A Report

on

### BUSPE (TICKET ONLINE PAYMENT)

### Submitted in partial fulfillment of the Academic Requirement for the Award of Degree of

**BACHELOR OF TECHNOLOGY**

in

## COMPUTER SCIENCE AND ENGINEERING

Submitted By

**ESHWAR BITLA 22R01A0505**

**B.SWATHI REDDY 22R01A0506**

**B.SHIVA SRI 22R01A0507**

**D.ROHIT REDDY 22R01A0512**

Under the esteemed guidance of

## Dr. S. Alagumuthukrishnan B.E., M.E.,Ph.D.

(Associate Professor, Dept. of CSE)



## CMR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMUS)

Approved by AICTE, Permanent Affiliation to JNTUH, Accredited by NBA and NAAC

### Kandlakoya(V), Medchal Dist – 501401 [www.cmrithyderabad.edu.in](http://www.cmrithyderabad.edu.in/)

**2023 - 24**

# ACKNOWLEDGEMENT

We are extremely grateful to Director, Principal and Head of Department, Department of Computer Science and Engineering, CMR Institute of Technology for their inspiration and valuable guidance during entire duration.

We are extremely thankful to our guide **Dr. S. Alagumuthukrishnan**, Associate Professor, Department of Computer Science and Engineering, CMR Institute of Technology for his constant guidance, encouragement and moral support throughout the project.

We express our thanks to all staff members and friends for all the help and coordination extended in bringing out this Project successfully in time.

Finally, we are very much thankful to our parents and relatives who guided directly or indirectly for successful completion of the project.

**ESHWAR BITLA 22R01A0505**

1. **SWATHI REDDY 22R01A0506**

**B.SHIVA SRI 22R01A0507**

**D.ROHIT REDDY 22R01A0512**

i

# ABSTRACT

**In recent years, social media platforms have emerged as popular spaces for communication, information dissemination, and social interaction. Among these platforms, Facebook is one of the most popular and influential, serving as a hub for diverse communities around the world. However, the prevalence of hate speech on Facebook has raised serious concerns about its effects on individuals and societies. Addressing this issue requires advanced techniques for detecting and effectively managing hate speech content.**

**This project proposes a comprehensive framework for detecting and integrating unstructured data containing hate speech from Facebook. The framework uses sentiment and emotion analysis techniques to accurately identify and classify hate speech content. The framework aims to improve hate speech detection efficiency and accuracy by combining Natural Language Processing (NLP) algorithms and machine learning models.**

**The first phase of the framework deals data collection and preprocessing. Facebook provides a large amount of unstructured textual data, such as posts, comments, and messages, that must be thoroughly preprocessed in order to extract relevant information. To prepare the data for further analysis, the text is normalized, tokenized, and noise is removed.**

**When it comes to this project, the existing system uses automatic tools and manual methods to detect the comments that spread hate speech. Existing system lacs in handling the unstructured data and is not applicable to other social media platforms. Some improvement is needed in results. The tests do not always reflect the true identity of the person . The data used by the researchers could be strengthened by an analysis.**

**Proposed system uses a framework called FADOHS, which uses sentiment and emotional analysis with k-means clustering algorithm i.e unsupervised algorithm to recognize all comments of hate on social media. Proposed system is better than the existing system in various aspects such as f1 scores , recall and precision . According to the experimental results, the proposed Framework FADHOS framework outperforms in terms of precision, recall and F1 scores by about 10%.**

**TABLE OF CONTENTS**

Chapter No. Particulars Page No.

**Acknowledgement iii**

**Abstract iv**

**Table of contents v**

**List of figures vi**

**List of table vii**

**1 Introduction 1**

**2 Literature Survey /Related Work 10**

**3. System Analysis**

**4. Existing System**

**4.1.Disadvantages of Existing System**

**5. Proposed System**

**5.1.Advantages of Proposed System**

**5.1.2.Limitations of Advanced System**

**6. System Study**

**7. Hardware and Software requirements**

**7.1.Hardware Requirements**

**7.2.Software Requirements**

**8. Modules**

**9. Implementation**

**10. Screen Shots**

**11. Testing**

**12. Conclusion**

**13. References**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Particulars** | **Page no.** |
| 6.1 | Architecture diagram | 16 |
| 8.1 | Data flow diagram | 21 |
| 8.2 | Class diagram | 23 |
| 8.3 | Sequence diagram | 25 |
| 8.4 | Activity Diagram | 27 |
| 9.1 | Flow chart | 29 |
| 10.1 | Dataset | 31 |
| 10.2 | Result | 31 |
| 10.3 | Admin | 32 |
| 10.4 | Local host | 32 |
| 10.5 | Command Prompt | 33 |
| 10.6 | User Interface |  |

# LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Particulars** | **Page no.** |
| 2.1 | Comparative analysis of FADHOS with respect to some previous work | 5 |

# INTRODUCTION

In the digital age, social media platforms such as Facebook have transformed how people communicate, share information, and connect on a global scale. However, in addition to its numerous benefits, social media presents significant challenges, the most pressing of which is the spread of hate speech. Hate speech, defined as any communication that disparages a person or a group based on characteristics such as race, religion, ethnicity, gender, or sexual orientation, not only violates the principles of free expression, but also endangers individuals' well-being and societal harmony.

Facebook, as one of the largest and most influential social media platforms, has been at the center of discussions about hate speech online. Despite efforts to combat it, hate speech continues to spread across the platform in various forms, such as posts, comments, and messages. Addressing this issue requires novel approaches that can detect and mitigate hate speech content while upholding the principles of free speech and user privacy.

The proposed major project aims to create a comprehensive framework for detecting and integrating unstructured data containing hate speech from Facebook. Unstructured data, such as textual content, lacks a predefined data model, making it more difficult to analyze and process. Using advanced sentiment and emotion analysis techniques, the framework aims to improve the efficiency and accuracy of hate speech detection on Facebook.

The significance of this project stems from its ability to mitigate the negative effects of hate speech on individuals, communities, and societies. Hate speech not only promotes intolerance and discrimination, but it also contributes to actual violence and conflict. Social media platforms, such as Facebook, can create safer and more inclusive online environments for users from diverse backgrounds by proactively identifying and responding to hate speech.

The proposed framework's objectives are multifaceted. First and foremost, it seeks to create robust sentiment and emotion analysis algorithms capable of accurately detecting hateful sentiments expressed in unstructured textual data.

