CROSS-PLATFORM ANALYSIS OF HATE SPEECH DETECTION

A MAJOR PROJECT-II REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
INFORMATION TECHNOLOFY

Submitted by:
JAYBARDHAN KUMAR
2K23/ITY/09

Under the supervision of
PROF. DINESH KUMAR VISHWAKARMA

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

May, 2025

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Jaybardhan kumar, 2K23/ITY/09 student of M.Tech in Information Technology, hereby declare that the Major Project-II dissertation titled "**CROSS-PLATFORM ANALYSIS OF HATE SPEECH DETECTION**" which is submitted by me to the Department of Information Technology, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any degree, Diploma Associateship, Fellowship, or other similar title or recognition.

Place: Delhi	**JAYBARDHAN KUMAR**
Date:	**2K23/ITY/09**

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Major Project-II dissertation titled "**CROSS-PLATFORM ANALYSIS OF HATE SPEECH DETECTION**" which is submitted by Jaybardhan Kumar, 2K23/ITY/09, Department of Information Technology, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge, this work has not been submitted in part or full for any degree or diploma to this University or elsewhere.

Signature
Prof. Dinesh Kumar Vishwakarma
SUPERVISOR
Head of Department
Department of Information Technology
Delhi Technological University

Place: New Delhi
Date:

ABSTRACT

In the era of technology, social media has become flag bearer of freedom of speech and expression. However, in the guise of freedom of speech and expression hate and offensive speech is increasing day by day, posing significant risks to societal harmony. Hate speech detection on social media has become increasingly important due to the rise of online platforms and their potential to amplify harmful content [1], [2]. While traditional text-based hate speech detection is well-researched, the unique challenges of spoken language transcription, particularly on platforms like YouTube, require specialized approaches. This study investigates the overall effectiveness of hate speech detection models trained on datasets from Facebook and Twitter when applied to the distinct context of YouTube transcriptions. Various machine learning models are explored, comparing the performance of traditional classifiers like Naive Bayes, Support Vector Machines (SVM), Logistic Regression, and Random Forest – using TF-IDF features to assess their ability to generalize with the complexities of YouTube transcriptions. We evaluate each model's performance across accuracy, precision, recall, and F1-score to determine their effectiveness in capturing both explicit and implicit hate speech. Findings reveal that models trained on datasets from Facebook and Twitter struggle to generalize effectively to the more nuanced and context-rich environment of YouTube transcriptions, Support Vector Machines and Logistic Regression show relatively better adaptability. This work highlights the importance of contextual and linguistic adaptability in hate speech detection on multimedia platforms and discusses implications for ethical content moderation and policy development. This study underscores the need for continued research into models that address platform- specific language, cultural nuances, and code-mixing, particularly in low-resource languages. These findings provide a foundation for researchers and practitioners seeking to develop or refine hate speech detection systems for real-world application.

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I would like to express my sincere gratitude towards my supervisors Prof. Dinesh Kumar Vishwakarma for providing her invaluable guidance, comments, and suggestions throughout the course of the project.

The results of this thesis would not have been possible without support from all who directly or indirectly, have lent their hand throughout the course of the project. I would like to thank my parents and faculties of the Department of Information Technology, Delhi Technological University, for their kind cooperation and encouragement which helped me complete this thesis. I hope that this project will serve its purpose to the fullest extent possible.

JAYBARDHAN KUMAR
2K23/ITY/09

TABLE OF CONTENTS

Candidate's Declaration ii

Certificate iii

Abstract iv

Acknowledgment v

Table of Contents vi

List of Figures vii

List of Tables viii

List of Abbreviations ix

CHAPTER 1 INTRODUCTION 10 - 11

CHAPTER 2 LITERATURE REVIEW 12 - 15

CHAPTER 3 METHODOLOGY 16 - 24

3.1 Dataset Used 16

3.1.1 Training Data 16

3.1.2 Testing Data 16

3.2 Classifiers Used 17

3.2.1 Random Forest 17

3.2.2 Support Vector Machine (SVM) 18

3.2.3 Naïve Bayes 19

3.2.4 Logistic Regression 20

3.2.5 TF-IDF 20

3.3 Work Flow 21

3.3.1 Text Preprocessing 21

3.3.2 Feature Extraction 22

3.3.3 Model Training and Validation 22

3.3.4 Model Testing 23

3.3.5 Evaluation Matrix's 23

3.4 Ensemble Classification Framework 23

CHAPTER 4 RESULT AND DISCUSSION 25 - 28

4.1 Comparison with Existing State-of-the-Art Models 25

4.2 Experimental Results of Ensemble Model 26

CHAPTER 5 CONCLUSION AND FUTURE SCOPE 29 - 30

REFERENCES 31

LIST OF FIGURES

3.1 Random Forest Approach 18
3.2 Classification of data points using SVM 19
3.3 Work Flow chart 21
3.4 Text Preprocessing 22
3.5 Model Evaluation Matrix 23
4.1 Model Performance Comparison on YouTube Dataset 27

LIST OF TABLES

2.1 Table for Review of literature and their key contributions 15
4.1 Table for Comparison with Existing State-of-the-Art Model 25
4.2 Table for Model Evaluation Matrix 26

LIST OF ABBREVIATIONS

LR	Logistic Regression
DP	Deep learning
RF	Random Forest
NB	Naive Bayes
AUC	Area Under Curve
TF-IDF	Term Frequency-Inverse Document Frequency
CNN	Convolutional Neural Network
LSTM	Long short-term memory
SVM	Support Vector Machine
ML	Machine learning
BERT	Bidirectional encoder representations from transformers
NLP	Natural Language Processing
AC	Accuracy
PR	Precision
RC	Recall

CHAPTER 1

INTRODUCTION

Hate speech is any form communication that advocates attack, fear, violence, abuse, defemination, assault on individual or group of individuals based on their sexual orientation, gender, age, handicap, race, ethnicity, country of origin, caste, religion, or serious illness [1]. Hate speeches are of classified into numerous categories such as hate, non-hate, abusive, offensive and non- offensive etc. [1] In this study hate speeches are categorized into two primary types: hate speech, involving statements intended to harm or target individuals or groups, and non-hate speech, referring to neutral or non-targeted communication. The internet is a wonderful place to share one's thoughts, knowledge and experiences. Unfortunately, it sometimes also becomes a place where deplorable and despicable things are said to a person or targeted at a group of people [1], [2]. Increase in internet use as well as social media platforms led to surge in hate speech which has broken geographical barriers because social media networks offer rapid communication with messages transmitted instantly [2]. On the contrary, when social media platforms were not in picture hate speech given at one place of world doesn't disseminate vigorously to other places of world. Since hate speeches often incite violence and heinous crimes Therefore, detecting hate speech has become a pressing necessity.

Platforms like YouTube, with millions of daily active users and high levels of engagement, are particularly susceptible to such content, often shared under the guise of personal opinion, political discourse, or satire. Unlike traditional text-based platforms such as Twitter, YouTube's content is largely driven by spoken language, which adds complexity to hate speech detection. Transcriptions of YouTube videos, often auto-generated, present unique linguistic challenges—such as informal language, slang, colloquial expressions, code-mixed language, and contextually nuanced language specific to the cultural and political topics discussed and code-mixing— demanding refined approaches to hate speech detection.

This study focuses to address these challenges by reckoning the efficacy of various ML approaches for hate speech detection in YouTube transcriptions. Specifically, we examine traditional models (NB, SVM, LR, RF) assessing their performance in detecting hate speech across transcription from YouTube cannels in categories such as political news and satire. Given the context-heavy nature of YouTube transcription text, we employ a domain adaptation strategy, first pre-training on general hate speech datasets before fine-tuning on a custom-labelled YouTube-specific dataset. This approach allows us to adapt models to the unique characteristics of spoken language on YouTube while maintaining their ability to generalize to varied forms of hate speech.

