech, highlighting the urgent need for effective detection and classification methods. The proposed work addresses this challenge by proposing a robust methodology that analyses diverse datasets, including Twitter posts in English and Hindi, as well as a multilingual dataset. The Hybrid T5-LSTM Hate Speech Classification Framework (T5-LSTM HSCF) is proposed that accommodates multiple languages and recognizes variations in hate speech across diverse linguistic and cultural contexts. By integrating the capabilities of Text-to-text transfer and Long Short-Term Memory models, our hybrid approach achieves significant improvements in accuracy rates, with results of 96%, 76%, and 71% for labelled text in English, Hindi, and the multilingual dataset, respectively. This advancement represents a crucial step forward in addressing the complexities of hate speech classification and enhancing our understanding of this pervasive issue in online communication. The proposed framework has shown promising results for English and Hindi, in future, further exploration into additional languages will be focused to enhance the applicability and effectiveness of hate speech detection on a global scale.

## CHAPTER 8

### CONCLUSION AND FUTURE SCOPE

This chapter presents a comprehensive summary of the research work done. It includes the research summary of the work done in Section 8.1. The chapter also presents the future aspects of the research work performed in Section 8.3 and how the study can help the future researchers in the said domain.

#### **8.1. Research Summary**

The advent of the internet and social media platforms has revolutionized communication, allowing individuals worldwide to connect and exchange ideas. However, despite the positive aspects of enhanced connectivity and information exchange, the digital sphere has also experienced a surge in detrimental and discriminatory content, commonly referred to as hate speech. Hate speech encompasses any form of communication, whether verbal, written, or visual, that conveys hatred, animosity, or bias towards individuals or communities based on attributes such as race, ethnicity, religion, gender, sexual orientation, disability, or other protected characteristics. It can take various forms, ranging from explicit threats and derogatory language to more subtle expressions of prejudice, micro-aggressions, and coded messages. When such occurrences take place on social media platforms, in blog posts, or as a means to target individuals, it is termed as Online Hate Speech (OHS) which poses significant challenges to online safety and inclusivity. Addressing this phenomenon requires sophisticated automated systems capable of detecting and mitigating harmful content effectively. Detecting hate speech using machine learning and deep learning techniques is pivotal in countering the proliferation of harmful content on online platforms, especially considering the challenges posed by the sheer volume of user-generated content. The process typically involves collecting labelled datasets from diverse sources like social media, forums, or news articles, followed by preprocessing steps such as cleaning and standardizing the text. Feature extraction then identifies relevant linguistic features like word frequencies and semantic embeddings, which serve as input to ML algorithms. DL techniques, particularly neural networks like CNNs and RNNs, have shown remarkable performance in hate speech detection due to their ability to learn hierarchical representations of data. Despite advancements, challenges such as dataset imbalance, biases, and ethical implications persist, prompting ongoing research into more robust and adaptable models capable of generalizing across languages and cultures. Efforts focus on domain adaptation, transfer learning, and cross-lingual approaches to enhance hate speech detection systems' effectiveness and inclusivity. In this work, advanced machine learning and

deep learning techniques tailored for hate speech detection are explored, aiming to develop more accurate, efficient, and scalable solutions.

Despite significant progress in online hate speech detection, several research gaps persist, warranting further exploration. Firstly, there is a need to refine and optimize supervised ML and DL methods by developing advanced feature extraction techniques and enhancing algorithm performance for more accurate classification. Secondly, addressing multilingual challenges is crucial, as existing models predominantly focus on English content, necessitating the development of techniques adaptable to diverse linguistic contexts. Thirdly, improving contextual understanding is vital, as current models may struggle to differentiate between hate speech, offensive language, and non-hate content, especially in dynamic online environments. Fourthly, scalability and adaptability are key considerations, requiring enhancements to ensure effective detection across various platforms and datasets. Additionally, diversifying feature representation by incorporating knowledge-based and semantic features alongside lexicon-based ones can enhance model accuracy. Moreover, bridging language gaps by including languages beyond English and leveraging unsupervised learning techniques can expedite hate speech detection efforts. Addressing these gaps can advance the field and contribute to the development of more effective and ethical hate speech detection systems. To bridge existing research gaps, various methods have been proposed in this work. A novel algorithm has been introduced HateSwarm, combining Particle Swarm Optimization and Genetic Algorithm techniques to enhance hate speech classification accuracy. Further, the 'HateDetector: Multilingual Hate Speech Detection Technique' has been introduced, utilizing Bidirectional Encoder Representations from Transformers and Multi-Layer Perceptron for precise tweet content analysis. Next, to address the absence of standardized multilingual hate speech datasets, a methodology has been proposed leveraging Rapid Automatic Keyword Extraction and the Twitter web API to create adaptable datasets across languages. Then, a hybrid approach integrating Bidirectional Encoder Representations from Transformers with topic modelling has been proposed for cross-platform prediction. Further, to establish a robust methodology for hate speech analysis across diverse linguistic contexts, the Hybrid T5-LSTM Hate Speech Classification Framework is proposed.

In conclusion, this work, represents a significant contribution to the ongoing efforts to combat online hate speech. By addressing key research gaps and proposing innovative solutions, this study has provided valuable insights into the development of more effective and ethical hate speech detection systems. Ultimately, the goal is to foster safer and more inclusive digital environments for users worldwide, safeguarding the principles of freedom of expression and respect for human dignity. Continued collaboration among researchers, policymakers, technology companies, and civil society organizations is essential to achieving this overarching objective.

## 8.2. Future Aspects

Following are the future aspects of the research work performed:

- More elaborate work can be done in the feature engineering part. Combining suitable feature extraction and selection methods with advanced ML and Deep Learning models can improve the results.
- In order to advance research in the field of Online Hate Speech (OHS), it is crucial to develop a multimodal dataset that encompasses diverse languages and modes of communication.

## REFERENCES

- [1] K. Dinakar, “Modeling the Detection of Textual Cyberbullying,” in *2011, Association for the Advancement of Artificial Intelligence*, 2011, pp. 11–17.
- [2] Z. Zhang, “Hate Speech Detection : A Solved Problem ? The Challenging Case of Long Tail on Twitter,” *SEMANTIC WEB IOS Press*, vol. 1, no. 0, pp. 1–5, 2018.
- [3] J. C. Pereira-Kohatsu, L. Quijano-Sánchez, F. Liberatore, and M. Camacho-Collados, “Detecting and monitoring hate speech in twitter,” *Sensors (Switzerland)*, vol. 19, no. 21, pp. 1–37, 2019, doi: 10.3390/s19214654.
- [4] A. Guiora and E. A. Park, “Hate Speech on Social Media,” *Philosophia (United States)*, vol. 45, no. 3, pp. 957–971, 2017, doi: 10.1007/s11406-017-9858-4.
- [5] M. Corazza, S. Menini, E. Cabrio, S. Tonelli, and S. Villata, “A Multilingual Evaluation for Online Hate Speech Detection,” *ACM Trans Internet Technol.*, vol. 20, no. 2, 2020, doi: 10.1145/3377323.
- [6] N. Ousidhoum, Z. Lin, H. Zhang, Y. Song, and D. Y. Yeung, “Multilingual and multi-aspect hate speech analysis,” *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp. 4675–4684, 2020, doi: 10.18653/v1/d19-1474.
- [7] K. Sreelakshmi, B. Premjith, and K. P. Soman, “Detection of Hate Speech Text in Hindi-English Code-mixed Data,” *Procedia Comput Sci*, vol. 171, no. 2019, pp. 737–744, 2020, doi: 10.1016/j.procs.2020.04.080.
- [8] T. Mandl, “Overview of the HASOC Track at FIRE 2020 : Hate Speech and Offensive Language Identification in Tamil , Malayalam , Hindi , English and German,” pp. 29–32, 2020.
- [9] M. Mozafari, R. Farahbakhsh, and N. Crespi, “Cross-Lingual Few-Shot Hate Speech and Offensive Language Detection Using Meta Learning,” *IEEE Access*, vol. 10, pp. 14880–14896, 2022, doi: 10.1109/ACCESS.2022.3147588.
- [10] M. Ridenhour, A. Bagavathi, E. Raisi, and S. Krishnan, “Detecting Online Hate Speech: Approaches Using Weak Supervision and Network Embedding Models,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12268 LNCS, pp. 202–212, 2020, doi: 10.1007/978-3-030-61255-9\_20.
- [11] L. Wang, J. Niu, and S. Yu, “SentiDiff: Combining Textual Information and Sentiment Diffusion Patterns for Twitter Sentiment Analysis,” *IEEE Trans*

