ABUSIVE COMMENTS DETECTION

by

**ABINAYA S 2021115004**

**KARTHIKA P 2021115049**

**YUTHIKSHAA M 2021115123**

# A summer project report

*submitted to the Faculty of*

**INFORMATION AND COMMUNICATION ENGINEERING**

# for the partial fulfillment of the

*award of the degree of*

## Bachelor of Technology

In

## Information Technology

A logo of a university

Description automatically generated

### DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING, ANNA UNIVERSITY

CHENNAI 600 025.

OCTOBER 2023

CERTIFICATE

Certified that this report titled “*ABUSIVE COMMENTS DETECTION”*, is a *bona fide* work of **Abinaya S (2021115004),** **Karthika P (2021115049),** **Yuthikshaa M (2021115123),** who carried out the work under my supervision, for the partial fulfillment of the requirements for the award of the degree of *Bachelor of Engineering* in *Information Technology*. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion.

Place: Chennai. **Dr. M. Deivamani**

Assistant Professor,

Date: Dept. of Information Science and

Technology,

College of Engineering, Guindy

Anna University,

Chennai - 600 025.

# **COUNTERSIGNED**

# **Dr. S. Sridhar**

# Professor and Head

Department of Information Science and Technology

College of Engineering, Anna University

Chennai 600 025.

ACKNOWLEDGEMENT

I would like to express my sincere thanks and deep sense of gratitude to my guide ***Dr. M. Deivamani***, Assistant Professor, Department of Information Science and Technology, for his valuable guidance suggestions and constant encouragement paved way for the successful completion of the project work.

I also thank ***Dr. S. Sridhar***, Professor & Head, Department of Information Science and Technology for his kind support and for providing necessary facilities to carry out the work.

I am bound to convey my sincere thanks to the project committee ***Dr. P. Varalakhsmi***, Professor, ***Dr. S. Bama***, Associate Professor, and

***Dr. K. Arul Deepa***, Assistant Professor, Department of Information Science and Technology for their suggestions and motivations.

## ABINAYA S

## KARTHIKA P

## YUTHIKSHAA M

**ABSTRACT**

In recent years, there has been a significant surge in the volume of digital content shared on various social media platforms. Online Social Networks (OSNs) have become an essential source of news, information, and entertainment. However, alongside the many advantages of using OSNs, there is a growing problem of individuals misusing these platforms to spread harmful and offensive content.

Abusive speech encompasses any form of communication that aims to demean, insult, or incite hatred against vulnerable individuals or groups based on factors such as gender, sexual orientation, ethnicity, or disability, using offensive language. Because of this diversity in thematic foci, the abusive comments are referred to as themes. Examples of topics include misogyny, misandry, homophobia, transphobia, and xenophobia.

Abusive speech causes severe psychological effects on the targeted individual. These abusive comments also can create controversy over social media on a specific individual or a group of people. This shows the need for restricting these kinds of abusive comments from being posted on social media. There is a need to develop a model that can detect and classify such abusive comments.

TABLE OF CONTENTS

CHAPTER NO. TITLE PAGE NO.

ABSTRACT 4

LIST OF TABLES 7

LIST OF FIGURES 8

1 INTRODUCTION 9

1.1 LITERATURE SURVEY 9

1.2 PROBLEM STATEMENT 11 1.3 OBJECTIVE 11

1.4 Overview 11

1.5 ORGANIZATION OF THE REPORT 13

2 SOFTWARE REQUIREMENTS SPECIFICATION 14

2.1 SOFTWARE REQUIREMENTS 14

3 SYSTEM DESIGN 15 3.1 SYSTEM ARCHITECTURE 15

**4 IMPLEMENTATION AND RESULTS 18**

4.1 IMPLEMENTATION 18

4.1.1 Algorithm for fetching comments 18

from YouTube

4.1.2 Algorithm for text preprocessing 18

­­

4.1.3 Algorithm for Abusive keyword 19

detection

4.1.4 Algorithm for user interface display 19

4.2 RESULTS 19

5 CONCLUSION AND FUTURE WORK 23

REFERENCES 24

LIST OF TABLES

TABLE NO. TITLE PAGE NO.

4.1 Demonstration of the analyzer 20

LIST OF FIGURES

FIGURE NO. TITLE PAGE NO.