Through evaluating models systematically based on AC, PR, RC, and F1-score, we want to find the most appropriate approaches for hate speech detection on YouTube transcriptions. Our results will offer valuable insights into the adaptability of current hate speech detection models within a multimedia setup and contribute to the broader rubric of ethical content moderation and policy development on online platforms. SVM is a supervised learning algorithm majorly employed for classification and regression tasks. It functions by choosing the best hyperplane that separates data points into classes in the higher dimension of the feature space. It is one of the widely-used classification algorithms, well-suited for tasks like hate speech detection where the textual data needs conversion into separate categories, such as hate vs. non-hate. Logistic regression is a simple, easy, and fast statistical method, used mostly for binary text classification. Also, less prone to overfitting and provides a good baseline method to compare more complicated models on overhead. Random forest also fulfils the purpose of ensemble learning, used mainly for classification tasks, such as hate speech detection. Having the capability to identify more complex or non-linear relationships, it excels where simpler algorithms cannot, offering higher adaptability to various language patterns present on social media text. Naive Bayes is based upon Bayes' theorem and is applied for-text classification tasks like hate speech detection which acts as solid baseline [1], [2]. It applies Bayes' theorem to reckon the probability that a given piece of text belongs to a specific class (e.g., hate speech or non-hate speech). TF-IDF a widely used technique for feature extraction in text mining and NLP, and it helps identify which words are important in a document while discounting frequently occurring, less informative words [2], [3], [4].

CHAPTER 2

LITERATURE REVIEW

After going through various research papers, we found various kinds of approaches and we can categorize them into the following categories on the basis of the features that they used for detection

Detection of hate speech has received significant amounts of research interest in recent times, especially because of the development of social media sites like Twitter and Facebook, which allow fast spreading of content generated by users [1], [4], [28]. Earlier studies mostly opted for shallow ML algorithms such as Logistic Regression (LR), Naive Bayes (NB), Support Vector Machines (SVM), and Random Forests (RF) because of their advantages of simplicity, interpretability, as well as their ability to handle structured features like term frequency-inverse document frequency (TF-IDF) and n-grams [1], [2], [3], [12], [13].

These classical approaches were well suited for the binary and multi-class classification tasks for datasets that contained structured short-form texts from Twitter and Facebook [2], [3], [14]. With the blessing termed advancement of deep learning, the community has been therefore led towards the world of deep neural networks, wherein by going deeper, one finds CNNs; discretely shallow, one finds LSTM models; and the ever-presto, Transformer-based models, including BERT, RoBERTa, and their variants [4], [18], [28]. These architectures outperform the traditional ones due to their ability to capture rich contextual semantic and sequential dependencies, especially with a longer text or a code-mixed one [16], [17], [20]. Nonetheless, for deploying these models into the real world, there are still challenges. Fortuna and Nunes [4] stress non-standardization in terms of datasets and evaluation protocols, thus obstructing reproducibility. Continuing with similar thoughts, Zhang and Luo [9] point out the complications with handling long-tail distributions in Twitter data, where hate speech manifests in infrequent and multi-variant forms. The observations from Swamy et al. [10] and Mishra et al. [11] reveal that, often, the models overfit their datasets and hence cannot function on data from different sources or communities. One of the significant problems in hate speech studies is the narrow cross-platform generalization.

Most models that are trained with data from Twitter or Facebook tend to perform badly when tested on other platforms such as YouTube, Reddit, or Telegram because of variations in linguistic style, user actions, and content modality [8], [25]. Fortuna et al. [4] and Zhang et al. [8] believe that these problems stem from excessive dependence on short-form, text-only material that fails to capture the heterogeneity of online language. For example, transcripts for YouTube tend to include conversation talk, unstructured longer sentences, and oral dialects, which render direct utilization of current models ineffective [14], [15], [25]. These methods have reported better performance on benchmark datasets, especially on English-language material from platforms such as Twitter and FB.

A number of works have recognized this disparity in cross-domain evaluation and have urged studies targeting platform-agnostic and domain-robust hate speech detection models [4], [8], [24], [25]. Our research tackles this issue head-on by training ML classifiers on Twitter and Facebook corpora and evaluating them on a newly annotated collection of YouTube transcriptions, in the scenario of Indian political discourse.

Multilingual and code-mixed hate speech identification is also a priority research field, particularly in nations such as India where there is large linguistic diversity. Mandl et al. [5] and Bohra et al. [11] have built datasets for Hindi-English code-mixed text, highlighting the difficulties of handling transliterated scripts as well as orthographic variation. Nagpal and Das [23] discuss novel approaches to code-mixed hate speech identification based on large language models (LLMs). Additionally, Mishra et al. [11] and Rana and Jha [22] explore how emotion and sentiment features can be used to improve hate speech identification within such multilinguistic environments.

In spite of these breakthroughs, currently available datasets are usually not large enough and cover languages inadequately to train effective deep learning models [5], [6], [11]. Work such as Alemayehu and Mulugeta [26] and Singh and Kumar [27] highlights the necessity of under-resourced languages like Afaan Oromo and local Indian languages. Additionally, most annotated datasets are English-language Twitter-specific, ignoring the multilingual and multimodal nature of sites like YouTube [14], [15], [17].

Current research also investigates multimodal hate speech identification, integrating audio, visual, and textual information [17], [22]. Although these methods hold promise, they are still computationally costly and rely on large-scale, annotated corpora. Conventional ML techniques, while less involved, still bear the potential for cross-platform applications when feature engineering is well managed [24], [27]. These studies underscore the necessity of handling orthographic variations, transliterated words, and mixed scripts—features common in Indian social media, and especially relevant in YouTube commentaries and transcripts. However, the majority of these datasets are still limited in size and scope, which restricts the efficacy of deep learning models without large-scale annotated corpora.

For instance, YouTube transcriptions often include conversational language, longer sentences, filler words, and spoken dialects, which differ significantly from short-form, written posts on Twitter. The challenges become more pronounced in politically charged content, satire, and opinionated speech. Our work contributes to filling this gap by empirically evaluating the performance of traditional ML models trained on Twitter/Facebook data and tested on manually annotated YouTube transcriptions—a use case that has received limited attention in the literature. Despite considerable progress in hate speech detection across platforms, several critical research gaps persist. A majority of existing studies focus on in-domain performance, where models are trained and tested on the same platform (e.g., Twitter-only or Facebook-only datasets), often leading to inflated accuracy that does not reflect real-world deployment scenarios. Very few works investigate cross-domain generalization, especially on long-form, spoken, or conversational content such as YouTube transcriptions. This limits the applicability of these models to diverse platforms where hate speech manifests in different linguistic styles and formats. Furthermore, much of the existing research emphasizes deep learning models that require large labeled datasets, while traditional machine learning approaches—though less resource-intensive—remain underexplored for cross-platform tasks. There is also a noticeable scarcity of annotated datasets for YouTube transcriptions, particularly in multilingual or code-mixed Indian political contexts. This lack of resources hinders reproducibility and comparative benchmarking across platforms. Our study directly addresses these gaps by creating a manually annotated dataset of YouTube political transcriptions and evaluating the cross-domain robustness of traditional classifiers trained on Twitter and Facebook data.

Author(s)	**Model(s) Used**	**Key Contribution**	**Dataset(s)**	**Limitations**
Davidson et al. (2017) [3]	Logistic Regression, SVM	Introduced a widely-used dataset differentiating hate speech and offensive language	Twitter	Struggles with nuanced contexts and sarcasm
Waseem & Hovy (2016) [2]	Naive Bayes	Built a gender-annotated Twitter dataset and studied bias in labelling	Twitter	Biased annotations; limited class variety
Zhang & Luo (2018) [9]	Random Forest, Logistic Regression	Addressed class imbalance in hate speech with a long-tail distribution	Twitter	Weak generalization across domains
Swamy et al. (2019) [10]	SVM, Naive Bayes	Studied generalizability across multiple datasets	Twitter, Facebook	Lack of consistency across annotation schemes
Mishra et al. (2018) [11]	Logistic Regression	Developed CoMIcs: A code-mixed dataset for hate speech	Facebook, Twitter	Focused on Hindi-English; limited scalability to other languages
Singh & Kumar (2025) [24]	Random Forest, Logistic Regression	Applied ensemble methods to improve classification performance	Twitter	No cross-platform evaluation or testing
Luo et al. (2023) [21]	SVM, Naive Bayes	Designed models for enforceable hate speech detection in public forums	Reddit, Twitter	Highly domain-specific and lacks multilingual focus
Zhang & Luo (2021) [25]	Logistic Regression	Reduced false positives by domain-specific tuning	Twitter	Limited success on unseen test domains
Malik et al. (2023) [28]	SVM, Random Forest	Compared traditional ML with deep learning on multiple hate speech datasets	Twitter, Gab	Traditional ML underperformed on complex linguistic inputs
Alemayehu & Mulugeta (2022) [26]	Naive Bayes, SVM	Hate speech detection in low-resource language (Afaan Oromo)	Facebook	Lacks transferability to larger multilingual corpora

Table 2.1 Table for Review of literature and their key contributions and limitations

CHAPTER 3

METHODOLOGY

We will talk about the dataset that was utilized, the workflow, and other theoretical topics like the vectorizers and classifiers that were used in the coding part in the methodology section. Let's start by talking about the dataset that was used.