- Knowl Data Eng*, vol. 32, no. 10, pp. 2026–2039, 2020, doi: 10.1109/TKDE.2019.2913641.
- [12] P. Alonso, R. Saini, and G. Kovács, “Hate Speech Detection Using Transformer Ensembles on the HASOC Dataset,” in *Speech and Computer*, A. Karpov and R. Potapova, Eds., Cham: Springer International Publishing, 2020, pp. 13–21.
- [13] Y. Wang, G. Huang, J. Li, H. Li, Y. Zhou, and H. Jiang, “Refined Global Word Embeddings Based on Sentiment Concept for Sentiment Analysis,” *IEEE Access*, vol. 9, pp. 37075–37085, 2021, doi: 10.1109/ACCESS.2021.3062654.
- [14] R. Cao, R. K.-W. Lee, and T. Hoang, “DeepHate: Hate Speech Detection via Multi-Faceted Text Representations,” 2020, pp. 11–20. doi: 10.1145/3394231.3397890.
- [15] M. F. Mridha, Md. A. H. Wadud, Md. A. Hamid, M. M. Monowar, M. Abdullah-Al-Wadud, and A. Alamri, “L-Boost: Identifying Offensive Texts From Social Media Post in Bengali,” *IEEE Access*, vol. 9, pp. 164681–164699, 2021, doi: 10.1109/ACCESS.2021.3134154.
- [16] N. DePaula, K. J. Fietkiewicz, T. J. Froehlich, A. J. Million, I. Dorsch, and A. Ilhan, “Challenges for social media: Misinformation, free speech, civic engagement, and data regulations,” in *Proceedings of the Association for Information Science and Technology*, 2018, pp. 665–668. doi: 10.1002/pra2.2018.14505501076.
- [17] R. Varade and V. Pathak, “Detection of Hate Speech in Hinglish Language,” 2020, pp. 265–276. doi: 10.1007/978-981-15-1884-3\_25.
- [18] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, “Hate Speech Detection with Comment Embeddings,” in *Proceedings of the 24th International Conference on World Wide Web*, in WWW ’15 Companion. New York, NY, USA: Association for Computing Machinery, 2015, pp. 29–30. doi: 10.1145/2740908.2742760.
- [19] T. Davidson, D. Warmesley, M. W. Macy, and I. Weber, “Automated Hate Speech Detection and the Problem of Offensive Language,” in *International Conference on Web and Social Media*, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:1733167>
- [20] F. Miró-Llinares, A. Moneva, and M. Esteve, “Hate is in the air! But where? Introducing an algorithm to detect hate speech in digital microenvironments,” *Crime Sci*, vol. 7, no. 1, pp. 1–12, 2018, doi: 10.1186/s40163-018-0089-1.
- [21] J. Salminen, M. Hopf, S. A. Chowdhury, S. gyo Jung, H. Almerexhi, and B. J. Jansen, “Developing an online hate classifier for multiple social media

- platforms,” *Human-centric Computing and Information Sciences*, vol. 10, no. 1, pp. 1–34, 2020, doi: 10.1186/s13673-019-0205-6.
- [22] S. Biere and M. B. Analytics, “Hate Speech Detection Using Natural Language Processing Techniques,” *VRIJE UNIVERSITEIT AMSTERDAM*, p. 30, 2018.
- [23] P. Fortuna and S. Nunes, “A survey on automatic detection of hate speech in text,” *ACM Comput Surv*, vol. 51, no. 4, 2018, doi: 10.1145/3232676.
- [24] G. Diwhu, W. K. H. Ghdkw, R. I. D. Ihpdoh, and X. Vwxghqw, “Automated Detection of Hate speech towards Woman on Twitter,” in *International Conference On Computer Science And Engineering*, 2018, pp. 7–10.
- [25] B. Gambäck and U. K. Sikdar, “Using Convolutional Neural Networks to Classify Hate-Speech,” in *Proceedings of the First Workshop on Abusive Language Online*, 2017, pp. 85–90.
- [26] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, “Deep Learning for Hate Speech Detection in Tweets,” in *arXiv:1706.00188v1 [cs.CL]*, 2017, p. 2.
- [27] J. H. Park and P. Fung, “One-step and Two-step Classification for Abusive Language Detection on Twitter,” in *Association for Computational Linguistics Proceedings of the First Workshop on Abusive Language Online, pages 41–45, Vancouver, Canada, July 30, 2017*, pp. 41–45.
- [28] Z. Waseem, “Are You a Racist or Am I Seeing Things ? Annotator Influence on Hate Speech Detection on Twitter,” in *Proceedings of 2016 EMNLP Workshop on Natural Language Processing and Computational Social Science*, 2016, pp. 138–142.
- [29] F. Del Vigna, A. Cimino, F. D. Orletta, M. Petrocchi, and M. Tesconi, “Hate me , hate me not : Hate speech detection on Facebook,” in *In Proceedings of the First Italian Conference on Cybersecurity (ITASEC17), Venice, Italy., 2017*, pp. 86–95.
- [30] A. Jha, “When does a Compliment become Sexist ? Analysis and Classification of Ambivalent Sexism using Twitter Data,” in *Proceedings of the Second Workshop on Natural Language Processing*, 2017, pp. 7–16.
- [31] T. Davidson, D. Warmesley, M. Macy, and I. Weber, “Automated Hate Speech Detection and the Problem of Offensive Language \*,” in *arXiv*, 2017.
- [32] W. Alorainy, P. Burnap, H. A. N. Liu, and M. L. Williams, ““ The Enemy Among Us ’: Detecting Cyber Hate Speech with Threats-based Othering Language Embeddings,” *ACM Transactions on the Web*, vol. 13, no. 3, 2019.