3.1 System Architecture 17

* 1. Main page taking YouTube link as input 21
  2. Result page labelling abusive and non-abusive 22

YouTube comments

**CHAPTER 1**

**INTRODUCTION**

* 1. **LITERATURE SURVEY**

Literature in the following areas has been analyzed:

[1] Pretrained Models and dataset.

[2] YouTube Universe of Comments: A Machine Learning Approach

[3] Social Media Insult Detection

The study [1] explores the use of transformer models to tackle offensive language detection. Offensive language identification is a complex problem, varying significantly across countries and languages.The datasets used in this research may not encompass all possible offensive language usages.

The investigation compares different methods for detecting offensive language within textual data.The research highlights the significance of pretrained transformer models for addressing the challenge of identifying offensive language.These models primarily target offensive language in English, with ongoing research in multilingual approaches encompassing both text and speech.

The research paper [2] addresses the issue of offensive comments on YouTube, one of the largest platforms for user-generated content. Due to its enormous volume of data, managing comments is crucial to efficiency and sustainability. The paper outlines a comprehensive annotation scheme for categorizing comments based on dimensions like sentiment, topic, toxicity, and engagement. Machine learning techniques are employed to identify and categorize offensive comments, which constitute a significant portion of redundant content.

The study evaluates multiple classification algorithms, such as support vector machines, random forests, and deep learning models, to categorize comments based on the predefined annotation scheme.

The paper [3] Defines social media insults and the potential harm they inflict on users.Discusses the need for automated methods to detect and prevent online abuse.Explains the relevance of natural language processing (NLP) and machine learning in this context. Details the significance of NLP techniques in processing and analyzing text data from social media platforms.

Explains how NLP can be employed for tasks such as text preprocessing, sentiment analysis, and abusive content detection. Presents an overview of the research methodology employed in the reference paper.Describes the data collection, preprocessing, and feature engineering steps.Explains the implementation details of insult detection techniques, including the comparative analysis.

* 1. **Problem statement**

Abusive speech causes severe psychological effects on the targeted individual. These abusive comments also can create controversy over social media on a specific individual or a group of people. This shows the need for restricting these kinds of abusive comments from being posted on social media (like YouTube). Therefore, there is a need to develop a model that can detect and label such abusive comments.

* 1. **Objective**

The objective of the project is to create a user-friendly web application that empowers users to input YouTube video links for comment analysis. Through integration with the YouTube API, the project aims to retrieve comments, preprocess them using various natural language processing techniques, and detect potentially abusive content using a predefined set of abusive keywords. The key objective is to provide a seamless and comprehensible interface for users to access analysis results. This project ultimately strives to foster a positive online community by promoting respectful interactions and safe content sharing.

**1.4 OVERVIEW of the Project**

**Develop a Web Application**

Build a web application that allows users to input a YouTube video link for analysis.

**Retrieve and Preprocess Comments**

Utilize the YouTube API to fetch comments from the specified video. Apply text preprocessing techniques such as tokenization, stemming, lemmatization, and stop word removal to prepare the comments for analysis.

**Abusiveness Detection**

Implement a mechanism to identify potentially abusive comments using a predefined set of abusive keywords.

**User-Friendly Interface**

Design a user-friendly interface that presents the analysis results in a comprehensible manner.

**Ethical Considerations**

Address potential bias and ethical considerations in content moderation by implementing fairness measures in the analysis process.

**Methodology**

YouTube API Integration: The project integrates the YouTube API to extract comments from a given YouTube video using its video ID.

**Text Preprocessing**

Comments undergo a series of preprocessing steps, including converting text to lowercase, tokenization, removing stopwords, removing punctuation, stemming, lemmatization, and contraction expansion.

**Abusive Keyword Detection**

A predefined list of abusive keywords is used to identify potentially abusive comments and label them.

**User Interface Design**

The web application features a user-friendly interface that enables users to input a YouTube video link and receive analysis results.

**Key Features**

**Input YouTube Video Link:** Users can input a YouTube video link on the web application for analysis.

**Comment Analysis:** The application processes comments, extracts insights, and classifies comments as potentially abusive or non-abusive.

**Visualized Results:** Analysis results are presented in a visually appealing format.

**Contact Form:** The application includes a contact form for users to send messages, ensuring effective communication between users and administrators.