Sentiment analysis identifies the emotional polarity (positive, negative, or neutral) of text, whereas emotion analysis recognizes specific emotional states such as anger, disgust, or hostility. By combining these analyses, the framework is able to capture nuanced expressions of hate speech and provide insights into the emotional context of the material.

Second, the framework aims to use machine learning techniques to classify textual instances as hate speech or non-hate speech. Machine learning models, trained on annotated datasets, learn to recognize patterns and characteristics associated with hate speech, automating the detection process. By incorporating sentiment and emotion features in addition to traditional linguistic features, the classification model can improve its accuracy and robustness when identifying hate speech content.

Moreover, the framework aims to provide practical tools and strategies for content moderation and community management on Facebook. By flagging potentially harmful content for review and intervention, the framework empowers platform administrators and moderators to take proactive measures against hate speech while respecting users' rights to freedom of expression. In conclusion, the proposed framework represents a proactive approach to combating hate speech on Facebook, leveraging advanced techniques in sentiment and emotion analysis to detect and integrate unstructured data effectively. By fostering a safer and more inclusive online environment, the framework contributes to the overarching goal of promoting tolerance, respect, and dialogue in digital spaces.

# LITERATURE SURVEY

Ben-David and Matamoros-Fernandez's related study on overt hatred and covert disrespectful practices on the Internet [1] relies on network and multimodal analysis. It examines information and images found on social media and uses tools like Netvizz to retrieve data from various Facebook pages with hate speech-related content [3]. In our proposed framework, we analysed the dataset [4] using a Facebook graph application programming interface (API) and emotional analysis [5].

In a similar study, the authors used the valence-aware Dictionary and Sentiment Reasoner (VADER) tool as a simple rule-based model for general sentiment analysis [6]. The VADER tool analyses and validates data using both qualitative and quantitative methods [7–19]. Following that, data validation is tailored to sentiments via microblog-like information. We also used the VADER tool for sentiment analysis (SA). Unlike the research in [6], we used the JAMMIN tool to conduct emotional analysis experiments, specifically tracking posts with negative comments. The study [2] developed a classification of abhorrence based on various "loathing levels." The authors used morpho-grammatical highlights, notion extremity, and word-installed dictionaries to design and implement two classifiers for the Italian language. In addition, they used SVM and LSTM [2].

However, our approach is intended to detect hate speech on Facebook, particularly "unmistakable" manifestations of hatred expressed as remarks on divisive topics (e.g., immigration, religion, and race). The study in [20] highlights future issues that Facebook and Twitter may face when identifying hate speech on their respective platforms. Their tool was crowdsourcing. Although their framework has yet to be fully evaluated, their research includes a quality-of-service (QoS) assessment for platform providers. They created an easy-to-use tool for combating hate speech. However, while their tool can detect information that does not comply with QoS policies, we believe this format is inefficient due to the use of Python programming tools [20]. In this study, we used a procedure to filter hate-filled posts and comments on social media.

The related research defines "platform racism" as an emerging form of racial prejudice emanating from social media pages [21].

Hatred is a form of discrimination that depends on the culture associated with a specific group. An annotation of the study that proposed a possible algorithm for compiling the contents was provided. Although the experiment revealed important trends relevant to our specific research question, we focused on Facebook as a platform and used both data extraction and experimental setup as specified in the seed pages. Online trends with offline consequences [22] investigated the link between social media and hate crimes using Facebook data.

Interestingly, it was concluded that social media is frequently used as a means of spreading hatred [22]. Although such a study is important, we believe that the data used by researchers could be strengthened by analyses like ours, which use social media analytics to target and identify negative comments posted on specific hate-promoting Facebook pages [4]. In contrast, in this linked study [23], the authors looked at the most effective ways to detect hate speech in written text. Based on this survey and our framework, we ran a number of tests to compare the accuracy of the top three methods. The authors of [24] suggested an optimization strategy based on metaheuristic searching.

The ant lion optimization (ALO) and moth flame optimization (MFO) algorithms were created to address the HSD problem. This is the first attempt to use optimization algorithms as solution-search strategies for automated HSDs. This was accomplished through the development of an efficient representation scheme and a flexible fitness function. However, the FADHOS approach not only identifies unstructured data from Facebook allegedly promoting hate speech, such as commonly discussed topics, but it also uses sentiment and emotion analysis to identify and integrate them by topic in clusters. In 2019, OpenAI released GPT-2 models [25]-​[26].

These were built with transformer decoder blocks. We investigated the best dataset in our framework (moderate level of hate speech dataset) and conducted several experiments using OpenAI's Nobel model, the GPT-2 model [25]. The primary goal of this experiment was to determine the extent to which our hate speech dataset can improve the performance of the GPT-2 model [37]. This section's literature review serves as both a foundation for this study and a potential motivation for future research.

Our primary research goal is to accurately and efficiently locate social media pages that discuss sensitive topics, as well as to develop a dependable system that categorizes posts and incorporates unstructured information with frequently discussed themes that intentionally or unintentionally spread hate speech. Table 2.1 presents a detailed comparison of these studies.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Approaches | Categories | Purpose | Features | Algorithms | Characteristics | |
| Advantages | Limitations |
| (Ben-David and Anat & Matamoros-Fernandez, 2016) | Facebook | Posts with discrimination from Spanish extreme-right political parties. | Word frequency and co-occurrence to identify patterns in texts. | Automatic and manual methods to determine common topics from the texts. | It supports longitudinal multimodal content and network analysis of data from extreme political parties from Facebook | It cannot be readily applied to other social networking platforms |
| (DelVigna, Cimomo, Dell’Orletta, Petrocchi, & Tesconi, 2017) | Text Italian Language | Addressing proposed categories to distinguish various kinds of hate being expressed. | It uses morpho syntactical features, sentiment polarity and word-embedded lexicons. | Two classifiers with SVM and LSTM. | The effectiveness of the two classifiers over the first manually annotated Italian Hate Speech Corpus of social media text. | They need to refine the results of their classifier when considering distinction among hate levels. |
| (Chopra, A., Dimri, A., & Rawat, S. 2019) | Twitter | Predictive model for the popularity of a news article on a particular social media platform, based on the input features. | They compared the effectiveness of various classification as prediction techniques. | They used  \*Logistic Regression.  XGBoost.  \*VADER | \*XGBoost provides the maximum accuracy and the appropriate model for the dataset and the problem statement. | Limitation of large datasets with more features. |
| (Miller, K., & Schwarz, C., 2021) | Facebook | \*To investigate the link between social media and hate crime.  \*To explore the role of social media of hate. | \*To analyze the effect of Facebook and internet outages in a flexible empirical framework. | \*Facebook API  \*They developed an identification strategy based on Facebook and internet outages. | They take a first step towards identifying potential harm arising from extended social media usage. | The data used by the researchers could be strengthened by an analysis such as ours. |
| (Davidson, T., Warmsley, D., Macy, M. & Weber, I. 2017, May) | Twitter | Multi-Class Classifier to distinguish different categories | Crowdsourcing tool to weight stem unigrams bigrams, and trigrams. | L1 Logistic Regression, NB, DT, RF, SVMs. | They used Python programming so their format falls short of efficiency | Their framework has not been fully evaluated. |
| FADOHS | Facebook,  Twitter | To explore a new way to detect topics from related posts from Facebook and Twitter. | Discovery, Sensitive social data collection, sentiment and emotion analysis, and clusters stages. | \*Facebook API  K-Means unsupervised algorithm,  \*JAMMIN  \*VADER | It uses graphs, sentiment, and emotion analysis, and clustering methods to analyse posts that include hate speech. This is the most difficult type of hate speech to detect | \*We can provide some improvement to ameliorate the results. The tests do not always reflect the true identity of the person.\*The information on social media changes frequently and constantly. |