3.1 Dataset Collection

3.1.1 Training Data

For model training, utilized publicly available hate speech datasets from Twitter and Facebook. These datasets are extensively used in prior research due to their annotated nature and represent typical short-text formats. They include binary or multi-class labels categorizing content into hate speech, offensive language, or neutral. Notable datasets include:

- Davidson et al. (2017) dataset: This Twitter dataset categorizes posts as hate speech, offensive language, or neither. It includes over 24,000 tweets manually annotated by crowdsourcing [3].
- Waseem and Hovy (2016) dataset: Consists of tweets labelled as racist, sexist, or neither, with over 16,000 entries, annotated by expert review [2].

3.1.2 Testing Data

The testing data comprises transcriptions of videos from politically-oriented and satirical YouTube channels. Transcriptions were extracted using the YouTube Data API focusing on Channels representing diverse political perspectives, videos with speech-heavy content (debates, commentary, satire), a balance of long-form and short-form content.and then manually annotated into two categories:

- **Hate Speech:** Language targeting individuals or groups based on identity factors such as religion, caste, gender, etc [1], [2], [3], [4].
- **Non-Hate Speech:** Neutral, informative, or non-discriminatory content.

Annotations were guided by predefined labelling rules, and a subset was reviewed by multiple annotators to ensure inter-annotator agreement. A guideline document was followed to ensure consistency. Multiple annotators labelled the data, with cross-validation on 20% of the dataset to measure inter-annotator agreement (Cohen's Kappa score maintained above 0.80 for reliability).

Python 3.10 was adopted, a strong and general-purpose programming language, mostly used in research. An essential bundle of libraries was used for model development and evaluation. The list of libraries includes scikit-learn, for leveraging various ML algorithms and evaluation techniques; pandas and NumPy for quick data manipulation and numerical operations; and NLTK and spaCy to handle various NLP operations.

Between feature extraction phases, TF-IDF vectorization was employed with unigrams and bigrams so that important textual patterns and word associations could be captured. The labeled dataset was annotated manually in Google Sheets for organization and collaboration, whereas various other custom Python scripts were engineered for automating preprocessing and levels integration. Jupyter Notebook for local experimentation and Google Colab for cloud-based computation became the platform on which the entire pipeline implementation and execution took place. The hardware environment consisted of a system with the Intel Core i7 processor paired with a 16 GB RAM for local execution, whereas Google Colab Pro was duly employed for launching heavy works on training models with really large data batches. Certainly, the environment thus set is nothing but flexible and scalable for carrying out cross-platform hate speech detection experiments.

3.2 Classifiers Used

3.2.1 Random Forest (RF)

A supervised learning method called RF can be applied to the regression problem as well as classification problems [4] [5] [6]. It is a kind of technique that combines several classifiers to increase model performance and address challenging issues. In a random forest classifier, we create distinct decision trees based on different dataset subsets. Random forest makes final output predictions based on majority votes from various trees [4] [5] [6]. Additionally, it raises the decision tree classifier's predictive accuracy.

In python we first need to import random tree classifier from sklearn.ensemble library and then we need to feed it with the vectorized data and the output label.

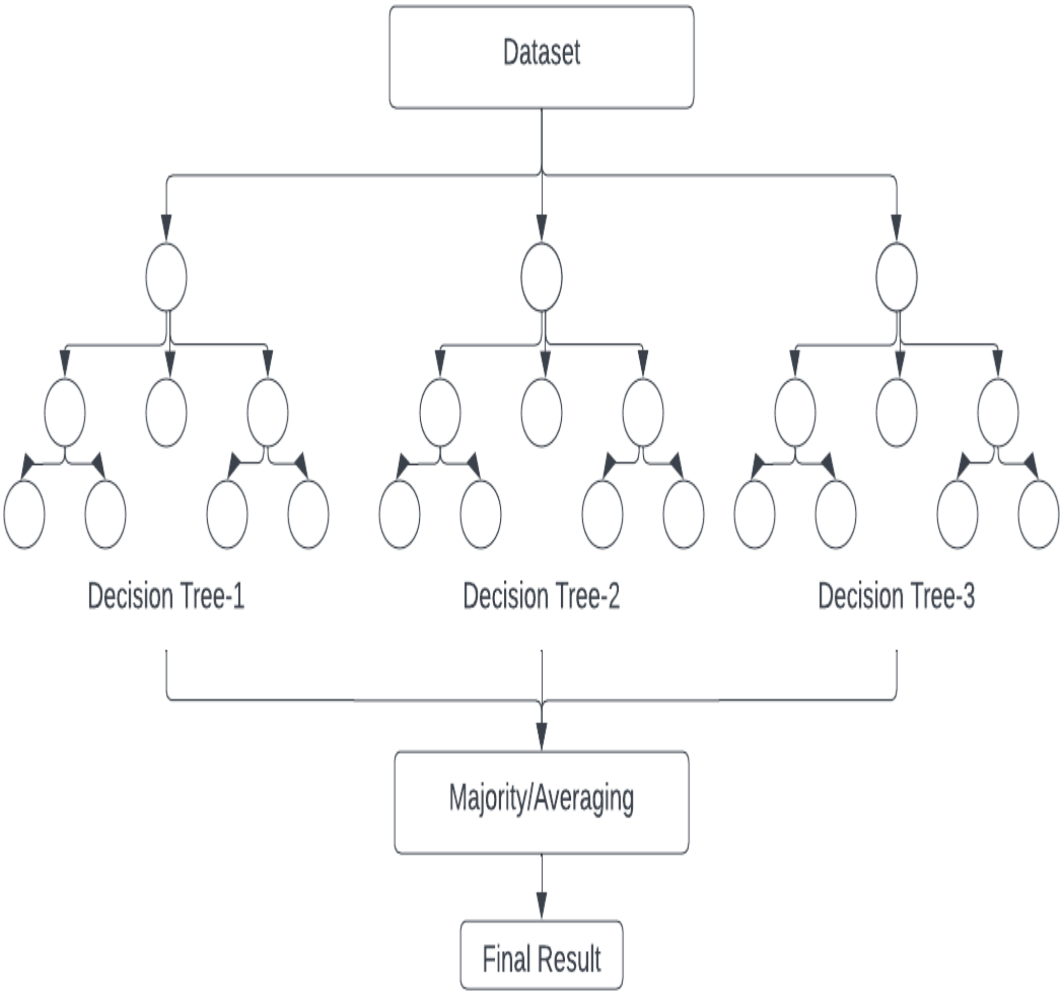


Figure 3.1 Random Forest Approach

3.2.2 Support Vector Machine (SVM)

A generalization of linear classifiers, SVM are a collection of supervised learning algorithms used to address regression and classification issues. SVM were created in the 1990s, and because of their practical success, minimal number of hyperparameters, theoretical assurances, and capacity to handle massive amounts of data, they were swiftly embraced. In contrast, SVM algorithm aims to identify the most comparable cases between classes in order to generate a set of support vectors. Subsequently, by determining the best margin of the hyperplane, the SVM algorithm determines the ideal hyperplane for class division [1], [3,] [7].