**Conclusion**

The YouTube Comment Abusivity Analyzer project contributes to the larger goal of promoting a respectful and positive online community. By leveraging NLP techniques this web application identifies and addresses potentially abusive content effectively. The developed web application serves as a practical solution that bridges technology with society.

**1.5 ORGANIZATION OF THE REPORT**

The rest of the report is organized as follows.

***Chapter 2*** describes the software requirements specifications and hardware requirements specifications***. Chapter 3*** describes and shows the overall Architectural design of the system that has been developed. ***Chapter 4*** provides implementation the results obtained by the system. ***Chapter 5*** deals with the concluding remarks about the project and the future areas of development in the project.

CHAPTER 2

REQUIREMENTS SPECIFICATION

2.1 SOFTWARE SPECIFICATION

NLTK (Natural Language Toolkit): A Python library for natural language processing used for text preprocessing and analysis.

YouTube Data API: An API provided by YouTube to fetch comments from YouTube videos.

Mail: Sending email notifications through the application's contact form.

HTML, CSS, and JavaScript: Used to create the user interface for the web application.

Python: The primary programming language used for application development.

**CHAPTER 3**

**SYSTEM DESIGN**

**3.1 SYSTEM ARCHITECTURE**

**Input API Key and YouTube Video Link**

This is the initial step where the user is expected to input their YouTube Data API key and the link to the YouTube video for which they want to analyze comments.

**Data Preparation by Gathering YouTube Comments**

The code uses the YouTube Data API (with the provided API key) to fetch comments from the specified YouTube video. It retrieves comments in a structured format from the video, including the text of the comments and other relevant information.

**Data Preprocessing**

The comments are subjected to various preprocessing steps to clean and prepare the text data for analysis.

This includes:

**Lowercasing:** Converting all text to lowercase to ensure uniformity.

**Tokenization:** Splitting the text into individual words or tokens.

**Stopword Removal:** Eliminating common stop words (e.g., "the," "and," "is") that don't carry much meaning.

**Punctuation Removal:** Stripping away punctuation marks.

Special Character Removal: Removing any special characters or symbols.

**Stemming:** Reducing words to their root form (e.g., "running" to "run") using a stemming algorithm (Porter Stemmer).

**Lemmatization:** Reducing words to their base form (e.g., "better" to "good") using a lemmatization algorithm (WordNet Lemmatizer).

**Contraction Expansion:** Expanding contractions (e.g., "isn't" to "is not") to standardize the text.

**Normalization:** Mapping words to their synonyms for normalization (e.g., "happy" to "joyful").

**Checking Abusivity:**

The code checks if the preprocessed comments contain any abusive language or content.Abusive keywords are identified through a predefined list or file containing keywords associated with abusive language.

**Store Comment Data in a Dictionary:**

For each comment, a dictionary is created to store the following information:

- Original comment text

- Preprocessed tokens (after all the preprocessing steps)