Table 2.1: Comparative analysis of FADHOS with respect to some previous work.

# SYSTEM ANALYSIS

System analysis is a critical phase in the project management and development process. It provides a solid foundation for project planning, design, development, and implementation, helping to ensure that the final system meets the intended objectives and user needs. This analysis aims to gather detailed information about the existing system and its disadvantages, proposed system and its attributes like its advantages and advancements.

## 3.1 EXISTING SYSTEM

The main requirement in hate speech detection is identifying the negative comments before they can exploit the Facebook by initiating offensive speech. The Existing System have numerous methods that claim to mitigate the hate speech against . These approaches focus on detection of negative comments under the Facebook posts using manual methods and Automatic tools.

The Existing System have three well-known machine learning algorithms to detect hate speech on Facebook and they are SVM, LSTM and logistic Regression. The findings show that SVM performs better in detecting hate speech having 86.43 % accuracy, while other algorithms have achieved 82.62% and 85.7% accuracy, respectively.

In existing system, to identify text patterns in hate speech using sentiment and emotional analysis, authors used algorithms such as Automatic tools and manual methods. The advantage of using this algorithm is to support multi-model content and network analysis o data.

Authors developed Facebook API which is an identification strategy to analyze the effect of Facebook and internet blockage

## 3.1.1 DISADVANTAGES OF EXISTING SYSTEM

In the realm of hate speech detection, several methodologies are employed, each with its own drawbacks like:

* + 1. The information on social media changes frequently and constantly which makes difficulty to identify the unstructured data.
    2. Can't be applied to other social networking platforms except Facebook.
    3. Moreover, some improvement is needed in results. The tests do not always reflect the true identity of the person.
    4. The data used by the researchers could be strengthened by an analysis.

## PROPOSED SYSTEM

In our proposed system, To detect hate speech and negative comments on Facebook we proposed a Framework that uses sentiment and emotional analysis with k-means algorithm. In proposed system, to explore a new way to detect topics from related posts from Facebook we can use features such as discovering sensitive social data collection, sentiment and emotional analysis and cluster stages.

In proposed system we use Facebook API and k-means unsupervised algorithm which gives optimal solution to our problem of hate speech. Firstly, collect and preprocess the unstructured data from Facebook, including text data containing comments, posts, and other user-generated content. Extract relevant features from the text, such as words or phrases that may indicate hate speech. Convert the textual data into numerical vectors using techniques word embeddings (e.g., Word2Vec, GloVe). Apply the k-means algorithm to cluster the data into k groups based on the features extracted. Each cluster represents a group of data points that share similar characteristics.

The clusters can help identify patterns or trends within the unstructured data, potentially revealing different types or themes of hate speech. Analyze the content within each cluster to understand the nature of hate speech present. This step involves sentiment and emotional analysis to discern the tone and emotions expressed in the text. Identify patterns in terms of language, sentiments, and emotions associated with hate speech within each cluster.

Train sentiment analysis and emotional analysis models on labeled data, considering the clusters as labeled groups. This will help the models learn to classify text into categories such as positive/negative sentiment or different emotions. Combine the results of sentiment and emotional analysis with the cluster information obtained from k-means. This integration provides a more comprehensive understanding of hate speech, including the sentiment and emotional context associated with different types of hate speech.

Visualize the clustered data, sentiment, and emotional analysis results to provide interpretable insights. Tools like graphs, word clouds, or heatmaps can help stakeholders understand the patterns and trends in hate speech on Facebook. Implement a monitoring system to continuously analyze new data and adapt the model as the nature of hate speech evolves over time.

The findings show that inculcating k-means algorithm into sentiment and emotional analysis has performed better in detecting hate speech having 87.02 % accuracy.

## 3.2.1 ADVANTAGES OF PROPOSED SYSTEM

The proposed system for identifying and integrating hate speech using Sentiment and Emotional analyis offers several distinct advantages when compared to existing systems and traditional approaches:

1. **Multifaceted Analysis:** Sentiment and emotional analysis provide a multifaceted understanding of text data. By considering both sentiment (positive, negative, neutral) and emotional tones (anger, sadness, joy, etc.), the framework can capture a more nuanced portrayal of hate speech, improving its detection accuracy.
2. **Contextual Understanding:** Hate speech often relies on subtle linguistic cues and context. Sentiment and emotional analysis allow the framework to grasp the underlying context of messages, enabling it to differentiate between harmless expressions of opinion and harmful hate speech.
3. **Identification of Implicit Hate Speech:** Hate speech is not always explicit and may manifest through implicit language or disguised expressions. Sentiment and emotional analysis can uncover implicit hate speech by detecting negative sentiments or underlying emotions indicative of hostility, discrimination.
4. **Early Detection:** Sentiment and emotional analysis can facilitate the early detection of hate speech by identifying potentially harmful content based on its linguistic characteristics. This early detection enables timely intervention and moderation to prevent the spread of hate speech and mitigate its impact on users.
5. **Comprehensive Monitoring:** The framework can continuously monitor Facebook content for hate speech by analyzing sentiment and emotional signals in real-time. This comprehensive monitoring ensures proactive identification and handling of hate speech instances, enhancing platform safety and user experience.
6. **Adaptability to Linguistic Variability:** Hate speech can vary significantly in its linguistic form, depending on factors such as language, dialect, slang, and cultural context. Sentiment and emotional analysis techniques can be adapted to accommodate linguistic variability, making the framework applicable across diverse user populations and languages.
7. **Integration with Moderation Tools:** The framework can be seamlessly integrated with moderation tools and workflows to automate the detection and handling of hate speech on Facebook. By incorporating sentiment and emotional analysis into existing moderation pipelines, the platform can efficiently identify and address hate speech content while minimizing human intervention.
8. **Data-Driven Insights:** Sentiment and emotional analysis generate valuable data insights into the prevalence, trends, and patterns of hate speech on Facebook. By analyzing sentiment and emotional trends over time, the framework can provide actionable insights for platform administrators to develop targeted interventions and policies to combat hate speech effectively.