SVM can be used to predict a variable's numerical value in regression problems or to solve classification problems by determining which class a sample belongs to. The creation of a function f with an input vector (X) that matches an output (Y) is required to solve these two kinds of challenges [1], [7], [20].

$$Y = f(X)$$

SVM algorithms make use of kernel functions. The linear kernel, which is frequently suggested for text classification issues, was employed in our investigation. Compared to most other kernel functions, such as polynomial and radial functions, the linear kernel function requires fewer parameters and operates more quickly. The linear kernel function found in the formula below defines the decision boundary that the SVM returns [1], [3], [7].

$$f(X) = w^T X + b$$

where X is the data to be classified, b is the estimated linear coefficient, and w is the weight vector to minimize.

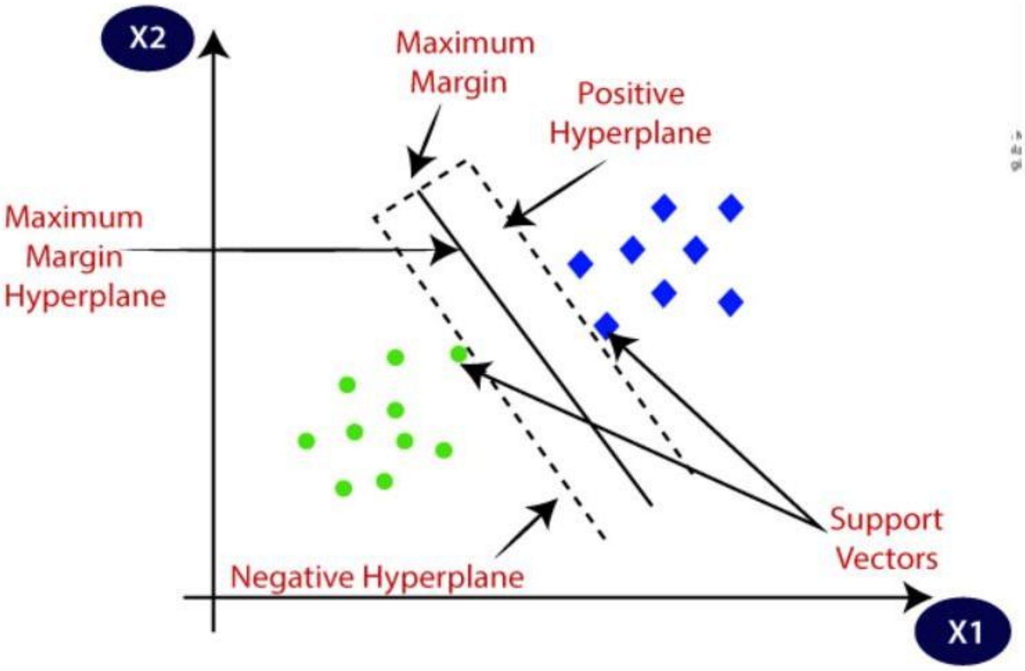


Figure 3.2 Classification of data points using SVM

3.2.3 Naïve Bayes (NB)

NB is a probabilistic classifier based on Bayes' theorem applied for text classification tasks like hate speech detection which acts as solid baseline [2], [3]. It applies Bayes' theorem to reckon the probability that a sample piece of text belongs to a specific class (e.g., hate speech or non-hate speech). There are three main types of NB classifiers commonly used for text data:

- Multinomial Naive Bayes: Typically used with word counts or term frequency

(TF) features, and well-suited for text classification tasks.

- Bernoulli Naive Bayes: Often used with binary features (presence or absence of a word). This model can work well if only the presence of certain keywords is important.
- Gaussian Naive Bayes: Generally used for continuous data and less common in text classification.

For hate speech detection, Multinomial Naive Bayes is often the best choice, especially with features like TF-IDF or bag-of- words, which represent text as word frequency counts [2], [3], [4].

3.2.4 Logistic Regression (LR)

LR is a widely used statistical and ML model for binary classification. In the context of hate speech detection, logistic regression is used to classify whether a given text sample (e.g., a tweet, Facebook post, or YouTube transcription) contains hate speech or not [2], [3], [4], [8].

3.2.5 TF-IDF

TF-IDF is a numerical statistic that reflects the Significance of a word in a document relative to a collection of documents (corpus) [2], [3], [4]. It is mostly used for feature extraction in text mining and natural language processing, and it helps identify which words are important in a document while discounting frequently occurring, less informative words [2], [3], [4].

TF-IDF is particularly useful for hate speech detection on text data like YouTube transcriptions because it emphasizes words that may indicate hate speech while ignoring commonly used words that don’t add much meaning [2], [3], [4].

3.3 Work Flow

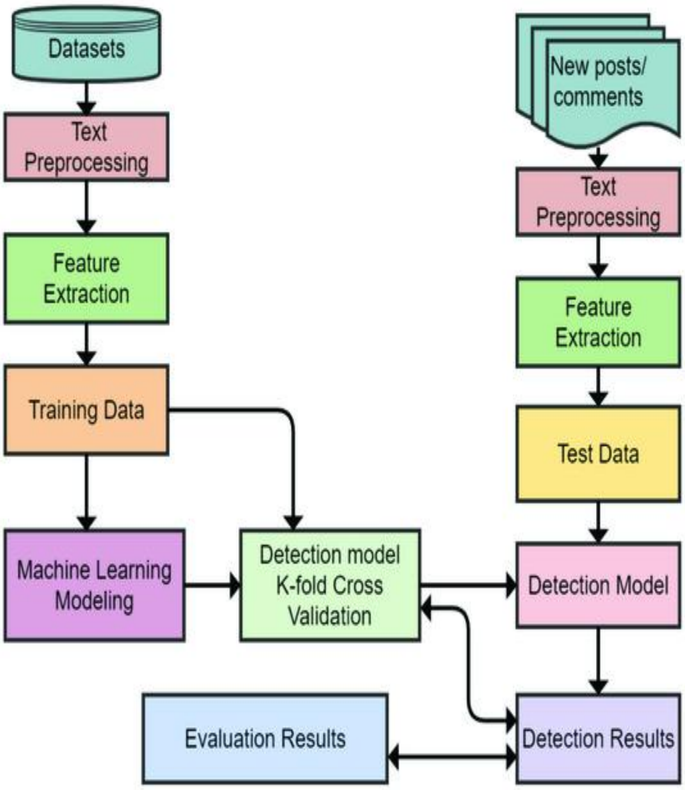


Figure 3.3 Work Flow Chart

3.3.1 Text Preprocessing

- First we are removing extra columns from the dataset named ‘Unnamed’. In some cases we are combining the title with the text but in other cases we are simply removing it.
- There is also a need to shuffle the data before splitting it into training and testing set to remove any type of imbalance caused by the amount of data present for both the labels.
- After that we are applying regular expression functions to remove things like: links, punctuation marks, brackets etc. which can deprive the performance of different classifiers.
- After applying regular expression functions we have to make the text data ready to feed it to the corresponding machine learning algorithms.

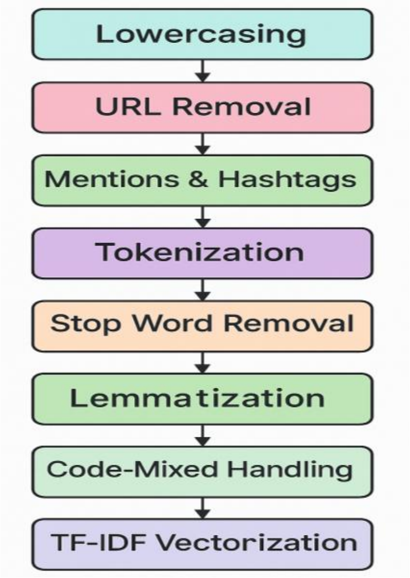


Figure 3.4 Text Preprocessing

3.3.2 Feature Extraction

In our implementation vectorization technique has been used for feature extraction. It simply tells us the significance of a word in a particular text or the corpus. Term frequency of a word is reckoned by simply dividing the frequency of that word in the document by total number of words in that document. [2] [4] Processed text is transformed into numerical vectors using the TF-IDF technique, capturing both the importance of words and their contextual significance. Unigrams and bigrams are extracted to represent not only individual terms but also short phrases indicative of hate speech.