- [33] C. Nobata and J. Tetreault, “Abusive Language Detection in Online User Content,” in *International World Wide Web Conference*, 2016, pp. 145–153.
- [34] Z. Al and M. Amr, “Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach,” *springer Computing*, no. 0123456789, 2019, doi: 10.1007/s00607-019-00745-0.
- [35] S. Agarwal and A. Sureka, “But i did not mean it! - Intent classification of racist posts on tumblr,” in *Proceedings - 2016 European Intelligence and Security Informatics Conference, EISIC 2016*, IEEE, 2017, pp. 124–127. doi: 10.1109/EISIC.2016.032.
- [36] “CodaLab Competition.” [Online]. Available: <https://competitions.codalab.org/competitions/19935>.
- [37] “Detecting Insults in Social Commentary.” [Online]. Available: <https://www.kaggle.com/c/detecting-insults-in-social-commentary>
- [38] F. O. (2019) MacAvaney S, Yao H-R, Yang E, Russell K, Goharian N, “Hate speech detection: Challenges and solutions. PLoS ONE 14(8): e0221152. <https://doi.org/10.1371/journal.pone.0221152>.” [Online]. Available: <https://sites.google.com/view/trac1/shared-task>
- [39] Timothy Quinn, “Hatebase database.” [Online]. Available: <https://www.hatebase.org/>
- [40] G. Diwlu, W. K. H. Ghdkw, R. I. D. Ihpdoh, and X. Vwxghqw, “Automated Detection of Hate speech towards Woman on Twitter,” in *International Conference On Computer Science And Engineering*, 2018, pp. 7–10.
- [41] P. Charitidis, S. Doropoulos, S. Vologiannidis, I. Papastergiou, and S. Karakeva, “Towards countering hate speech against journalists on social media,” *Online Soc Netw Media*, vol. 17, p. 10, 2020, doi: 10.1016/j.osnem.2020.100071.
- [42] N. Albadi, M. Kurdi, and S. Mishra, “Are they our brothers? analysis and detection of religious hate speech in the Arabic Twittersphere,” in *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018*, IEEE, 2018, pp. 69–76. doi: 10.1109/ASONAM.2018.8508247.
- [43] A. Al-Hassan and H. Al-Dossari, “Detection of hate speech in Arabic tweets using deep learning,” *Multimed Syst*, no. 0123456789, 2021, doi: 10.1007/s00530-020-00742-w.
- [44] H. Mulki, H. Haddad, C. Bechikh Ali, and H. Alshabani, “L-HSAB: A Levantine Twitter Dataset for Hate Speech and Abusive Language,” pp. 111–118, 2019, doi: 10.18653/v1/w19-3512.



- [45] N. Ljubešić, T. Erjavec, and D. Fišer, “Datasets of Slovene and Croatian Moderated News Comments,” pp. 124–131, 2019, doi: 10.18653/v1/w18-5116.
- [46] P. Charitidis, S. Doropoulos, S. Vologiannidis, I. Papastergiou, and S. Karakeva, “Towards countering hate speech against journalists on social media,” *Online Soc Netw Media*, vol. 17, 2020, doi: 10.1016/j.osnem.2020.100071.
- [47] A. Schmidt and M. Wiegand, “A Survey on Hate Speech Detection using Natural Language Processing,” in *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 1–10. doi: 10.18653/v1/W17-1101.
- [48] E. Greevy and A. F. Smeaton, “Classifying racist texts using a support vector machine,” *Proceedings of Sheffield SIGIR - Twenty-Seventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 468–469, 2004, doi: 10.1145/1008992.1009074.
- [49] H. Watanabe, M. Bouazizi, and T. Ohtsuki, “Hate Speech on Twitter A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection,” *IEEE Access*, vol. PP, p. 1, 2018, doi: 10.1109/ACCESS.2018.2806394.
- [50] F. Del Vigna, A. Cimino, F. D. Orletta, M. Petrocchi, and M. Tesconi, “Hate me , hate me not : Hate speech detection on Facebook,” in *In Proceedings of the First Italian Conference on Cybersecurity (ITASEC17), Venice, Italy., 2017*, pp. 86–95.
- [51] A. Rodriguez, C. Argueta, and Y. L. Chen, “Automatic Detection of Hate Speech on Facebook Using Sentiment and Emotion Analysis,” *1st International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2019*, pp. 169–174, 2019, doi: 10.1109/ICAIIC.2019.8669073.
- [52] S. Agarwal and A. Sureka, “But i did not mean it! - Intent classification of racist posts on tumblr,” in *Proceedings - 2016 European Intelligence and Security Informatics Conference, EISIC 2016*, IEEE, 2017, pp. 124–127. doi: 10.1109/EISIC.2016.032.
- [53] Z. Waseem, “Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter,” 2016, pp. 138–142. doi: 10.18653/v1/W16-5618.
- [54] T. Lynn, P. T. Endo, P. Rosati, I. Silva, G. L. Santos, and D. Ging, “Data set for automatic detection of online misogynistic speech,” *Data Brief*, vol. 26, p. 104223, 2019, doi: 10.1016/j.dib.2019.104223.

- [55] B. Raufi and I. Xhaferri, “Application of Machine Learning Techniques for Hate Speech Detection in Mobile Applications,” in *2018 International Conference on Information Technologies (InfoTech-2018), IEEE Conference Rec. No. 46116 20-21 September 2018, St. St. Constantine and Elena, Bulgaria*, IEEE, 2018.
- [56] F. M. Plaza-Del-Arco, M. D. Molina-González, L. A. Ureña-López, and M. T. Martín-Valdivia, “Detecting Misogyny and Xenophobia in Spanish Tweets Using Language Technologies,” *ACM Trans Internet Technol*, vol. 20, no. 2, 2020, doi: 10.1145/3369869.
- [57] B. Pelzer, L. Kaati, and N. Akrami, “Directed digital hate,” in *2018 IEEE International Conference on Intelligence and Security Informatics, ISI 2018*, 2018, pp. 205–210. doi: 10.1109/ISI.2018.8587396.
- [58] R. Martins, M. Gomes, J. J. Almeida, P. Novais, and P. Henriques, “Hate speech classification in social media using emotional analysis,” in *Proceedings - 2018 Brazilian Conference on Intelligent Systems, BRACIS 2018*, 2018, pp. 61–66. doi: 10.1109/BRACIS.2018.00019.
- [59] R. Basak, S. Sural, N. Ganguly, and S. K. Ghosh, “Online Public Shaming on Twitter: Detection, Analysis, and Mitigation,” *IEEE Trans Comput Soc Syst*, vol. 6, no. 2, pp. 208–220, 2019, doi: 10.1109/TCSS.2019.2895734.
- [60] K. Sreelakshmi, B. Premjith, and K. P. Soman, “Detection of Hate Speech Text in Hindi-English Code-mixed Data,” *Procedia Comput Sci*, vol. 171, no. 2019, pp. 737–744, 2020, doi: 10.1016/j.procs.2020.04.080.
- [61] T. Davidson, D. Bhattacharya, and I. Weber, “Racial Bias in Hate Speech and Abusive Language Detection Datasets,” in *arXiv:1905.12516v1*, 2019, pp. 25–35. doi: 10.18653/v1/w19-3504.
- [62] A. Andreou, K. Orphanou, and G. Pallis, “MANDOLA : A Big-Data Processing and Visualization,” *ACM Trans Internet Technol*, vol. 20, no. 2, 2020.
- [63] J. H. Park and P. Fung, “One-step and Two-step Classification for Abusive Language Detection on Twitter,” in *Association for Computational Linguistics Proceedings of the First Workshop on Abusive Language Online, pages 41–45, Vancouver, Canada, July 30, 2017*, pp. 41–45.
- [64] D. Zimbra, A. Abbasi, D. Zeng, and H. Chen, “The State-of-the-Art in Twitter Sentiment Analysis,” *ACM Trans Manag Inf Syst*, vol. 9, no. 2, pp. 1–29, 2018, doi: 10.1145/3185045.
- [65] E. Mariconti *et al.*, “‘You know what to do’: Proactive detection of YouTube videos targeted by coordinated hate attacks,” *Proc ACM Hum Comput Interact*, vol. 3, no. CSCW, 2019, doi: 10.1145/3359309.