## 4. SYSTEM STUDY

System Study is the process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

For this project, we have gathered information and images from IEEE explore website about the detection of botnet .we have divided them into training and testing sets where training set images are used to train the machine learning model and based on the training the machine will predict the result for testing data whether the botnet is detected or not detected.

## FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

* Economical Feasibility
* Technical Feasibility
* Social Feasibility

## Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

## Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

## Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

# 5. HARDWARE AND SOFTWARE REQUIREMENTS

The hardware and software requirements outlined for the proposed system are pivotal elements that warrant careful consideration to guarantee the system's optimal functionality and high-performance capabilities. The following mentioned hardware and software requirements are those which is suggestable for the successful working and execution of the proposed system.

## HARDWARE REQUIREMENTS

In parallel, the hardware requirements have been meticulously defined to strike a balance between accessibility and computational process.

|  |  |
| --- | --- |
| Processor | Pentium-iv or intel I3 |
| RAM | 4 GB (minimum) |
| Hard Disk | 20 GB |
| Key Board | Standard Windows Keyboard |
| Mouse | Two or Three Button Mouse |
| Monitor | SVGA |

### Processor

Provides the instructions and processing power the computer needs to do its work. The more powerful and updated your processor, the faster your computer can complete its tasks.

The Processor we used for this project is: Intel core i3/Pentium-IV

### RAM

RAM provides the shorter-term memory the CPU needs to open files and move data around as it responds to the tasks given to it by your apps. Both RAM and the CPU work synchronously and complementarity to ensure that your computer's performance fits your needs and you have a good experience when using your device. The RAM used for this project is 4 GB minimum.

### Hard disk

A hard drive is the hardware component that stores all of your digital content. Your documents, pictures, music, videos, programs, application preferences, and operating system represent digital content stored on a hard drive. A minimum of 500GB hard disk is suggestable.

The Hard disk used for this project is 20GB.

## SOFTWARE REQUIREMENTS

On the software front, A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development.

|  |  |
| --- | --- |
| Operating System | Windows 10 |
| Coding Language | Python |
| Front-End | Python |
| Back-End | Django-ORM |
| Designing | Html, CSS, JavaScript |
| Data Base | MYSQL (WAMP Server) |

### Operating system

The operating system (OS) manages all of the software and hardware on the computer. It performs basic tasks such as file, memory and process management, handling input and output, and controlling peripheral devices such as disk drives and printers.

The operating system we have used is Windows 10.

### Coding language

The programming language used for this project is python. It brings an exceptional amount of power and versatility to machine learning environments. The language's simple syntax simplifies data validation and streamlines the scraping, processing, refining, cleaning, arranging and analyzing processes, thereby making collaboration with other programmers less

of an obstacle.

The coding language we have used is Python with the latest version of 3.7.0

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English words frequently whereas other languages use punctuation, and it has fewer syntactic constructions than other languages.

* + - 1. Python is Interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
      2. Python is Interactive: You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
      3. Python is Object-Oriented**:** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
    1. **Frame Works**

Django ORM (Object Relational Mapper) is a powerful tool within the Django web framework that simplifies interacting with relational databases. Here's a breakdown of its key functionalities:

* **Mapping Objects to Database Tables:** Django ORM acts as a bridge between your Python objects (models) and the underlying relational database tables. It automatically creates and manages database tables based on the structure you define in your models.
* **Data Manipulation:** Django ORM provides a convenient and Pythonic way to perform CRUD (Create, Read, Update, Delete) operations on your database. You can write queries using a syntax similar to working with Python objects, eliminating the need to write raw SQL statements.
* **Database Abstraction:** Django ORM acts as a layer of abstraction between your Python code and the specific database you're using. This allows you to switch between different relational databases (like MySQL, PostgreSQL, SQLite) without modifying your application logic, as Django ORM handles the communication with the specific database dialect.
  + 1. **DataBase**