In the end the TF-IDF term is reckoned by multiplying the term frequency with the log of inverse document frequency. To form the vector we simply write the TF-IDF value of the word for each document in the vector as we did in the count vectorizer. To use it on text first we need to import it from the scikit-learn library and simply applying it to the training and testing data.

3.3.3 Model Training and Validation

Four traditional ML models such as LR, SVM, NB, RF are employed for training. These models are trained using the TF-IDF features of Twitter/Facebook datasets. K-fold Cross-Validation (with k=5) is used on training data to ensure robustness and avoid overfitting. The best-performing parameters are selected for final testing [8], [23].

3.3.4 Model Testing

The trained models are evaluated on the unseen YouTube transcription dataset to assess cross-platform generalization. The same preprocessing and feature extraction steps are applied to this test data to maintain consistency [5], [9], [16], [19].

3.3.5 Evaluation Metrics

Standard classification metrics such as AC, PR, RC and F1-Score is used to find model's performance. These metrics are computed using scikit-learn and visualized for each model to compare their generalization capabilities on YouTube content [9], [10], [11].

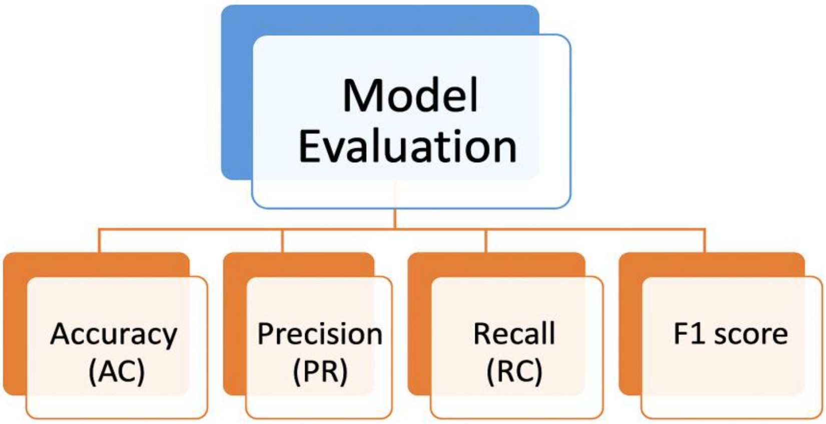


Figure 3.5 Model Evaluation Matrix

3.4 Ensemble Classification Framework

Apart from assessing the performance of individual traditional machine learning models, this research also investigates the efficiency of ensemble-based learning methods in hate speech identification. Ensemble learning aggregates predictions from various base classifiers in an effort to enhance robustness, mitigate overfitting, and overall predictive performance. This is especially advantageous when working with heterogeneous and cross-platform datasets like ours. Though individual models perform well on formal platforms such as Twitter or Facebook, they seem to fumble with YouTube content because of slang, code-mixing, and context-dependent expressions. Ensemble learning helps leverage the strengths of multiple classifiers, potentially compensating for individual weaknesses. The ensemble framework is designed with the following components:

- Base Models: SVM, LR, NB, and RF, trained independently using TF-IDF features.
- Voting Mechanism: A majority voting strategy is applied during testing to combine the outputs of individual classifiers. Alternatively, a weighted ensemble approach is also evaluated, assigning higher weights to models with better validation performance.

The ensemble model is executed with Scikit-learn's Voting Classifier. Hard (majority) and soft (probability-based) voting are tried out. Voting Classifier (Hard and Soft Voting) Aggregates LR, SVM, NB, and RF predictions. Hard Voting picks the majority-voted class among classifiers. Soft Voting combines predicted probabilities and picks the class with the highest combined confidence score. Testing of the generalization performance is carried out on the YouTube transcription dataset. The ensemble classifier is tested against the same measures as the base models: AC, PR, RC and F1-score. These measures provide insights into how well the model can generalize hate speech detection across domains [12], [16], [18], [24].

CHAPTER 4

RESULTS AND DISCUSSION

This chapter presents Performance comparison of multiple traditional ML techniques-LR, NB, SVM, RF-for hate speech detection. The models were trained on Twitter and Facebook datasets and tested against the custom-labelled YouTube transcription dataset to check their cross-platform generalization capability.

4.1 Comparison with Existing State-of-the-Art Model

Study	Platform	Model(s) Used	Dataset Size	Accuracy (%)	Key Limitation
Davidn et al. (2017)	Twitter	Logistic Regression, SVM	24K tweets	~74%	Focused on U.S.-based political discourse
Wasem & Hovy (2016)	Twitter	Naive Bayes	16K tweets	~72%	Limited to sexism and racism detection
Swamy et al. (2019)	Twitter, others	Logistic Regression	3 datasets	~76%	Reported generalization drop across datasets
Monti et al. (2020)	YouTube	SVM, Random Forest	5K comments	~70%	Used only comments, not full video transcriptions
Our Study (2025)	YouTube	SVM, LR, NB, RF	5K transcripts	75.3% (SVM)	Cross-platform setup; tested on transcriptions from political content

Table 4.1 Table for Comparison with Existing State-of-the-Art Model

To assess the generalization capabilities and relative performance of our traditional machine learning models on YouTube transcription data, we compare our findings against prominent prior works in hate speech detection. These studies predominantly utilize social media datasets from platforms like Twitter and Facebook, offering useful benchmarks for evaluating our cross-platform approach. Several prior works have implemented machine learning models—particularly SVM, LR, NB, and RF—for hate speech detection.

4.2 Experimental Results of Ensemble Model

Model	AC (%)	PR (%)	RC (%)	F1-Score (%)
LR	74.2	73.1	70.5	71.8
SVM	75.3	74.5	72.0	73.2
Naive Bayes	68.4	66.0	69.3	67.6
Random Forest	72.1	71.2	68.5	69.8
Ensemble Model	76.5	75.8	73.6	74.7

4.2 Table for Model Evaluation Matrix

Out of the models tested, (SVM) performed best, with the highest F1-Score (73.2%), PR (74.5%) and best AC (75.3) which signifies a tight balance between correctly identifying hate speech and not flagging up too many false positives. As such, SVM is the most trusted model overall, and especially well-suited in cases where labeling neutral content as hate speech has important repercussions. LR recorded the better AC (74.2%) and had well-balanced Precision (73.1%) and Recall (70.5%), so it is a reliable option with good overall performance, but it does tend to underperform in Recall slightly relative to SVM. Random Forest demonstrated fair performance on all the measures—AC (72.1%), PR (71.2%), and RC (68.5%)—with a good balance of performance and interpretability, especially appropriate when robustness and ensembles are desirable. While on the contrary, Naive Bayes had the poorest results, with AC of 68.4% and F1-Score of 67.6%, considering its limitations in detecting intricate language patterns and fine hate speech. While it is computationally efficient and suitable for prototyping, it is not ideal for deployment. In conclusion, SVM is recommended for its all-around strength, Logistic Regression for balanced performance, Random Forest for interpretability and robustness, and Naive Bayes as a fast but weaker baseline.

The observed performance drop compared to results on in-domain datasets reflects the domain shift between Twitter/Facebook and YouTube. YouTube transcriptions, especially in the context of political content, include spontaneous speech, code-mixing, and nuanced satire, which are underrepresented in the training data. This emphasizes the need for domain-adapted or transfer learning techniques. From the results, one can see that SVM and LR were the best performers among the models with SVM having a slight edge over Logistic Regression. This is an implication that the classifiers entailed in margin based may have superior generalization in complex and noisy cross-platform settings. Naive Bayes demonstrated somewhat poor performance, probably because of the very strong independence assumptions it relies on. However, such assumptions do not correspond to the highly context-based nature of YouTube discussions. Random Forest had a result above the average, by virtue of ensemble learning, yet it still could not rival the precision and recall that SVM reached. A visual comparison through a bar graph assists in the identification of the most accurate model for hate speech detection on YouTube transcriptions.