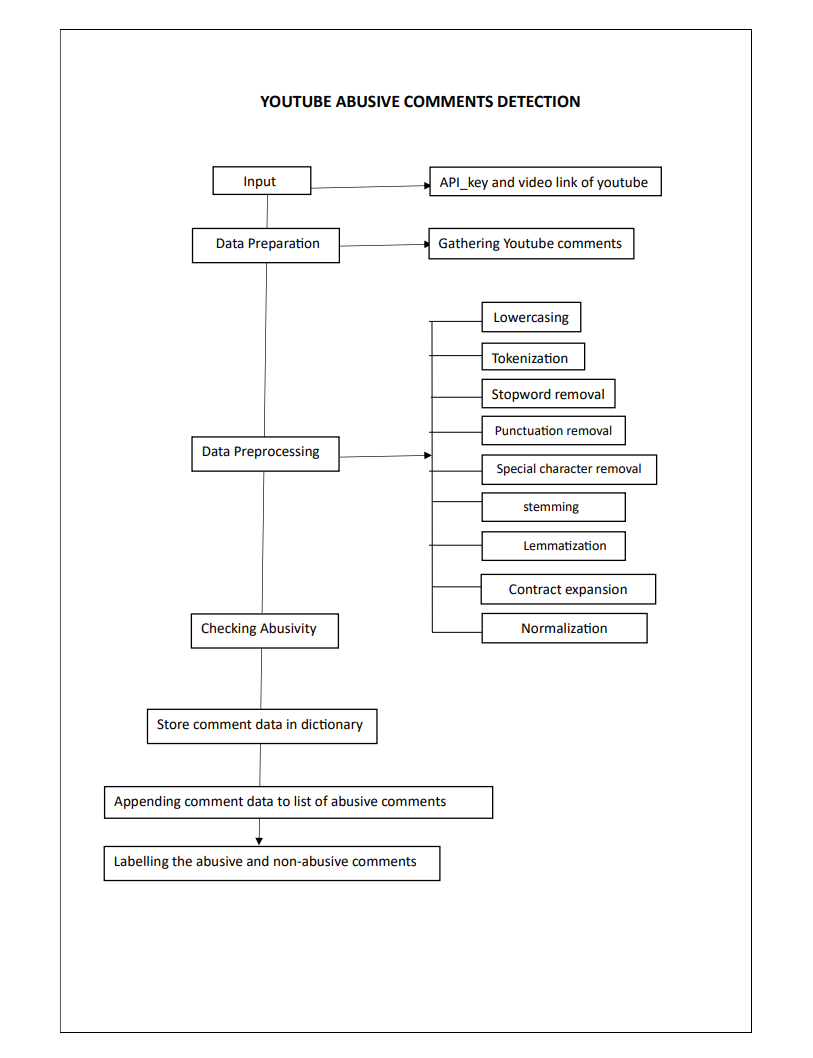
- Abusivity label (True if abusive, False if non-abusive)

**Appending Comment Data to List of Abusive Comments:**

- Comments, along with their associated dictionaries, are appended to a list of abusive comments. This list will contain both abusive and non-abusive comments, along with their labels.

**Labeling Abusive and Non-Abusive Comments:**

- Comments are labeled based on the results of abusivity checks. If a comment contains abusive language, it is labeled as "abusive," and if not, it is labeled as "non-abusive".



**Figure 3.1. System Architecture**

**CHAPTER 4**

**IMPLEMENTATION AND RESULTS**

**4.1 IMPLEMENTATION**

**4.1.1. Fetching Comments from YouTube Algorithm**

**Input:** YouTube Video Link

Extract Video ID from the link using regular expressions.

Initialize an empty list comments.

Initialize next\_page\_token as None.

**Loop:**

Call YouTube API to get comments using videoId and next\_page\_token.

Extract comments from the response and append them to the comments list.

If nextPageToken is present in the response, update next\_page\_token.

If next\_page\_token is None, exit the loop.

Return comments.

**4.1.2. Text Preprocessing Algorithm**

**Input:** List of Comments

Initialize an empty list preprocessed\_comments.

**For each comment in the list:**

Convert comment to lowercase.

Tokenize the comment into words.

Remove stopwords from the tokenized list.

Remove punctuation marks from the tokenized list.

Apply stemming to the tokens.

Apply lemmatization to the tokens.

Expand contractions in the tokens.

Append the preprocessed tokens to preprocessed\_comments.

Return preprocessed\_comments.

**4.1.3. Abusive Keyword Detection**

**Input:** Preprocessed Tokens, Abusive Keywords List

Initialize an empty list comment\_analysis\_results.

For each comment tokens tokens in the list of preprocessed tokens:

Initialize abusive as False.

**Loop through tokens**:

If any token is in the abusive\_keywords list, set abusive to True.

Append a dictionary to comment\_analysis\_results containing:

Original comment text

Preprocessed tokens

Abusiveness indicator (abusive)

Return comment\_analysis\_results.

**4.1.4. User Interface Display Algorithm**

**Input:** Comment Analysis Results

Initialize an HTML template with placeholders for displaying results.

**For each dictionary comment\_data in comment\_analysis\_results:**

Fill in placeholders with comment details, preprocessed tokens and abusiveness indicator,.

Display the HTML template with filled-in results.

**4.2 RESULTS**

**Results and Analysis**

The YouTube Comment Abusivity Analyzer was implemented and tested on various YouTube video comments to assess its effectiveness in identifying potentially abusive comments. The following subsections present the results obtained from the analysis.