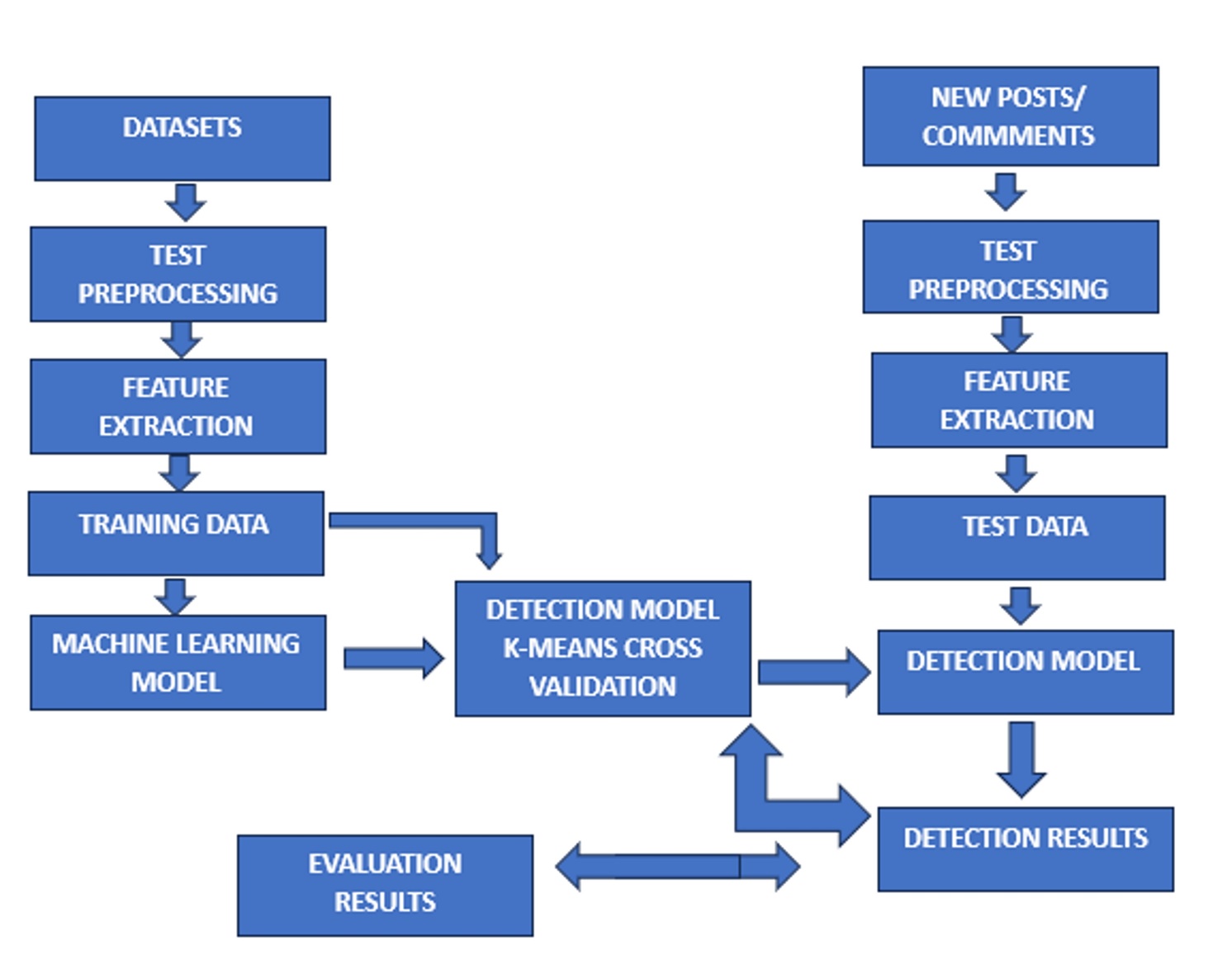
**MySQL:** MySQL is a popular open-source relational database management system (RDBMS). An RDBMS is a type of database that stores data in structured tables with rows and columns. MySQL is known for its speed, reliability, and ease of use. It's a popular choice for web applications and online services.

**WampServer**: WampServer is a free software package that includes Apache, MySQL, and PHP. Apache is a web server software that processes incoming requests from users and delivers web content. PHP is a server-side scripting language that is often used to create dynamic web content.

**Combining MySQL and WampServer:** WampServer bundles MySQL along with Apache, which simplifies the setup process for creating a development environment for web applications. Developers can use this environment to test and run their applications before deploying them to a live server.

# ARCHITECTURE

Architecture typically refers to the structural and organizational framework or design of a system, software, building, or any complex project. The architecture provides a high-level overview of how the project is structured and how its various components or elements interact with each other. It outlines the fundamental design principles, components, and their relationships, and helps stakeholders understand the project's overall framework.



**Figure 6.1 Architecture Diagram**

# The flow chart represents the process of building and evaluating a machine learning model for text classification or detection. Here's a detailed explanation of each step:

# 1. Datasets: This represents the initial data sources or corpora that will be used for training and testing the machine learning model.

# 2. Text Preprocessing (for Datasets): The raw text data from the datasets goes through preprocessing steps, which may include tasks like tokenization, stemming, lemmatization, removing stop words, and other cleaning operations to prepare the text for feature extraction.

# 3. Feature Extraction (for Datasets): In this step, the preprocessed text data is transformed into numerical feature vectors that can be fed into the machine learning model. Common techniques for text feature extraction include bag-of-words, TF-IDF, word embeddings, or more advanced methods like BERT or GPT.

# 4. Training Data: After feature extraction, the transformed data is split into training and test sets. The training data will be used to train the machine learning model.

# 5. Machine Learning Modeling: This step involves selecting and training a suitable machine learning algorithm (e.g., logistic regression, naive Bayes, support vector machines, or deep learning models) on the training data for the text classification or detection task.

# 6. Detection Model K-fold Cross Validation: Cross-validation is a technique used to evaluate the performance of the trained model on different subsets of the training data. K-fold cross-validation splits the training data into K equal parts, trains the model on K-1 parts, and evaluates it on the remaining part. This process is repeated K times, and the results are averaged to estimate the model's performance.

# 7. Evaluation Results: The cross-validation step provides evaluation metrics or results, such as accuracy, precision, recall, F1-score, or other relevant measures, which can be used to assess the model's performance and potentially tune hyperparameters or refine the model.

# The right side of the flow chart represents the process for new or unseen data:

# 8. New Posts/Comments: This represents new text data, such as posts or comments, that need to be classified or analyzed by the trained model.

# 9. Text Preprocessing (for New Data): Similar to the preprocessing step for the datasets, the new text data goes through the same preprocessing steps to prepare it for feature extraction.

# 10. Feature Extraction (for New Data): The preprocessed new text data is transformed into feature vectors using the same feature extraction techniques as before.

# 11. Test Data: The transformed new data serves as the test data for the trained model.

# 12. Detection Model: The trained and validated machine learning model from the left side of the flow is now used to make predictions or classifications on the test data (new posts/comments).

# 13. Detection Results: Finally, the model outputs the detection or classification results for the new text data, which could be labels, scores, or any other relevant output format.

# This flow chart illustrates the end-to-end process of training a machine learning model for text classification or detection tasks, as well as applying the trained model to new, unseen data for making predictions or detections.

# 7. MODULES

### 7.1 Service Provider

In the Service Provider module, the primary functionality revolves around user authentication and various operations that can be performed post-login. The initial step involves the Service Provider logging in using valid credentials, namely a username and password. Upon successful authentication, the Service Provider gains access to a range of functionalities.

1. **Login:** After successful authentication, the Service Provider gains access to their account, enabling them to utilize the platform's features and functionalities.
2. **Browse URLs Datasets and Train & Test Data Sets:** Once logged in, the Service Provider can browse through URL datasets and perform operations related to training and testing datasets. This functionality allows them to access and manipulate data for further analysis and processing.
3. **View URLs Datasets Trained and Tested Accuracy in Bar Chart:** The Service Provider has the capability to visualize the accuracy of trained and tested datasets using a bar chart representation. This feature provides an intuitive way to interpret the performance of the system based on accuracy metrics.
4. **View URLs Datasets Trained and Tested Accuracy Results:** Alongside the visual representation provided by the bar chart, the Service Provider can also view detailed accuracy results of trained and tested datasets. This allows for a more granular analysis of the system's performance.
5. **View Prediction Of URLs Type**: The module enables the Service Provider to view predictions related to URL types. This feature facilitates understanding the categorization or classification of URLs based on predefined criteria or parameters.
6. **View URLs Ratio**: The Service Provider can access information regarding the ratio of

different types of URLs present in the datasets. This insight aids in understanding the distribution and composition of URLs within the dataset.