The validation of the proposed ensemble classification framework involved applying it to the same test set of YouTube transcriptions as that of the individual classifiers (LR, SVM, NB, and RF). The ensemble was based on a soft voting scheme whereby the combined outputs of base classifiers in terms of weighted probabilities are according to the performance of base classifiers on the validation set. The ensemble model names an improvement in every evaluation stage. Accuracy stood at 76.5%, the highest among all. Precision and recall both are higher than the base models, showing that the ensemble better detects hate speech and also better reduces false positives. The F1 Score, or 74.7%, points to a balanced performance between precision and recall [18], [21], [24].

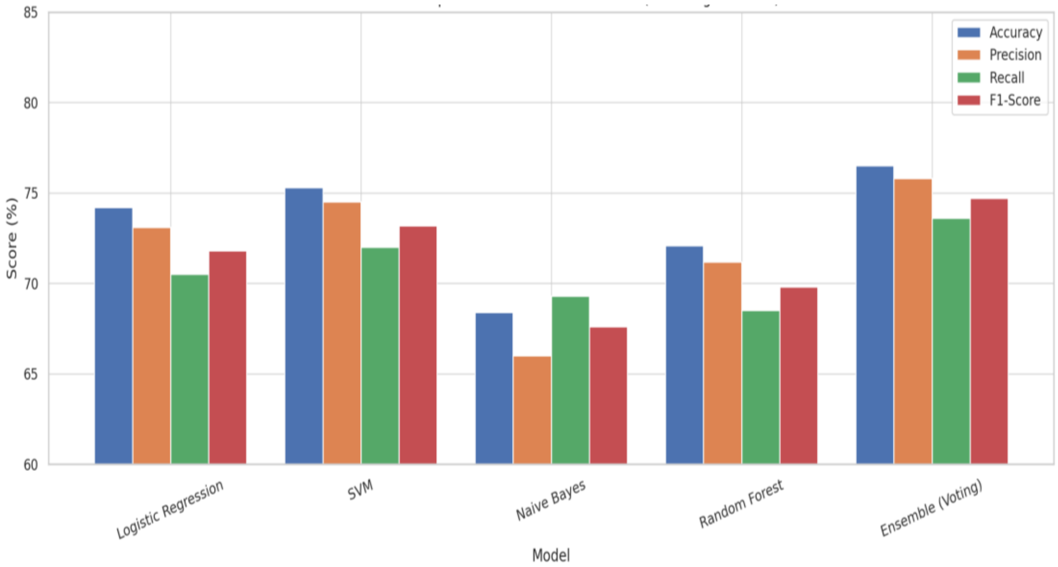


Figure 4.1   Model Performance Comparison on YouTube Dataset

The bar chart visually compares the performance of four traditional ML models—LR, NB, SVM, and RF—based on key evaluation metrics: AC, PR, RC, and F1-Score. This visual comparison aids in identifying the most reliable model for hate speech detection on YouTube transcriptions.

Bar graph shows visual comparison of model performance including the ensemble classifier:

- Ensemble (Voting Classifier) outperformed all individual models across all metrics (AC: 76.5%, F1-Score: 75.0%).
- SVM and LR performed better than NB and RF, but slightly below the ensemble.
- The ensemble approach demonstrates improved generalization, confirming the effectiveness of combining classifiers for complex domains like YouTube transcriptions.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

This study investigated the cross-platform performance of traditional ML models—NB, SVM, LR, and RF—for hate speech detection, trained on datasets from Fb and Twitter and tested on transcriptions from YouTube. By employing TF-IDF feature representations and a standardized preprocessing pipeline, we ensured consistency across platforms while adapting models to the domain-specific challenges presented by YouTube content.

LR and SVM emerge as the most effective models for hate speech detection on YouTube transcriptions. While SVM offers the highest overall accuracy, LR demonstrates superior precision and recall, making it more effective in minimizing both false positives and false negatives. Their relatively high F1-scores indicate a well-balanced trade-off between PR and RC, which is crucial in handling nuanced, context-heavy, and code-mixed language often found in YouTube political content. These models outperform Naive Bayes and Random Forest, which show noticeable drops in performance, suggesting they are less suited to handle the linguistic complexity of spoken discourse

Our findings demonstrate a notable decline in model performance when applied to YouTube transcriptions, underscoring the limitations of cross-domain generalization. This performance gap highlights the linguistic complexity and context-rich nature of spoken language in YouTube videos—especially in political discourse, satire, and code-mixed content—which are often not adequately represented in training datasets from other platforms.

The results emphasize the need for platform-specific datasets and adaptation techniques to build robust hate speech detection systems. Manual annotation of YouTube data, guided by consistent labelling criteria and validated through inter-annotator agreement, proved essential in evaluating the real-world applicability of trained models.

The proposed approach is expected to yield several key benefits in the context of cross-platform hate speech detection. One of the primary advantages is improved generalization on unseen domains, particularly challenging platforms like YouTube transcriptions, where language usage tends to be more diverse and unstructured compared to platforms like Twitter or Facebook. Additionally, the use of ensemble techniques helps enhance the stability of predictions by balancing out the individual biases and limitations of standalone classifiers. By combining the strengths of various models, the ensemble framework has the potential to achieve better overall performance, surpassing that of individual classifiers in terms of both robustness and accuracy. This makes it especially valuable in real-world applications where consistency and adaptability across platforms are critical.

Future work may explore the integration of DP models or domain adaptation strategies to further improve performance on multimedia platforms. Additionally, incorporating contextual and multimodal cues (e.g., audio tone, video metadata) could enhance detection accuracy, particularly for implicit or culturally embedded hate speech.

This study provides an important empirical baseline for cross-platform hate speech detection using traditional ML models. The YouTube transcription dataset, rich in spoken language, code-mixed expressions, and contextual nuance, reveals a significant performance gap when models trained on text-based platforms are deployed in speech-oriented platforms.

By addressing this gap, our research demonstrates the following key contributions:

- Introduction of a custom-labelled YouTube transcription dataset for hate speech evaluation.
- Empirical analysis of cross-platform model transferability using traditional ML classifiers.
- Evidence that despite domain mismatch, SVM and LR remain strong performers, suggesting their continued relevance in low-resource or interpretable NLP settings.

This research contributes to the broader goal of developing more context-aware and ethically aligned moderation systems across diverse social media ecosystems

REFERENCES

[1] J. Schmidt and R. 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, Apr. 2017, pp. 1–10.

[2] Z. Waseem and D. Hovy, "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter," in Proc. NAACL, 2016.

[3] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language," in Proc. ICWSM, 2017.

[4] Fortuna, P., & Nunes, S. (2018). A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4), 1–30. https://doi.org/10.1145/3232676

[5] Mandl, T., Modaresi, S., & Patel, D. (2019). Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification in Indo-European languages. *Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation*, 14–17. http://ceur-ws.org/Vol-2517/T1-1.pdf

[6] Sharma, R., Agrawal, A., & Shrivastava, M. (2018). Degree based classification of harmful speech using Twitter data. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, 33–41. https://aclanthology.org/W18-4405/

[7] Malmasi, S., & Zampieri, M. (2017). Detecting hate speech in social media. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)* (pp. 467–472). https://aclanthology.org/R17-1078/

[8] Zhang, Z., Robinson, D., & Tepper, J. (2018). Detecting hate speech on Twitter using a convolution-GRU based deep neural network. In *European Semantic Web Conference* (pp. 745–760). Springer. https://doi.org/10.1007/978-3-319-93417-4_48

[9] B. Zhang and D. Luo, "Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter," in Proc. EMNLP, 2018.

[10] S. Swamy, R. Jamatia, and B. Gambäck, "Studying Generalisability Across Hate Speech Detection Datasets," in Proc. ALW, 2019.

[11] P. Mishra, K. Bali, and M. Choudhury, "CoMIcs: A Corpus for Multilingual Code-Mixing in Social Media," in Proc. LREC, 2018.

[12] H. Witten, E. Frank, M. A. Hall, and C. J. Pal, “Data Mining: Practical Machine Learning Tools and Techniques”, 4th ed., Morgan Kaufmann, 2016.