- [66] N. D. Gitari, Z. Zuping, H. Damien, and J. Long, "A Lexicon-based Approach for Hate Speech Detection," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, no. 4, pp. 215–230, 2015.
- [67] K. Dinakar, "Modeling the Detection of Textual Cyberbullying," in *2011, Association for the Advancement of Artificial Intelligence*, 2011, pp. 11–17.
- [68] L. Lima *et al.*, "Inside the right-leaning echo chambers: Characterizing gab, an unmoderated social system," in *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018*, 2018, pp. 515–522. doi: 10.1109/ASONAM.2018.8508809.
- [69] N. D. T. Ruwandika and A. R. Weerasinghe, "Identification of Hate Speech in Social Media," in *2018 International Conference on Advances in ICT for Emerging Regions (ICTer) : 273 - 278 Identification*, IEEE, 2018, pp. 273–278.
- [70] H. Liu, W. Alorainy, P. Burnap, and M. L. Williams, "Fuzzy multi-task learning for hate speech type identification," *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, pp. 3006–3012, 2019, doi: 10.1145/3308558.3313546.
- [71] N. A. Setyadi, M. Nasrun, and C. Setianingsih, "Text Analysis For Hate Speech Detection Using Backpropagation Neural Network," *2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, pp. 159–165, 2018, [Online]. Available: <https://api.semanticscholar.org/CorpusID:155106622>
- [72] N. I. Pratiwi, I. Budi, and I. Alfina, "Hate speech detection on Indonesian instagram comments using FastText approach," in *2018 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2018*, IEEE, 2019, pp. 447–450. doi: 10.1109/ICACSIS.2018.8618182.
- [73] F. M. Plaza-Del-Arco, M. D. González, L. Ureña-López, and M. Martín-Valdivia, "Comparing pre-trained language models for Spanish hate speech detection," *Expert Syst Appl*, vol. 166, p. 114120, 2021, doi: 10.1016/j.eswa.2020.114120.
- [74] H. K. Sharma, T. P. Singh, K. Kshitiz, H. Singh, and P. Kukreja, "Detecting Hate Speech and Insults on Social Commentary using NLP and Machine Learning," *International Journal of Engineering Technology Science and Research*, vol. 4, no. 12, pp. 279–285, 2017.
- [75] T. L. Sutejo and D. P. Lestari, "Indonesia Hate Speech Detection Using Deep Learning," *2018 International Conference on Asian Language Processing (IALP)*, pp. 39–43, 2018, [Online]. Available: <https://api.semanticscholar.org/CorpusID:59554228>

- [76] I. K. Lekea, "Detecting Hate Speech within the Terrorist Argument : A Greek Case," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, IEEE, 2018, pp. 1084–1091.
- [77] H. Liu, P. Burnap, W. Alorainy, and M. L. Williams, "A Fuzzy Approach to Text Classification with Two-Stage Training for Ambiguous Instances," *IEEE Trans Comput Soc Syst*, vol. 6, no. 2, pp. 227–240, 2019, doi: 10.1109/TCSS.2019.2892037.
- [78] J. Wang, W. Zhou, J. Li, Z. Yan, J. Han, and S. Hu, "An online sockpuppet detection method based on subgraph similarity matching," in *Proceedings - 16th IEEE International Symposium on Parallel and Distributed Processing with Applications, 17th IEEE International Conference on Ubiquitous Computing and Communications, 8th IEEE International Conference on Big Data and Cloud Computing, 11t*, IEEE, 2019, pp. 391–398. doi: 10.1109/BDCLOUD.2018.00067.
- [79] K. Wu, S. Yang, and K. Q. Zhu, "False rumors detection on Sina Weibo by propagation structures," *Proc Int Conf Data Eng*, vol. 2015-May, pp. 651–662, 2015, doi: 10.1109/ICDE.2015.7113322.
- [80] A. S. Saksesi, M. Nasrun, and C. Setianingsih, "Analysis Text of Hate Speech Detection Using Recurrent Neural Network," in *The 2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC) Analysis*, IEEE, 2018, pp. 242–248.
- [81] E. Sazany, "Deep Learning-Based Implementation of Hate Speech Identification on Texts in Indonesian: Preliminary Study," in *2018 International Conference on Applied Information Technology and Innovation (ICAITI) Deep*, IEEE, 2018, pp. 114–117.
- [82] L. H. Son, A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar, and M. Abdel-Basset, "Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network," *IEEE Access*, vol. 7, pp. 23319–23328, 2019, doi: 10.1109/ACCESS.2019.2899260.
- [83] R. L. Coste, "Fighting speech with speech: David Duke, the anti-defamation league, online bookstores, and hate filters," in *Proceedings of the Hawaii International Conference on System Sciences*, 2000, p. 72.
- [84] K. Gelber, "Terrorist-Extremist Speech and Hate Speech: Understanding the Similarities and Differences," *Ethical Theory and Moral Practice*, vol. 22, no. 3, pp. 607–622, 2019, doi: 10.1007/s10677-019-10013-x.

- [85] Z. Zhang and L. Luo, "Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter," *Semant Web*, vol. Accepted, 2018, doi: 10.3233/SW-180338.
- [86] F. Hara, "Adding emotional factors to synthesized voices," in *Robot and Human Communication - Proceedings of the IEEE International Workshop*, 1997, pp. 344–351.
- [87] N. R. Fatahillah, P. Suryati, and C. Haryawan, "Implementation of Naive Bayes classifier algorithm on social media (Twitter) to the teaching of Indonesian hate speech," in *Proceedings - 2017 International Conference on Sustainable Information Engineering and Technology, SIET 2017*, 2018, pp. 128–131. doi: 10.1109/SIET.2017.8304122.
- [88] E. Ombui, M. Karani, and L. Muchemi, "Annotation Framework for Hate Speech Identification in Tweets: Case Study of Tweets During Kenyan Elections," in *2019 IST-Africa Week Conference (IST-Africa)*, IST-Africa Institute and Authors, 2019, pp. 1–9.
- [89] I. M. Ahmad Niam, B. Irawan, C. Setianingsih, and B. P. Putra, "Hate Speech Detection Using Latent Semantic Analysis (LSA) Method Based on Image," in *Proceedings - 2018 International Conference on Control, Electronics, Renewable Energy and Communications, ICCEREC 2018*, IEEE, 2019, pp. 166–171. doi: 10.1109/ICCEREC.2018.8712111.
- [90] G. K. Pitsilis, H. Ramampiaro, and H. Langseth, "Detecting Offensive Language in Tweets Using Deep Learning," *arXiv:1801.04433v1*, pp. 1–17, 2018, doi: 10.1007/s10489-018-1242-y.
- [91] G. Pitsilis, H. Ramampiaro, and H. Langseth, "Effective hate-speech detection in Twitter data using recurrent neural networks," *Applied Intelligence*, vol. 48, p. in press., 2018, doi: 10.1007/s10489-018-1242-y.
- [92] W. Warner and J. Hirschberg, "Detecting Hate Speech on the World Wide Web," in *Proceedings of the Second Workshop on Language in Social Media*, S. O. Sood, M. Nagarajan, and M. Gamon, Eds., Montréal, Canada: Association for Computational Linguistics, Jun. 2012, pp. 19–26. [Online]. Available: <https://aclanthology.org/W12-2103>
- [93] C. Nobata and J. Tetreault, "Abusive Language Detection in Online User Content," in *International World Wide Web Conference*, 2016, pp. 145–153.
- [94] K. Dinakar, B. Jones, C. Havasi, and H. Lieberman, "Common Sense Reasoning for Detection, Prevention, and Mitigation of Cyberbullying," *ACM Trans Interact Intell Syst*, vol. 2, no. 3, p. 30, 2012, doi: 10.1145/2362394.2362400.