1. **Download Predicted Data Sets:** The module allows the Service Provider to download predicted datasets, which contain information regarding predicted URL types. This functionality enables further analysis or integration of the predicted data into external systems or applications.
2. **View URLs Type Ratio Results:** Similar to viewing the overall URL ratio, the Service Provider can access detailed results regarding the distribution of URL types. This information provides deeper insights into the composition and classification of URLs within the datasets.
3. **View All Remote Users:** Lastly, the Service Provider has the ability to view details of all remote users registered within the system. This feature provides visibility into the user base and facilitates management and interaction with remote users as necessary.

### View and Authorize Users

In the View and Authorize Users module, the focus shifts to administrative tasks related to user management and authorization.

1. **View List of Users:** The module enables the admin to view a comprehensive list of all registered users within the system. This functionality provides visibility into the user base and facilitates administrative tasks such as user management and support.
2. **View User Details:** In addition to viewing the list of users, the admin can access detailed information about each user, including their username, email, address, and other relevant details. This feature allows for a more comprehensive understanding of individual user profiles.
3. **Authorize Users:** The admin has the authority to authorize users within the system. This process involves granting permissions or access rights to users based on their roles or responsibilities. By authorizing users, the admin can control access to specific features or functionalities within the platform.

### Remote User

The Remote User module caters to the needs and interactions of individual users registered within the system.

**1.User Registration:** Users are required to register before accessing any functionalities within the system. During registration, user details are collected and stored in the database for future reference.

1. **Login**: Upon successful registration, users can log in using their authorized username and password. The login process grants access to various features and functionalities available to registered users.

#### **Operations Available to Users:**

* + - 1. **REGISTER AND LOGIN:** Users can register for an account and subsequently log in to access the system's functionalities.
      2. **PREDICT THREAT DETECTION:** Registered users have the capability to predict threat detection based on predefined algorithms or models.
      3. **VIEW YOUR PROFILE:** Users can view and manage their profile information, including personal details and preferences, within the system.

8. DIAGRAMS

## DATA FLOW DIAGRAM

Login, Browse Datasets and Train & Test Data Sets, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Results

predict sentiment and emotion analysis,

Login

Service Provider

System

View Sentiment and Emotion Analysis Status, View Sentiment and Emotion Analysis Status Ratio,

Response

Register and Login with the system

view your profile

Request

Download Predicted Data Sets, View Sentiment and Emotion Analysis Ratio Results, View All Remote Users.

Remote User

Tweet Server

Tweet Server

Tweet Server

Tweet Server

Tweet ServerTweet Server

Tweet Server

**Figure 8.1 Data Flow Diagram**

A Data Flow Diagram (DFD) is a visual representation that illustrates how data moves within a system.

The service provider can log in, browse datasets, train and test datasets, and view the trained and tested accuracy in a bar chart. Remote users can register and log in to the system, request URL type prediction, view their profile, and access various functionalities like viewing URL datasets trained and tested accuracy results, URL type ratios, downloading predicted datasets, and viewing all remote users. The system facilitates these interactions and processes the requests between the service provider and remote users.

## CLASS DIAGRAMS

Methods

Members

**Service Provider**

Login, Browse Datasets and Train & Test Data Sets, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Results, View Sentiment and Emotion Analysis Status, View Sentiment and Emotion Analysis Status Ratio, Download Predicted Data Sets, View Sentiment and Emotion Analysis Ratio Results, View All Remote Users.

Post \_id, post, Prediction.

**Login**

**Register**

Login (), Reset (), Register ().

User Name, Password.

Methods

Methods

Register (), Reset ()

User Name, Password, E-mail, Mobile, Address, DOB, Gender, Pin code, Image

Members

Members

**Remote User**

Tweet Servervvv

Tweet Server

Tweet Server

Tweet Server

Tweet Server

REGISTER AND LOGIN, PREDICT SENTIMENT AND EMOTION ANALYSIS, VIEW YOUR PROFILE.

Post \_id, post, Prediction.

Methods

Members

**Figure 8.2 Class Diagram**

A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

The diagram shows the relationships and interactions between these classes, such as the Service Provider allowing login and registration, and the Remote User being able to perform various tasks like registering, logging in, predicting URL types, and viewing their profile. The main entities are:

1. **Service Provider:** This class has members (properties/data fields) representing various functionalities for URL datasets, training, testing, prediction, and analysis. The methods include URL prediction and detecting malicious URLs.
2. **Login**: A class for user login functionality, with members like username and password.
3. **Register:** A class for user registration, with members like username, email, gender, country, city, address, mobile number, and state.
4. **Remote User:** This class represents the end-users of the system. Its members include "REGISTER AND LOGIN", "PREDICT URL TYPE", and "VIEW YOUR PROFILE". The methods are the same as the Service Provider class, indicating shared functionality for URL prediction and malicious URL detection.

## SEQUENCE DIAGRAMS

Service Provider

Remote User

Web Server



Register and login

Predict Sentiment And Emotion Analysis,

View Your Profile

Login,

Browse Datasets and Train & Test Data Sets,

View Trained and Tested Accuracy in Bar Chart,

View Trained and Tested Accuracy Results,

View Sentiment and Emotion Analysis Status,

View Sentiment and Emotion Analysis Status Ratio,

Download Predicted Data Sets,

View Sentiment and Emotion Analysis Ratio Results,

View All Remote Users.

**Fig 8.3 Sequence Diagram**

**Figure 8.3 Sequence Diagram**

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scena os, and timing diagrams. . It shows how objects communicate with each other to accomplish a specific task or scenario.

This sequence diagram separates the responsibilities of data management, analysis, and prediction (handled by the Service Provider) from the user interaction and prediction tasks (performed by Remote Users), with the Web Server acting as the intermediary facilitating these interactions. The key entities are:

1. **Service Provider:** This entity can log in, browse URL datasets, train and test datasets, view the trained and tested accuracy results in a bar chart or tabular form, view predictions of URL types, view URL type ratios, download predicted datasets, view URL type ratio results, and view all registered remote users.
2. **Web Server:** It serves as the central component, facilitating registration and login form remote users, allowing them to predict URL types and view their profiles.
3. **Remote User:** Remote users can register and log in to the system, predict URL types, and view their profiles.