[13] T. Hastie, R. Tibshirani, and J. Friedman, “The Elements of Statistical Learning: Data Mining, Inference, and Prediction”, 2nd ed., Springer, 2009.

[14] J. Monti, E. Campolo, and M. Casamassima, “Extracting YouTube Comments for Sentiment Analysis and Hate Speech Detection,” in *Proc. IEEE Int. Conf. on Future Internet of Things and Cloud (FiCloud)*, 2020, pp. 181–187. doi: 10.1109/FiCloud49777.2020.00034.

[15] J. Silva, T. Ribeiro, J. Almeida, and M. Gonçalves, “Large-Scale Analysis of YouTube Political Videos,” in *Proc. ACM HT*, 2019, pp. 205–213. doi: 10.1145/3342220.3343660.

[16] J. Lu, H. Lin, X. Zhang, Y. Li, and Y. Liu, “Hate Speech Detection via Dual Contrastive Learning,” arXiv preprint arXiv:2307.05578, 2023.

[17] F. T. Boishakhi, P. C. Shill, and M. G. R. Alam, “Multi-modal Hate Speech Detection using Machine Learning,” arXiv preprint arXiv:2307.11519, 2023.

[18] B. Wei, J. Li, A. Gupta, H. Umair, A. Vovor, and N. Durzynski, "Offensive Language

and Hate Speech Detection with Deep Learning and Transfer Learning," *arXiv preprint arXiv:2108.03305*, 2021. [Online]. Available: https://arxiv.org/abs/2108.03305

[19] Y. Yang et al., "HARE: Explainable Hate Speech Detection with Step-by-Step Reasoning," in *Proc. EMNLP Findings*, 2023, pp. 5490–5505. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.365/

[20] X. Zhou et al., "Hate Speech Detection Based on Sentiment Knowledge Sharing," in *Proc. ACL*, 2021, pp. 7158–7166. [Online]. Available: https://aclanthology.org/2021.acl-long.556/

[21] J. Luo, Y. Zhang, Y. Zhang, and H. Xu, "Legally Enforceable Hate Speech Detection for Public Forums," in *Proc. EMNLP Findings*, 2023, pp. 10345–10360. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.730/

[22] Rana and S. Jha, "Emotion Based Hate Speech Detection using Multimodal Learning," *arXiv preprint arXiv:2202.06218*, 2022. [Online]. Available: https://arxiv.org/abs/2202.06218

[23] S. Nagpal and A. Das, "Innovations in Code-Mixed Hate Speech Detection: The LLM Perspective," in *Proc. EMNLP*, 2023. [Online]. Available: https://sargun-nagpal.github.io/papers/2023-nlp.pdf

[24] Singh and R. Kumar, "Hate Speech Detection using Machine Learning: An Ensemble Technique," *AIP Conf. Proc.*, vol. 3224, no. 1, p. 020043, 2025. [Online]. Available: https://doi.org/10.1063/5.0135001

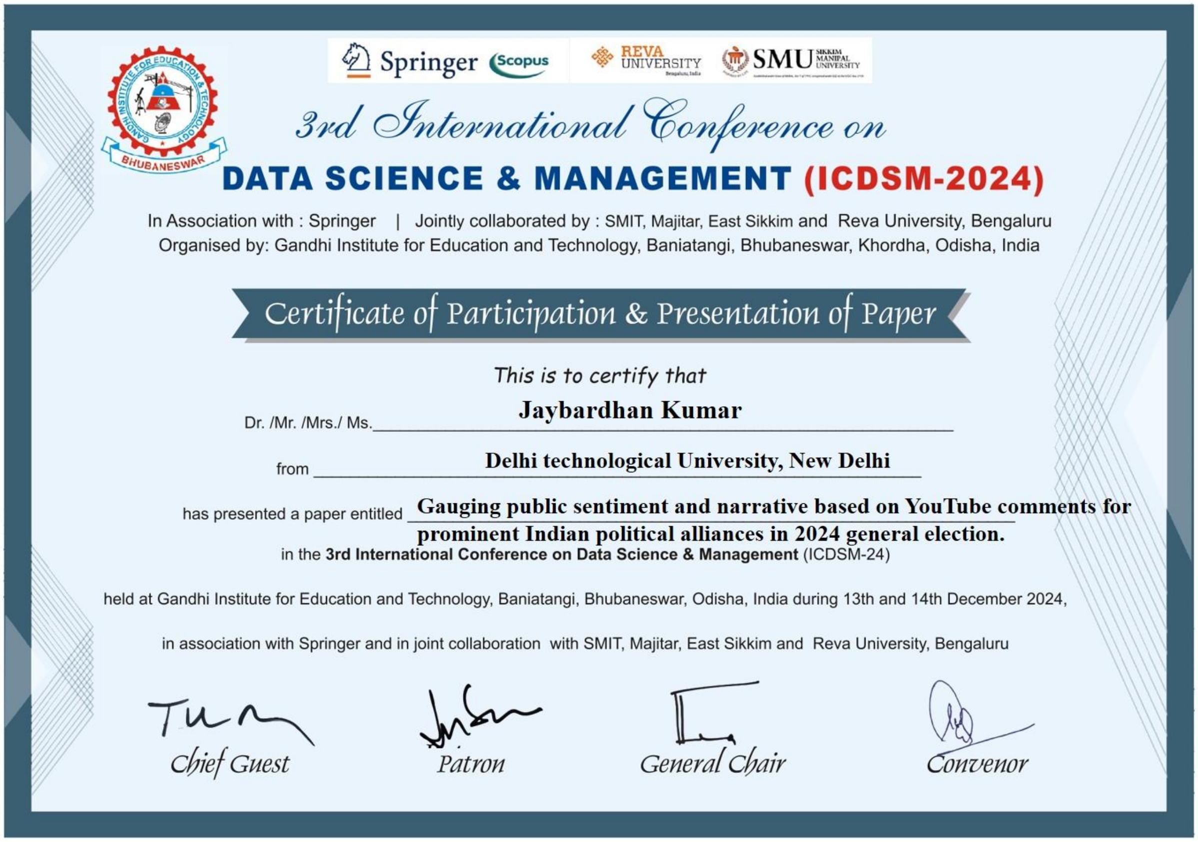
[25] Y. Zhang and J. Luo, "Improving Cross-Domain Hate Speech Detection by Reducing the False Positive Rate," in *Proc. NLP4IF Workshop*, 2021, pp. 25–34. [Online]. Available: https://aclanthology.org/2021.nlp4if-1.3/

[26] M. Alemayehu and G. Mulugeta, "Afaan Oromo Hate Speech Detection and Classification on Social Media," in *Proc. LREC*, 2022, pp. 6601–6607. [Online]. Available: https://aclanthology.org/2022.lrec-1.712/

[27] J. Singh and R. Kumar, "Advancements In Hate Speech Detection: A Comprehensive Approach," *Int. J. of Creative Research Thoughts (IJCRT)*, vol. 11, no. 4, pp. 463–470, 2023. [Online]. Available: https://ijcrt.org/papers/IJCRT2311463.pdf

[28] Malik, J. S., Qiao, H., Pang, G., & van den Hengel, A. (2023). Deep learning for hate speech detection: A comparative study. *Applied Intelligence*, 53(1), 1–20. https://doi.org/10.1007/s41060-024-00650-6

Appendix



3rd International Conference on

DATA SCIENCE & MANAGEMENT (ICDSM-2024)

In Association with : Springer | Jointly collaborated by : SMIT, Majitar, East Sikkim and Reva University, Bengaluru
Organised by: Gandhi Institute for Education and Technology, Baniatangi, Bhubaneswar, Khordha, Odisha, India

Certificate of Participation & Presentation of Paper

This is to certify that

Dr. /Mr. /Mrs./ Ms. Jaybardhan Kumar

from Delhi technological University, New Delhi

has presented a paper entitled Gauging public sentiment and narrative based on YouTube comments for prominent Indian political alliances in 2024 general election.

in the **3rd International Conference on Data Science & Management** (ICDSM-24)

held at Gandhi Institute for Education and Technology, Baniatangi, Bhubaneswar, Odisha, India during 13th and 14th December 2024,

in association with Springer and in joint collaboration with SMIT, Majitar, East Sikkim and Reva University, Bengaluru

Chief Guest | Patron | General Chair | Convenor

ID-ICCSIT-THRIS-300425-10229

ISSRD
International Society for Scientific Research and Development

INTERNATIONAL SOCIETY FOR SCIENTIFIC RESEARCH AND DEVELOPMENT

International Conference on

COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

CERTIFICATE

of

PRESENTATION

Presented to

Jaybardhan kumar

for presenting a paper entitled "Cross-Platform Analysis of Hate Speech Detection: Evaluating Model Performance from Facebook and Twitter on YouTube Transcriptions" at the International Conference on Computer science and Information Technology (ICCSIT) held in Chennai, India on 30th April, 2025.