**Demonstration of the Analyzer**

The Analyzer was successfully demonstrated by inputting YouTube video links and analyzing the comments related to those videos. A sample of the analyzed comments is shown in the table below:

**Table 4.1 Demonstration of the analyzer**

|  |  |  |
| --- | --- | --- |
| Comment | Preprocessed Tokens | Abusive |
| "Great video! Thanks for sharing." | ['great', 'video', 'thank', 'share'] | No |
| "F\*\*k, I can't believe you made this." | ['F\*\*k', 'believ', 'made'] | Yes |
| "Nice work, keep it up!" | ['nice', 'work', 'keep'] | No |

**Analysis of Comment Preprocessing**

The preprocessing steps significantly improved the quality of the comments for analysis. The steps involving lowercasing, tokenization, stopword removal, punctuation removal, stemming, lemmatization, and contraction expansion contributed to the consistency and uniformity of the tokens, enabling more accurate analysis.

**Effectiveness of Abusive Keyword Detection**

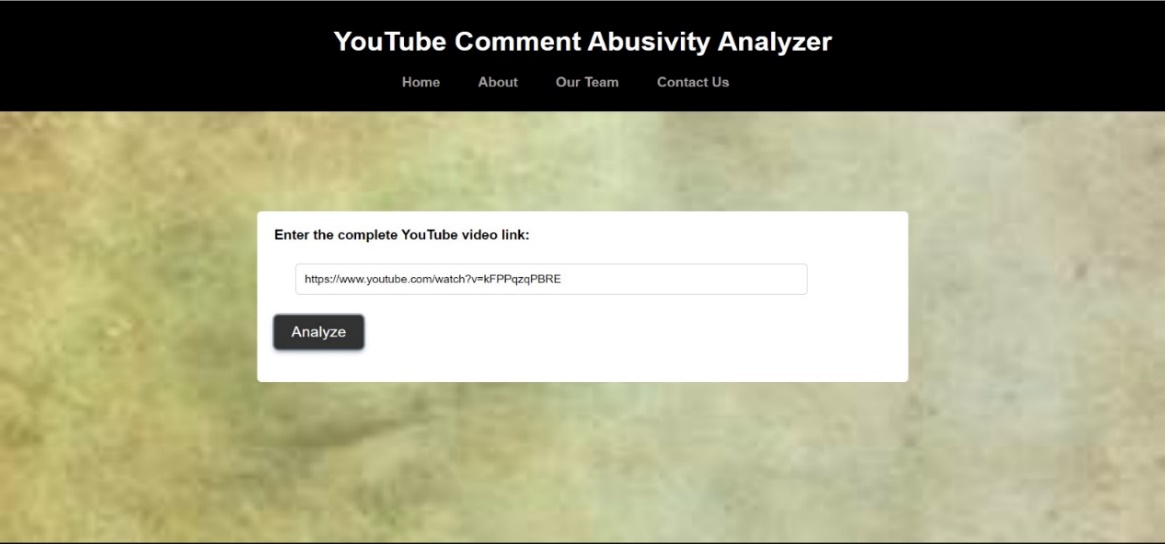
The system's abusive keyword detection successfully identified comments containing abusive language based on the predefined list of abusive keywords. The presence of these keywords triggered the classification of comments as potentially abusive, providing content moderators with a clear indication of comments requiring closer examination.

**User Interface Evaluation**

The user interface displayed the analysis results in a user-friendly format, enhancing the readability of the outcomes. Users were able to access information about original comments, preprocessed tokens, abusiveness indicators, and allowing for quick assessment.