**9. IMPLEMENTATION**

**START**

**LOGIN**

**SUCCESS?** Status

Yes No

**USERNAME & PASSWORD WRONG**

**REGISTER AND LOGIN**

**PREDICT SENTIMENT AND EMOTION ANALYSIS,**

**VIEW YOUR PROFILE**

**LOGOUT**

**Figure 9.1 Remote User Flow Chart**

1. **START**: This is the initial step where the user begins the process or launches the application.
2. **LOGIN:** After starting the application, the user is prompted to log in. This typically involves entering their credentials, such as a username and password.
3. **DECISION POINT (YES/NO):** Upon attempting to log in, the system checks the validity of the provided credentials. If the credentials are correct, the flow proceeds to the "Yes" path; otherwise, it follows the "No" path.
4. **"REGISTER AND LOGIN":** If the user's credentials are valid (following the "Yes" path), they can proceed to register or log in to the system. This step may involve creating a new account if the user is not already registered or simply logging in if they have an existing account.
5. **"PREDICT SENTIMENT AND EMOTION ANALYSIS**": After successful registration and login, the user can access a feature that allows them to predict sentiment and perform emotion analysis. This could potentially involve analyzing text, audio, or other forms of data to determine the underlying sentiment or emotional context.
6. **"VIEW YOUR PROFILE":** Another option available to the user after logging in is to view their profile. This could include personal information, settings, preferences, or any other relevant data associated with their account.
7. **"USERNAME & PASSWORD WRONG":** If the user's credentials are invalid (following the "No" path from the decision point), the system indicates that the entered username and password combination is incorrect.
8. **LOGOUT:** This is the final step in the flow, where the user can log out of the application or system, effectively ending the current session.

**START**

**LOGIN**

**SUCCESS ?**

Yes No

**Username & Password Wrong**

Login, Browse Datasets and Train & Test Data Sets,

View Trained and Tested Accuracy in Bar Chart

View Trained and Tested Accuracy Results

View Sentiment and Emotion Analysis Status,

**Log Out**

View Sentiment and Emotion Analysis Status Ratio,

View All Remote Users

View Sentiment and Emotion Analysis Ratio Results,

Download Predicted Data Sets

**Figure 9.2 Service Provider Flow Chart**

### Login, Browse URLs Datasets and Train & Test Datasets: This functionality is reserved for the service provider (administrator) and allows them to manage the training and testing data used by the probabilistic model. This includes uploading new datasets, browsing existing datasets, and potentially visualizing data distribution.

### Train & Test Datasets, View URLs Datasets Trained & Tested Accuracy in Bar Chart, View URLs Datasets Trained & Tested Accuracy Results: These steps enable the service provider to train the probabilistic model on the available datasets and evaluate its performance through metrics like accuracy. Visualizations (bar charts) can aid in understanding the model's effectiveness.

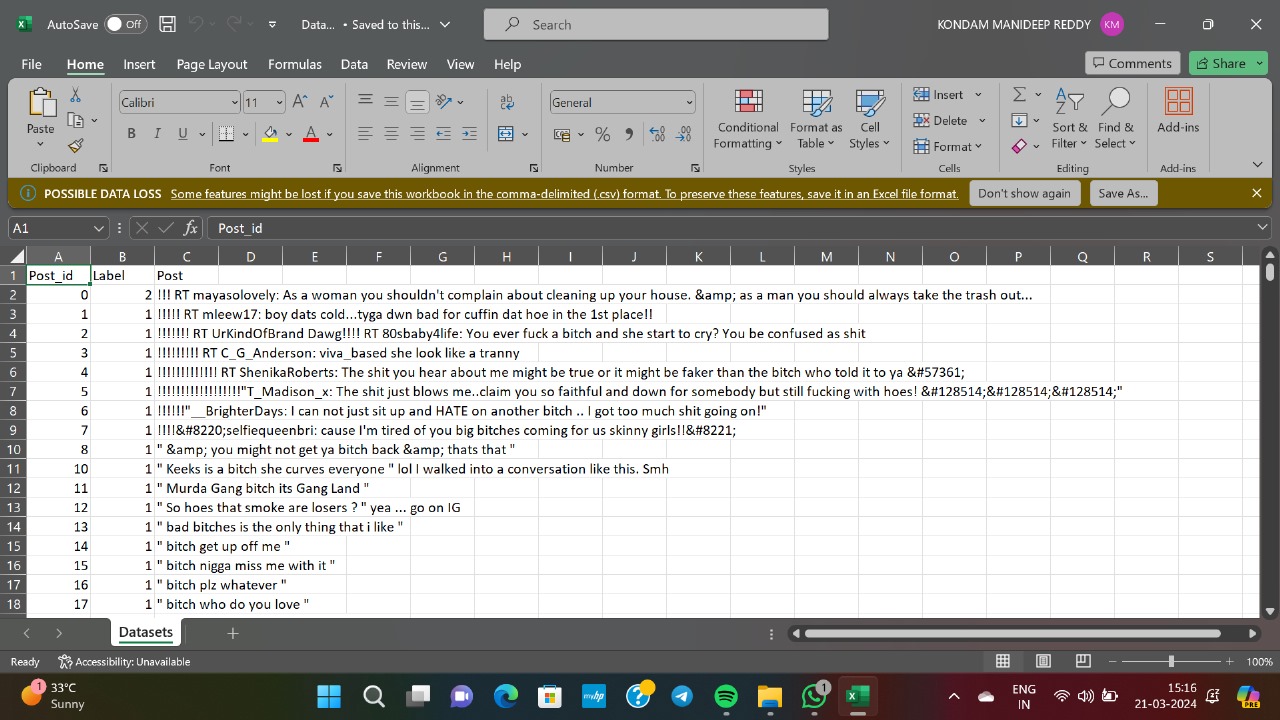
### View Prediction of URLs Type: This allows the service provider to view a sample of user-submitted URLs and their corresponding classifications (malicious or benign) made by the system. This can be used for monitoring purposes and potential model improvement.

### View URLs Type Ratio, View All Remote Users: These functionalities provide insights into user browsing behavior and overall system usage. The service provider can view the distribution of URL types encountered by users (e.g., e-commerce, news, social media) and potentially track the number of active users.

### Download Predicted Datasets: This feature might be used by the service provider to download datasets containing URLs classified as malicious. These datasets can be further analyzed to identify emerging attack patterns or refine the probabilistic model.