- [95] P. Burnap and M. L. Williams, "Cyber Hate Speech on Twitter : An Application of Machine Classification and Statistical Modeling for Policy and Decision Making," *Wiley Periodicals*, vol. 9999, no. 9999, 2015.
- [96] O. de Gibert Bonet, N. Perez, A. García-Pablos, and M. Cuadros, "Hate Speech Dataset from a White Supremacy Forum," 2018, pp. 11–20. doi: 10.18653/v1/W18-5102.
- [97] A. Jha, "When does a Compliment become Sexist ? Analysis and Classification of Ambivalent Sexism using Twitter Data," in *Proceedings of the Second Workshop on Natural Language Processing*, 2017, pp. 7–16.
- [98] H. Hosseinmardi, S. A. Mattson, R. I. Rafiq, R. Han, Q. Lv, and S. Mishra, "Detection of Cyberbullying Incidents on the Instagram Social Network," in *arXiv:1503.03909v1 [cs.SI] 12 Mar 2015 Abstract*, 2015.
- [99] F. Miro-Llinares and J. J. Rodriguez-Sala, "Cyber hate speech on twitter: Analyzing disruptive events from social media to build a violent communication and hate speech taxonomy," in *International Journal of Design and Nature and Ecodynamics*, 2016, pp. 406–415. doi: 10.2495/DNE-V11-N3-406-415.
- [100] G. Wang, B. Wang, T. Wang, A. Nika, H. Zheng, and B. Y. Zhao, "Whispers in the dark: Analysis of an anonymous social network," in *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, 2014, pp. 137–149. doi: 10.1145/2663716.2663728.
- [101] B. Mathew, P. Saha, S. M. Yimam, C. Biemann, P. Goyal, and A. Mukherjee, "HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection," in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2020, p. 12. [Online]. Available: <http://arxiv.org/abs/2012.10289>
- [102] E. Greevy and A. F. Smeaton, "Classifying Racist Texts Using A Support Vector Machine," in *ACM proceeding*, 2004, pp. 468–469.
- [103] G. H. Paetzold, S. Malmasi, and M. Zampieri, "UTFPR at SemEval-2019 Task 5: Hate Speech Identification with Recurrent Neural Networks," in *arXiv:1904.07839v1*, 2019, p. 5. [Online]. Available: <http://arxiv.org/abs/1904.07839>
- [104] N. Albadi, M. Kurdi, and S. Mishra, "Are they our brothers? analysis and detection of religious hate speech in the Arabic Twittersphere," *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018*, pp. 69–76, 2018, doi: 10.1109/ASONAM.2018.8508247.

- [105] A. S. Adekotujo, J. Y. Lee, A. O. Enikuomelin, M. Mazzara, and S. B. Aribisala, *Bi-lingual Intent Classification of Twitter Posts: A Roadmap*, vol. 925. Springer International Publishing, 2020. doi: 10.1007/978-3-030-14687-0\_1.
- [106] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, “Deep Learning for Hate Speech Detection in Tweets,” in *Proceedings of the 26th International Conference on World Wide Web Companion*, in WWW ’17 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, pp. 759–760. doi: 10.1145/3041021.3054223.
- [107] W. Alorainy, P. Burnap, H. A. N. Liu, and M. L. Williams, ““ The Enemy Among Us ’: Detecting Cyber Hate Speech with Threats-based Othering Language Embeddings,” *ACM Transactions on the Web*, vol. 13, no. 3, 2019.
- [108] S. Modha, P. Majumder, T. Mandl, and C. Mandalia, “Detecting and visualizing hate speech in social media: A cyber Watchdog for surveillance,” *Expert Syst Appl*, vol. 161, p. 113725, 2020, doi: 10.1016/j.eswa.2020.113725.
- [109] T. Wullach, A. Adler, and E. M. Minkov, “Towards Hate Speech Detection at Large via Deep Generative Modeling,” *IEEE Internet Comput*, 2020, doi: 10.1109/MIC.2020.3033161.
- [110] M. Behzadi, I. G. Harris, and A. Derakhshan, “Rapid Cyber-bullying detection method using Compact BERT Models,” *Proceedings - 2021 IEEE 15th International Conference on Semantic Computing, ICSC 2021*, pp. 199–202, 2021, doi: 10.1109/ICSC50631.2021.00042.
- [111] F. M. Plaza-del-Arco, M. D. Molina-González, L. A. Ureña-López, and M. T. Martín-Valdivia, “Comparing pre-trained language models for Spanish hate speech detection,” *Expert Systems with Applications*, vol. 166. 2021. doi: 10.1016/j.eswa.2020.114120.
- [112] M. S. Jahan and M. Oussalah, “A systematic review of hate speech automatic detection using natural language processing,” *Neurocomputing*, vol. 546. Elsevier B.V., Aug. 14, 2023. doi: 10.1016/j.neucom.2023.126232.
- [113] G. K. Pitsilis, H. Ramampiaro, and H. Langseth, “Effective hate-speech detection in Twitter data using recurrent neural networks,” *Applied Intelligence*, vol. 48, no. 12, pp. 4730–4742, Dec. 2018, doi: 10.1007/s10489-018-1242-y.
- [114] W. Yin and A. Zubiaga, “Towards generalisable hate speech detection: a review on obstacles and solutions,” *PeerJ Comput Sci*, vol. 7, pp. 1–38, 2021, doi: 10.7717/PEERJ-CS.598.

- [115] F. E. Ayo, O. Folorunso, F. T. Ibharalu, I. A. Osinuga, and A. Abayomi-Alli, “A probabilistic clustering model for hate speech classification in twitter,” *Expert Syst Appl*, vol. 173, Jul. 2021, doi: 10.1016/j.eswa.2021.114762.
- [116] N. Chetty and S. Alathur, “Hate speech review in the context of online social networks,” *Aggression and Violent Behavior*, vol. 40. Elsevier Ltd, pp. 108–118, May 01, 2018. doi: 10.1016/j.avb.2018.05.003.
- [117] A. Matamoros-Fernández and J. Farkas, “Racism, Hate Speech, and Social Media: A Systematic Review and Critique,” *Television and New Media*, vol. 22, no. 2, pp. 205–224, Feb. 2021, doi: 10.1177/1527476420982230.
- [118] F. Alkomah and X. Ma, “A Literature Review of Textual Hate Speech Detection Methods and Datasets,” *Information (Switzerland)*, vol. 13, no. 6. MDPI, Jun. 01, 2022. doi: 10.3390/info13060273.
- [119] S. MacAvaney, H. R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder, “Hate speech detection: Challenges and solutions,” *PLoS One*, vol. 14, no. 8, Aug. 2019, doi: 10.1371/journal.pone.0221152.
- [120] R. Cao, R. K. W. Lee, and T. A. Hoang, “DeepHate: Hate Speech Detection via Multi-Faceted Text Representations,” in *WebSci 2020 - Proceedings of the 12th ACM Conference on Web Science*, Association for Computing Machinery, Inc, Jul. 2020, pp. 11–20. doi: 10.1145/3394231.3397890.
- [121] P. K. Roy, A. K. Tripathy, T. K. Das, and X. Z. Gao, “A framework for hate speech detection using deep convolutional neural network,” *IEEE Access*, vol. 8, pp. 204951–204962, 2020, doi: 10.1109/ACCESS.2020.3037073.
- [122] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, “Hate speech detection with comment embeddings,” in *WWW 2015 Companion - Proceedings of the 24th International Conference on World Wide Web*, Association for Computing Machinery, Inc, May 2015, pp. 29–30. doi: 10.1145/2740908.2742760.
- [123] F. M. Plaza-Del-Arco, D. Nozza, and D. Hovy, “Respectful or Toxic? Using Zero-Shot Learning with Language Models to Detect Hate Speech,” 2023. [Online]. Available: <https://github.com/MilaNLPProc/>
- [124] K. Miok, B. Škrlj, D. Zaharie, and M. Robnik-Šikonja, “To BAN or Not to BAN: Bayesian Attention Networks for Reliable Hate Speech Detection,” *Cognit Comput*, vol. 14, no. 1, pp. 353–371, Jan. 2022, doi: 10.1007/s12559-021-09826-9.
- [125] M. Corazza, S. Menini, E. Cabrio, S. Tonelli, and S. Villata, “Hybrid Emoji-Based Masked Language Models for Zero-Shot Abusive Language Detection.”