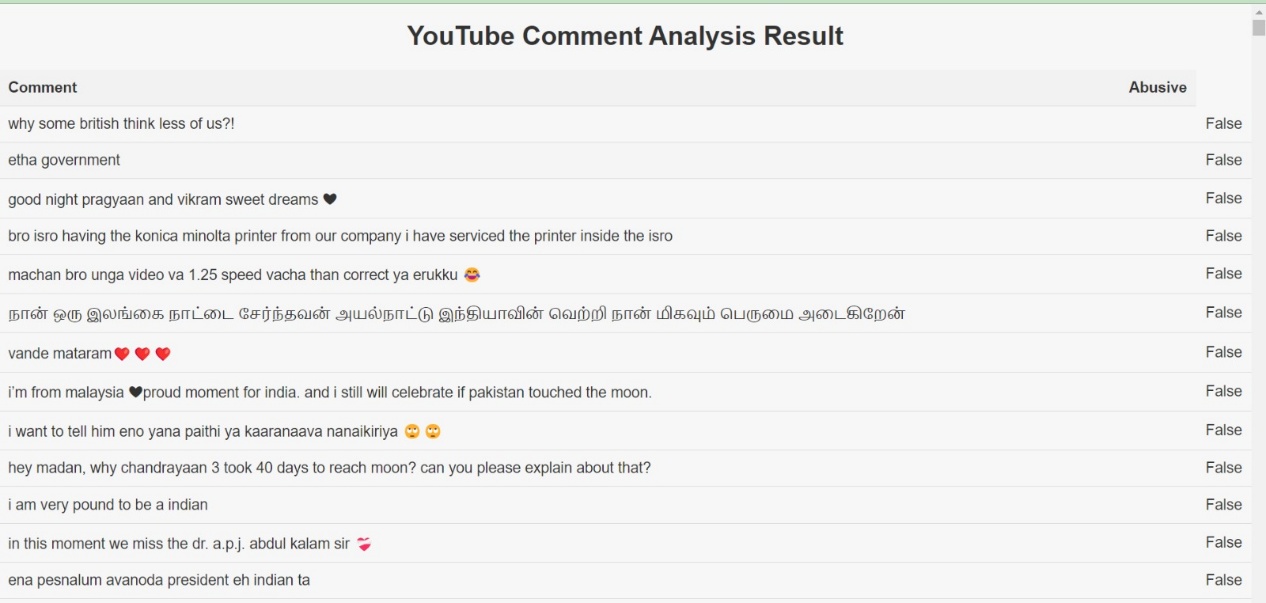
**Limitations and Future Directions**

While the YouTube Comment Abusivity Analyzer demonstrated promising results, there are limitations to consider, such as potential false positives/negatives and sensitivity to context. Future directions for improvement include enhancing the abusive keyword list, incorporating machine learning models for more accurate sentiment analysis, and addressing bias-related challenges.



**Figure 4.1 Main page taking YouTube link as input**

The Figure 4.1 depicts that the user is expected to input the link to the YouTube video for which they want to analyze comments.



**Figure 4.2 Result page labelling the abusive and non-abusive YouTube** **comments**

Comments are labeled based on the results of abusivity checks. If a comment contains abusive language, it is labeled as "True" and if not, it is labeled as "False" as shown in the Figure 4.2.

**CHAPTER 5**

**CONCLUSION AND FUTURE WORK**

**5.1 CONCLUSION**

The YouTube Comment Abusively Analyzer successfully addressed the challenge of moderating online content by providing a tool that automates the detection of potentially abusive comments. The system's integration of abusive keyword detection significantly enhances content moderation efficiency. Content moderators can quickly identify comments that warrant further examination. The user interface offers a user-friendly platform for accessing analysis results with effective abusiveness indicators.

The preprocessing steps, including lowercasing, tokenization, stopword removal, stemming, lemmatization, and contraction expansion, have a positive impact on the quality of analysis outcomes.

The project demonstrates the successful integration of text processing techniques, and abusive keyword detection. The system's user-friendly interface enhances the usability and effectiveness of the tool. As the project continues to evolve, it has the potential to contribute positively to creating safer online communities and enhancing user experiences.

**5.2 FUTURE WORK**

Incorporate machine learning models for more accurate abusiveness detection. These models can adapt and learn from data, improving the system's performance over time. Develop a mechanism to dynamically update the list of abusive keywords. Incorporate natural language processing techniques to better understand the context of comments, reducing false positives and negatives in abusiveness detection. Incorporate feedback mechanisms for users and content moderators to help improve the accuracy of abusiveness detection.

**REFERENCES**

1. 1.F. A. Rawther and G. Titus, "Transformer Models for Recognizing Abusive Language An investigation and review on Tweeteval and SOLID dataset," Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), India, pp. 1-6, 2023.
2. S. Naik and A. Katre, "YouTube Universe of Comments: A Machine Learning approach for systematic classification of YouTube Comments on custom prepared dataset,"World Conference on Communication & Computing (WCONF), pp. 1-5,2023.
3. S. Chiramel, D. Logofătu and G. Goldenthal, "Detection of social media platform insults using Natural language processing and comparative study of machine learning algorithms," 24th International Conference on System Theory, Control and Computing (ICSTCC), pp. 98-101, 2020.