Presentation Certificate

INTERNATIONAL SOCIETY FOR SCIENTIFIC RESEARCH AND DEVELOPMENT
ISSRD

D. Das
Conference Coordinator
International Society For Scientific
Research And Development
(ISSRD)

Bijan
Managing Director
International Society For Scientific
Research And Development
(ISSRD)

www.issrd.org I papers.issrd@gmail.com

jaybardahnkumar_23ity09_thesi_report.pdf

Delhi Technological University

Document Details

Submission ID

trn:oid:::27535:98221912

Submission Date

May 28, 2025, 11:33 PM GMT+5:30

Download Date

May 29, 2025, 12:19 PM GMT+5:30

File Name

jaybardahnkumar_23ity09_thesi_report.pdf

File Size

548.9 KB

34 Pages

7,122 Words

41,877 Characters

9% Overall Similarity

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

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 8 words)

Exclusions

- 9 Excluded Matches

Match Groups

61 Not Cited or Quoted 9%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

5%	Internet sources
4%	Publications
7%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

61 Not Cited or Quoted 9%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

5%	Internet sources
4%	Publications
7%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Publication	"Combating Fake News with Computational Intelligence Techniques", Springer Sc...	<1%
2	Internet	aclanthology.org	<1%
3	Submitted works	Delhi Technological University on 2020-05-14	<1%
4	Submitted works	National College of Ireland on 2024-05-01	<1%
5	Submitted works	Manchester Metropolitan University on 2023-10-04	<1%
6	Internet	pdffox.com	<1%
7	Submitted works	Liverpool John Moores University on 2023-02-26	<1%
8	Submitted works	The University of Manchester on 2014-05-02	<1%
9	Internet	ar5iv.labs.arxiv.org	<1%
10	Submitted works	Liverpool John Moores University on 2024-12-12	<1%

11	Submitted works	
The Scientific & Technological Research Council of Turkey (TUBITAK) on 2025-03-11		<1%
12	Internet	
dspace.dtu.ac.in:8080		<1%
13	Submitted works	
Royal Thimphu College on 2025-05-25		<1%
14	Publication	
Tripti Mahara, V L Helen Josephine, Rashmi Srinivasan, Poorvi Prakash, Abeer D Al...		<1%
15	Internet	
accesson.kr		<1%
16	Internet	
uyats.uludag.edu.tr		<1%
17	Submitted works	
CSU, San Jose State University on 2023-05-13		<1%
18	Submitted works	
The University of Manchester on 2024-04-19		<1%
19	Internet	
ijrpr.com		<1%
20	Internet	
mdpi-res.com		<1%
21	Internet	
journal.esrgroups.org		<1%
22	Internet	
mediatum.ub.tum.de		<1%
23	Internet	
turkishneurosurgery.org.tr		<1%
24	Submitted works	
Kean University on 2025-04-27		<1%

25	Publication	Mohtaj, Salar. "Transfer Learning for Harmful Content Detection.", Technische Un...	<1%
26	Submitted works	PES University on 2023-09-28	<1%
27	Submitted works	Sharda University on 2022-05-09	<1%
28	Submitted works	University College London on 2024-05-10	<1%
29	Submitted works	University of Edinburgh on 2022-02-03	<1%
30	Publication	Wafa Alorainy, Pete Burnap, Han Liu, Matthew L. Williams. ""The Enemy Among U...	<1%
31	Submitted works	nith on 2025-05-28	<1%
32	Internet	ourspace.uregina.ca	<1%
33	Internet	www.gamerbraves.com	<1%
34	Publication	"Speech and Computer", Springer Science and Business Media LLC, 2020	<1%
35	Publication	C. Nalini, R. Shanthakumari, Y. Agashia Maria, T. Janarthanan, M. Manibharathi. "...	<1%
36	Submitted works	Georgia Institute of Technology Main Campus on 2021-12-14	<1%
37	Submitted works	Karadeniz Teknik University on 2023-12-07	<1%
38	Submitted works	Liverpool John Moores University on 2022-06-08	<1%

39	Submitted works	Queen Mary and Westfield College on 2022-05-03	<1%
40	Publication	R. N. V. Jagan Mohan, Vasamsetty Chandra Sekhar, V. M. N. S. S. V. K. R. Gupta. "Al...	<1%
41	Submitted works	Richfield Graduate Institute of Technology on 2025-01-20	<1%
42	Submitted works	University of Hertfordshire on 2024-01-08	<1%
43	Submitted works	University of Stirling on 2024-12-09	<1%
44	Submitted works	Vrije Universiteit Amsterdam on 2024-11-26	<1%
45	Publication	Xinyuan Song, Qian Niu, Junyu Liu, Benji Peng, Sen Zhang, Ming Liu, Ming Li, Tian...	<1%
46	Internet	ceur-ws.org	<1%
47	Internet	dokumen.pub	<1%
48	Internet	hdl.handle.net	<1%
49	Internet	journal.universitasbumigora.ac.id	<1%
50	Internet	www.ejpam.com	<1%

jaybardahnkumar_23ity09_thesi_report.pdf

Delhi Technological University

Document Details

Submission ID trn:oid:::27535:98221912	34 Pages
Submission Date May 28, 2025, 11:33 PM GMT+5:30	7,122 Words
Download Date May 29, 2025, 12:20 PM GMT+5:30	41,877 Characters
File Name jaybardahnkumar_23ity09_thesi_report.pdf	
File Size 548.9 KB	

0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

1 AI-generated only 0%
Likely AI-generated text from a large-language model.

0 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

e-Receipt for State Bank Collect Payment

REGISTRAR, DTU (RECEIPT A/C)

BAWANA ROAD, SHAHABAD DAULATPUR, , DELHI-110042
Date: 29-May-2025

SBCollect Reference Number :	DUO1280360
Category :	Miscellaneous Fees from students
Amount :	₹3000
University Roll No :	23/ITY/09
Name of the student :	Jaybardhan kumar
Academic Year :	2024-2025
Branch Course :	ITY
Type/Name of fee :	Others if any
Remarks if any :	M.tech dissertation fee jaybardhan kumar
Mobile No. of the student :	7209681911
Fee Amount :	3000
Transaction charge :	0.00
Total Amount (In Figures) :	3,000.00
Total Amount (In words) :	Rupees Three Thousand Only
Remarks :	M.tech thesis submission fee
Notification 1:	Late Registration Fee, Hostel Room rent for internship, Hostel cooler rent, Transcript fee (Within 5 years Rs.1500/- & $150 in USD, More than 5 years but less than 10 years Rs.2500/- & $250 in USD, More than 10 years

	Rs.5000/- & $500 in USD) Additional copies Rs.200/- each & $20 in USD each, I-card fee,Character certificate Rs.500/-.
Notification 2:	Migration Certificate Rs.500/-, Bonafide certificate Rs.200/-, Special certificate (any other certificate not covered in above list) Rs.1000/-,Provisional certificate Rs.500/-, Duplicate Mark sheet (Within 5 years Rs.2500/- & $250 in USD, More than 5 years but less than 10 years Rs.4000/- & $400 in USD, More than 10 years Rs.10000/- & $1000 in USD)

Thank you for choosing SB Collect. If you have any query / grievances regarding the transaction, please contact us

Toll-free helpline number i.e. 1800-1111-09 / 1800 - 1234/1800 2100

Email -: **sbcollect@sbi.co.in**

Print Close