- [126] A. Arango, J. Pérez, and B. Poblete, “Hate Speech Detection is Not as Easy as You May Think: A Closer Look at Model Validation.”
- [127] M. Mozafari, R. Farahbakhsh, and N. Crespi, “A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media,” Oct. 2019, [Online]. Available: <http://arxiv.org/abs/1910.12574>
- [128] D. Sultan *et al.*, “Cyberbullying-related Hate Speech Detection Using Shallow-to-deep Learning,” *Computers, Materials and Continua*, vol. 74, no. 1, pp. 2115–2131, 2023, doi: 10.32604/cmc.2023.032993.
- [129] H. Saleh, A. Alhothali, and K. Moria, “Detection of Hate Speech using BERT and Hate Speech Word Embedding with Deep Model,” *Applied Artificial Intelligence*, vol. 37, no. 1, 2023, doi: 10.1080/08839514.2023.2166719.
- [130] J. P. Ferraro, H. D. Iii, S. L. Duvall, W. W. Chapman, H. Harkema, and P. J. Haug, “Improving performance of natural language processing part-of-speech tagging on clinical narratives through domain adaptation,” pp. 931–939, 2013, doi: 10.1136/amiajnl-2012-001453.
- [131] S. Ghosal and A. Jain, “HateCircle and Unsupervised Hate Speech Detection Incorporating Emotion and Contextual Semantics,” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 4, Mar. 2023, doi: 10.1145/3576913.
- [132] A. C. Mazari, N. Boudoukhani, and A. Djeflal, “BERT-based ensemble learning for multi-aspect hate speech detection,” *Cluster Comput*, vol. 27, no. 1, pp. 325–339, Feb. 2024, doi: 10.1007/s10586-022-03956-x.
- [133] I. Bigoulaeva, V. Hangya, I. Gurevych, and A. Fraser, “Label modification and bootstrapping for zero-shot cross-lingual hate speech detection,” *Lang Resour Eval*, vol. 57, no. 4, pp. 1515–1546, Dec. 2023, doi: 10.1007/s10579-023-09637-4.
- [134] S. Khan *et al.*, “HCovBi-Caps: Hate Speech Detection Using Convolutional and Bi-Directional Gated Recurrent Unit With Capsule Network,” *IEEE Access*, vol. 10, pp. 7881–7894, 2022, doi: 10.1109/ACCESS.2022.3143799.
- [135] W. Warner and J. Hirschberg, “Detecting Hate Speech on the World Wide Web,” 2012. [Online]. Available: <http://info.yahoo.com/legal/us/yahoo/utos/utos-173.html>
- [136] S. Khan *et al.*, “BiCHAT: BiLSTM with deep CNN and hierarchical attention for hate speech detection,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 4335–4344, Jul. 2022, doi: 10.1016/j.jksuci.2022.05.006.

- [137] J. Qian, A. Bethke, Y. Liu, E. Belding, and W. Y. Wang, “A benchmark dataset for learning to intervene in online hate speech,” *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp. 4755–4764, 2020, doi: 10.18653/v1/d19-1482.
- [138] D. Chicco and G. Jurman, “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation,” *BMC Genomics*, vol. 21, no. 1, pp. 1–13, 2020, doi: 10.1186/s12864-019-6413-7.
- [139] J. C. Pereira-Kohatsu, L. Quijano-Sánchez, F. Liberatore, and M. Camacho-Collados, “Detecting and monitoring hate speech in twitter,” *Sensors (Switzerland)*, vol. 19, no. 21, pp. 1–37, 2019, doi: 10.3390/s19214654.
- [140] P. Zhou, K. Wang, L. Guo, S. Gong, and B. Zheng, “A Privacy-Preserving Distributed Contextual Federated Online Learning Framework with Big Data Support in Social Recommender Systems,” *IEEE Trans Knowl Data Eng*, vol. 33, no. 3, pp. 824–838, 2021, doi: 10.1109/TKDE.2019.2936565.
- [141] M. Ridenhour, A. Bagavathi, E. Raisi, and S. Krishnan, “Detecting Online Hate Speech: Approaches Using Weak Supervision and Network Embedding Models,” in *Social, Cultural, and Behavioral Modeling*, R. Thomson, H. Bisgin, C. Dancy, A. Hyder, and M. Hussain, Eds., Cham: Springer International Publishing, 2020, pp. 202–212.
- [142] A. Omar, T. M. Mahmoud, and T. Abd-El-Hafeez, “Comparative Performance of Machine Learning and Deep Learning Algorithms for Arabic Hate Speech Detection in OSNs,” *Advances in Intelligent Systems and Computing*, vol. 1153 AISC, pp. 247–257, 2020, doi: 10.1007/978-3-030-44289-7\_24.
- [143] R. Cao, R. K.-W. Lee, and T. Hoang, “DeepHate: Hate Speech Detection via Multi-Faceted Text Representations,” 2020, pp. 11–20. doi: 10.1145/3394231.3397890.
- [144] G. K. Pitsilis, H. Ramampiaro, and H. Langseth, “Effective hate-speech detection in Twitter data using recurrent neural networks,” *Applied Intelligence*, vol. 48, pp. 4730–4742, 2018, [Online]. Available: <https://api.semanticscholar.org/CorpusID:51694919>
- [145] A. Dixit, “Emotion Detection Using Decision Tree Technique,” 2017.
- [146] A. L. and M. Wiener, “Classification and Regression by randomForest. R News 2,” vol. 3, no. December 2002, pp. 18–22, 2003.

- [147] B. Trstenjak, S. Mikac, and D. Donko, “KNN with TF-IDF Based Framework for Text Categorization,” *Procedia Eng*, vol. 69, pp. 1356–1364, 2014, doi: 10.1016/j.proeng.2014.03.129.
- [148] S. K. Sien, “Adapting word2vec to Named Entity Recognition,” *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, no. Nodalida, pp. 239–243, 2015.
- [149] E. Altszyler, M. Sigman, and D. Fernández Slezak, “Corpus Specificity in LSA and Word2vec: The Role of Out-of-Domain Documents,” pp. 1–10, 2019, doi: 10.18653/v1/w18-3001.
- [150] C. F. Tsai, Z. Y. Chen, and S. W. Ke, “Evolutionary instance selection for text classification,” *Journal of Systems and Software*, vol. 90, no. 1, pp. 104–113, 2014, doi: 10.1016/j.jss.2013.12.034.
- [151] M. Bishnoi and P. Singh, “Modularizing Software Systems using PSO optimized hierarchical clustering,” *2016 International Conference on Computational Techniques in Information and Communication Technologies, ICCTICT 2016 - Proceedings*, pp. 659–664, 2016, doi: 10.1109/ICCTICT.2016.7514660.
- [152] B. Xu, X. Xie, L. Shi, and C. Nie, “Application of genetic algorithms in software testing,” *Advances in Machine Learning Applications in Software Engineering*, vol. 3, no. 4, pp. 287–317, 2006, doi: 10.4018/978-1-59140-941-1.ch012.
- [153] I. Repository, “Description logic-based knowledge merging for concrete- and fuzzy- domain ontologies,” 2015, doi: 10.1177/0954405414564404.Additional.
- [154] T. T. Quan, S. C. Hui, T. H. Cao, T. T. Quan, S. C. Hui, and T. H. Cao, “A Fuzzy Fuzzy FCA-based FCA-based Approach Approach to to Conceptual Conceptual Clustering for Automatic Generation Clustering for Automatic Generation of of Concept Concept Hierarchy Hierarchy on on Uncertainty Uncertainty Data Data,” pp. 1–12, 2004.
- [155] T. Davidson, D. Warmsley, M. W. Macy, and I. Weber, “Automated Hate Speech Detection and the Problem of Offensive Language,” in *International Conference on Web and Social Media*, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:1733167>
- [156] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE : Synthetic Minority Over-sampling Technique,” vol. 16, pp. 321–357, 2002.
- [157] T. Zhang and X. Yang, “G-SMOTE: A GMM-based synthetic minority oversampling technique for imbalanced learning,” no. 20170540097, 2018, [Online]. Available: <http://arxiv.org/abs/1810.10363>

- [158] M. Corazza, S. Menini, E. Cabrio, S. Tonelli, and S. Villata, “A Multilingual Evaluation for Online Hate Speech Detection,” *ACM Trans Internet Technol*, vol. 20, no. 2, 2020, doi: 10.1145/3377323.
- [159] N. D. Ousidhoum, Z. Lin, H. Zhang, Y. Song, and D. Y. Yeung, “Multilingual and Multi-Aspect Hate Speech Analysis,” *ArXiv*, vol. abs/1908.1, 2019, [Online]. Available: <https://api.semanticscholar.org/CorpusID:201669180>
- [160] T. Mandl *et al.*, “Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification in Indo-European languages,” *ACM International Conference Proceeding Series*, pp. 14–17, 2019, doi: 10.1145/3368567.3368584.
- [161] M. Mozafari, R. Farahbakhsh, and N. Crespi, “A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media,” *Studies in Computational Intelligence*, vol. 881 SCI, pp. 928–940, 2020, doi: 10.1007/978-3-030-36687-2\_77.
- [162] F. Morstatter, L. Wu, U. Yavanoglu, S. R. Corman, and H. Liu, “Identifying Framing Bias in Online News,” *ACM Transactions on Social Computing*, vol. 1, no. 2, pp. 1–18, 2018, doi: 10.1145/3204948.
- [163] A. Bohra, D. Vijay, V. Singh, S. S. Akhtar, and M. Shrivastava, “A Dataset of Hindi-English Code-Mixed Social Media Text for Hate Speech Detection,” 2018, pp. 36–41. doi: 10.18653/v1/W18-1105.
- [164] A. Omar, T. M. Mahmoud, and T. Abd-El-Hafeez, “Comparative Performance of Machine Learning and Deep Learning Algorithms for Arabic Hate Speech Detection in OSNs,” *Advances in Intelligent Systems and Computing*, vol. 1153 AISC, pp. 247–257, 2020, doi: 10.1007/978-3-030-44289-7\_24.
- [165] Y. Zhou, Y. Yang, H. Liu, X. Liu, and N. Savage, “Deep Learning Based Fusion Approach for Hate Speech Detection,” *IEEE Access*, vol. 8, pp. 128923–128929, 2020, doi: 10.1109/ACCESS.2020.3009244.
- [166] A. Chaudhari, A. Parseja, and A. Patyal, “CNN based Hate-o-Meter: A Hate Speech Detecting Tool,” in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2020, pp. 940–944. doi: 10.1109/ICSSIT48917.2020.9214247.
- [167] À. A. Carracedo and R. J. Mondéjar, “Profiling Hate Speech Spreaders on Twitter,” *CEUR Workshop Proc*, vol. 2936, no. September, pp. 1801–1807, 2021.
- [168] P. Mathur, R. R. Shah, R. Sawhney, and D. Mahata, “Detecting Offensive Tweets in Hindi-English Code-Switched Language,” *Proceedings of the Annual*

- Meeting of the Association for Computational Linguistics*, pp. 18–26, 2018, doi: 10.18653/v1/w18-3504.
- [169] A. C. Mazari, N. Boudoukhani, and A. Djeflal, “BERT-based ensemble learning for multi-aspect hate speech detection,” *Cluster Comput*, vol. 0123456789, 2023, doi: 10.1007/s10586-022-03956-x.
- [170] R. Ali, U. Farooq, U. Arshad, W. Shahzad, and M. O. Beg, “Hate speech detection on Twitter using transfer learning,” *Comput Speech Lang*, vol. 74, p. 101365, 2022, doi: <https://doi.org/10.1016/j.csl.2022.101365>.
- [171] P. M. Hiren Madhu, Shrey Satapara, Sandip Modha, Thomas Mandl, “SentBERT,” *Expert Syst Appl*, 2023.
- [172] A. Ryzhova, D. Devyatkin, S. Volkov, and V. Budzko, “Training Multilingual and Adversarial Attack-Robust Models for Hate Detection on Social Media,” *Procedia Comput Sci*, vol. 213, pp. 196–202, 2022, doi: <https://doi.org/10.1016/j.procs.2022.11.056>.
- [173] M. Bojkovský and M. Pikuliak, “STUFIT at SemEval-2019 task 5: Multilingual hate speech detection on Twitter with MUSE and ELMo embeddings,” *NAACL HLT 2019 - International Workshop on Semantic Evaluation, SemEval 2019, Proceedings of the 13th Workshop*, pp. 464–468, 2019, doi: 10.18653/v1/s19-2082.
- [174] J. Cokley, “The Reuters Institute’s Digital News Report 2012,” *Digital Journalism*, vol. 1, no. 2, pp. 286–287, Jun. 2013, doi: 10.1080/21670811.2012.744561.
- [175] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, “Hate speech detection with comment embeddings,” in *WWW 2015 Companion - Proceedings of the 24th International Conference on World Wide Web*, Association for Computing Machinery, Inc, May 2015, pp. 29–30. doi: 10.1145/2740908.2742760.
- [176] Z. Al-Makhadmeh and A. Tolba, “Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach,” *Computing*, vol. 102, no. 2, pp. 501–522, Feb. 2020, doi: 10.1007/s00607-019-00745-0.
- [177] A. Al-Hassan and H. Al-Dossari, “Detection of hate speech in Arabic tweets using deep learning,” in *Multimedia Systems*, Springer Science and Business Media Deutschland GmbH, Dec. 2022, pp. 1963–1974. doi: 10.1007/s00530-020-00742-w.

- [178] K. Sreelakshmi, B. Premjith, and K. P. Soman, "Detection of Hate Speech Text in Hindi-English Code-mixed Data," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 737–744. doi: 10.1016/j.procs.2020.04.080.
- [179] I. Alfina, R. Mulia, M. I. Fanany, and Y. Ekanata, "Hate speech detection in the Indonesian language: A dataset and preliminary study," in *2017 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2017*, Institute of Electrical and Electronics Engineers Inc., Jul. 2017, pp. 233–237. doi: 10.1109/ICACSIS.2017.8355039.
- [180] D. Sharma, V. K. Singh, V. Gupta, J. Global, B. School, and O. P. Jindal, "TABHATE: A Target-based Hate Speech Detection Dataset in Hindi," 2023, doi: 10.21203/rs.3.rs-2800717/v1.
- [181] H. Bonaldi, S. Dellantonio, S. S. Tekiroglu, and M. Guerini, "Human-Machine Collaboration Approaches to Build a Dialogue Dataset for Hate Speech Countering," Nov. 2022, [Online]. Available: <http://arxiv.org/abs/2211.03433>
- [182] Y.-L. Chung, M. G. Fondazione, B. Kessler, and V. Patti, "ICT International Doctoral School Counter Narrative Generation for Fighting Online Hate Speech."
- [183] M. Schütz *et al.*, "DeTox at GermEval 2021: Toxic Comment Classification." [Online]. Available: <https://pypi.org/project/emosent-py/>
- [184] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic keyword extraction from individual documents," 2010.
- [185] E. S. Summarization, A. Barrera, and R. Verma, "Combining Syntax and Semantics for Automatic," vol. 1, pp. 366–377, 2001.
- [186] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic Keyword Extraction from Individual Documents," *Text Mining: Applications and Theory*, no. March, pp. 1–20, 2010, doi: 10.1002/9780470689646.ch1.
- [187] A. Kaur and S. Goyal, "Text analytics based severity prediction of software bugs for apache projects," *International Journal of System Assurance Engineering and Management*, vol. 10, 2019, doi: 10.1007/s13198-019-00807-8.
- [188] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Oct. 2018, [Online]. Available: <http://arxiv.org/abs/1810.04805>
- [189] Y. Zhuang, H. Gao, F. Wu, S. Tang, Y. Zhang, and Z. Zhang, "Probabilistic Word Selection via Topic Modeling," vol. 27, no. 6, pp. 1643–1655, 2015.

- [190] J. Willems, “Comparison of Data Preprocessing Techniques on Software Sources for Topic Modeling,” 2014.
- [191] I. Vogel and M. Meghana, “Profiling Hate Speech Spreaders on Twitter: SVM vs. Bi-LSTM,” 2021. [Online]. Available: <https://dictionary.cambridge.org/de/worterbuch/englisch/hate-speech>
- [192] A.-M. Founta *et al.*, “Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior,” 2018. [Online]. Available: [www.aaai.org](http://www.aaai.org)
- [193] Y. Zhao, “Text Mining with R – Twitter Data Analysis 1 Introduction,” vol. 2014, no. May, pp. 1–34, 2015.
- [194] T. Adewumi, S. S. Sabry, N. Abid, F. Liwicki, and M. Liwicki, “T5 for Hate Speech, Augmented Data and Ensemble,” Oct. 2022, [Online]. Available: <http://arxiv.org/abs/2210.05480>
- [195] L. Han, T. Finin, P. Mcnamee, A. Joshi, Y. Yesha, and I. C. Society, “Improving Word Similarity by Augmenting PMI with Estimates of Word Polysemy,” vol. 25, no. 6, pp. 1307–1322, 2013.
- [196] P. William, R. Gade, R. esh Chaudhari, A. B. Pawar, and M. A. Jawale, “Machine Learning based Automatic Hate Speech Recognition System,” in *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, 2022, pp. 315–318. doi: 10.1109/ICSCDS53736.2022.9760959.
- [197] O. Araque and C. A. Iglesias, “An Ensemble Method for Radicalization and Hate Speech Detection Online Empowered by Sentic Computing,” *Cognit Comput*, no. 0123456789, 2021, doi: 10.1007/s12559-021-09845-6.
- [198] S. Malmasi and M. Zampieri, “Challenges in Discriminating Profanity from Hate Speech,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 30, pp. 1–16, 2017, doi: 10.1080/0952813X.2017.1409284.
- [199] A. Matamoros-Fernández and J. Farkas, “Racism, Hate Speech, and Social Media: A Systematic Review and Critique,” *Television and New Media*, vol. 22, no. 2, pp. 205–224, 2021, doi: 10.1177/1527476420982230.
- [200] S. Dowlagar and R. Mamidi, “HASOCOne@FIRE-HASOC2020: Using BERT and Multilingual BERT models for Hate Speech Detection.” 2021.
- [201] G. Rizos, K. Hemker, and B. Schuller, “Augment to Prevent: Short-Text Data Augmentation in Deep Learning for Hate-Speech Classification,” 2019, pp. 991–1000. doi: 10.1145/3357384.3358040.

- [202] H. Saleh, A. Alhothali, and K. Moria, "Detection of Hate Speech using BERT and Hate Speech Word Embedding with Deep Model," *Applied Artificial Intelligence*, vol. 37, no. 1, 2023, doi: 10.1080/08839514.2023.2166719.
- [203] O. de Gibert, N. Perez, A. García-Pablos, and M. Cuadros, "Hate Speech Dataset from a White Supremacy Forum," Sep. 2018, [Online]. Available: <http://arxiv.org/abs/1809.04444>
- [204] F. M. Plaza-del-Arco, M. D. Molina-González, L. A. Ureña-López, and M. T. Martín-Valdivia, "Comparing pre-trained language models for Spanish hate speech detection," *Expert Syst Appl*, vol. 166, Mar. 2021, doi: 10.1016/j.eswa.2020.114120.
- [205] "Hate Speech detection in the Bengali language: A dataset and its baseline evaluation." [Online]. Available: <https://github.com/strohne/Facepager>
- [206] K. Miok, B. Škrlj, D. Zaharie, and M. Robnik-Šikonja, "To BAN or Not to BAN: Bayesian Attention Networks for Reliable Hate Speech Detection," *Cognit Comput*, vol. 14, no. 1, pp. 353–371, Jan. 2022, doi: 10.1007/s12559-021-09826-9.
- [207] Z. Al-Makhadmeh and A. Tolba, "Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach," *Computing*, vol. 102, no. 2, pp. 501–522, Feb. 2020, doi: 10.1007/s00607-019-00745-0.
- [208] L. Yuan, T. Wang, G. Ferraro, H. Suominen, and M. A. Rizoio, "Transfer learning for hate speech detection in social media," *J Comput Soc Sci*, vol. 6, no. 2, pp. 1081–1101, Oct. 2023, doi: 10.1007/s42001-023-00224-9.
- [209] S. S. Sabry, T. Adewumi, N. Abid, G. Kovacs, F. Liwicki, and M. Liwicki, "HaT5: Hate Language Identification using Text-to-Text Transfer Transformer," Feb. 2022, [Online]. Available: <http://arxiv.org/abs/2202.05690>
- [210] A. Kaur and S. G. Jindal, "Text analytics based severity prediction of software bugs for apache projects," *International Journal of Systems Assurance Engineering and Management*, vol. 10, no. 4, pp. 765–782, 2019, doi: 10.1007/s13198-019-00807-8.
- [211] C. Raffel *et al.*, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," 2020. [Online]. Available: <http://jmlr.org/papers/v21/20-074.html>.
- [212] M. Tahmid, R. Laskar, E. Hoque, and J. Huang, "Query Focused Abstractive Summarization via Incorporating Query Relevance and Transfer Learning with



- Transformer Models.” [Online]. Available: <https://github.com/tahmedge/QR-BERTSUM-TL-for-QFAS>
- [213] C. Raffel *et al.*, “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.” 2023.
- [214] T. Davidson, D. Warmsley, M. Macy, and I. Weber, “Automated Hate Speech Detection and the Problem of Offensive Language,” Mar. 2017, [Online]. Available: <http://arxiv.org/abs/1703.04009>
- [215] K. Sreelakshmi, B. Premjith, and K. P. Soman, “Detection of Hate Speech Text in Hindi-English Code-mixed Data,” in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 737–744. doi: 10.1016/j.procs.2020.04.080.
- [216] T. Davidson, D. Warmsley, M. Macy, and I. Weber, “Automated Hate Speech Detection and the Problem of Offensive Language,” 2017. [Online]. Available: [www.aaai.org](http://www.aaai.org)
- [217] T. Davidson, D. Warmsley, M. Macy, and I. Weber, “Automated Hate Speech Detection and the Problem of Offensive Language,” Mar. 2017, [Online]. Available: <http://arxiv.org/abs/1703.04009>
- [218] Z. Waseem, “Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter,” 2016. [Online]. Available: [www.spacy.io](http://www.spacy.io)