### View URLs Type Ratio Results: Similar to "View URLs Type Ratio," this step allows the service provider to visualize the distribution of URL types, potentially providing insights into user behavior patterns.

**10. SCREENSHOTS**



**Figure 10.1 Data Sets**

**Figure 10.2 Results**

**Figure 10.3 Admin**

**Figure 10.4 Local Host**

**Figure 10.5 Command Prompt**

**Figure 10.6 User Interface**

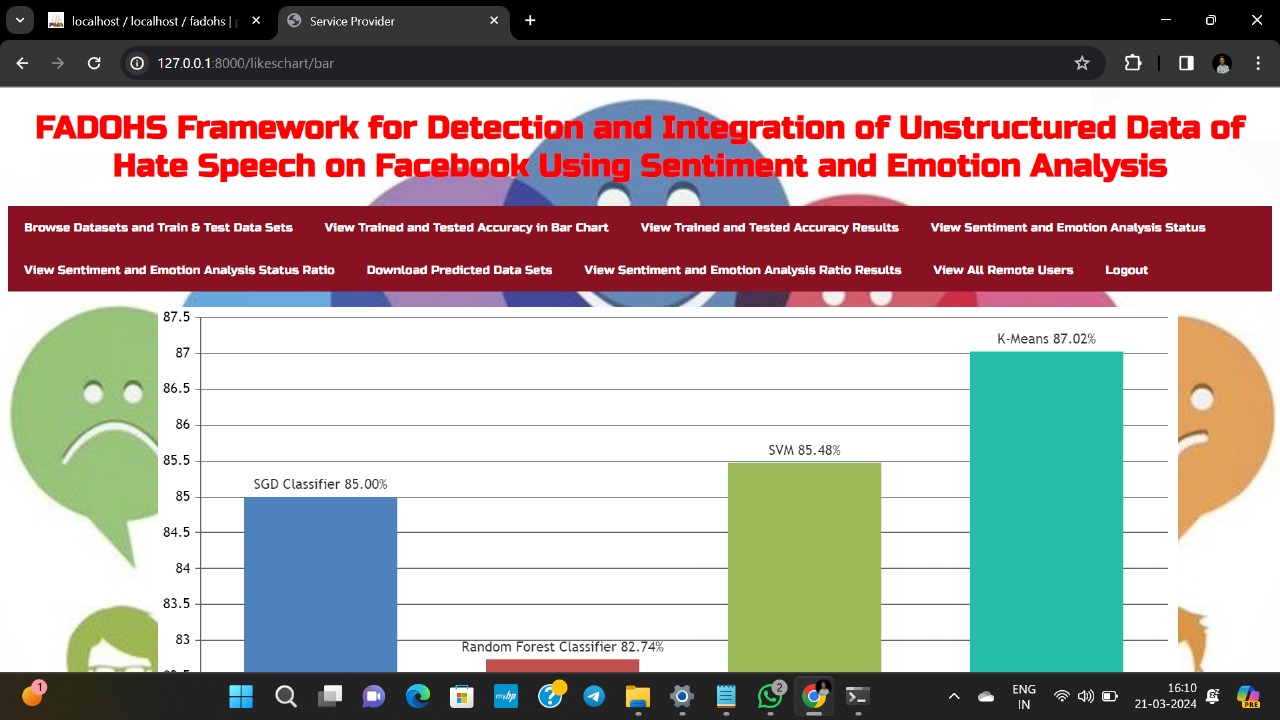
**Figure 10.7 Login**

**Figure 10.8 Register Your Details**

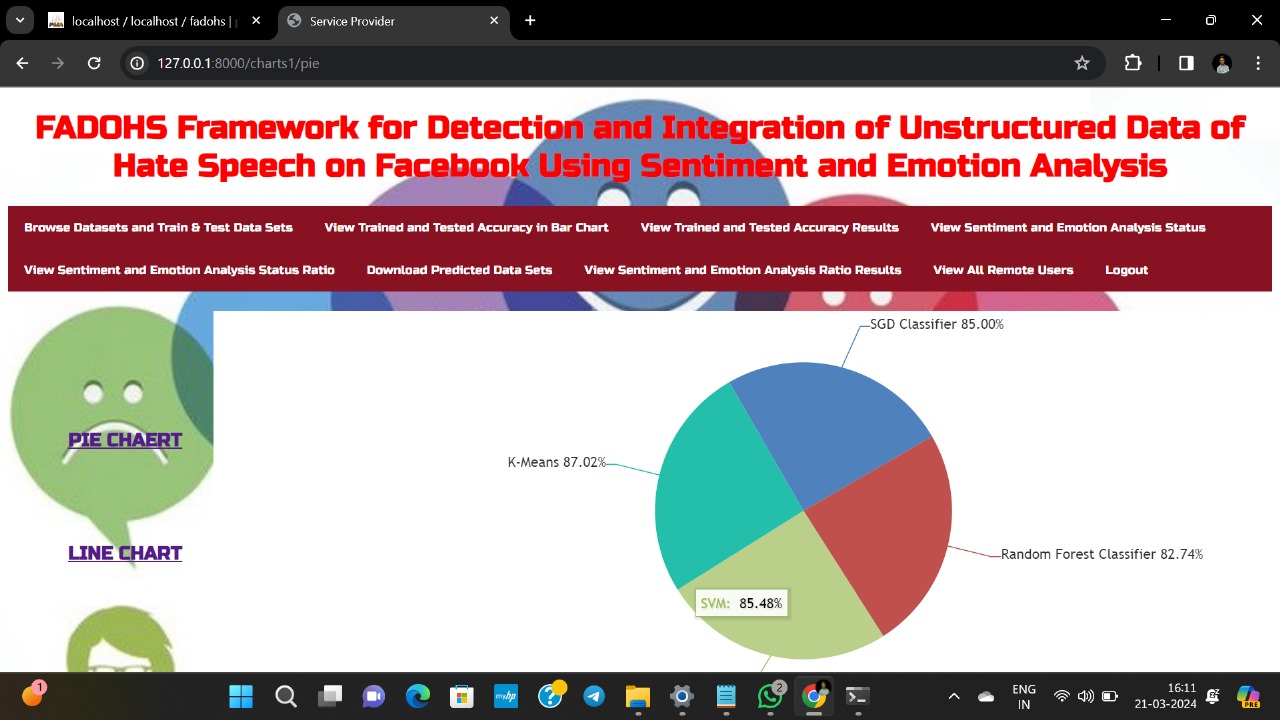
**Figure 10.9 Prediction**

**Figure 10.10 Outcome of Prediction**

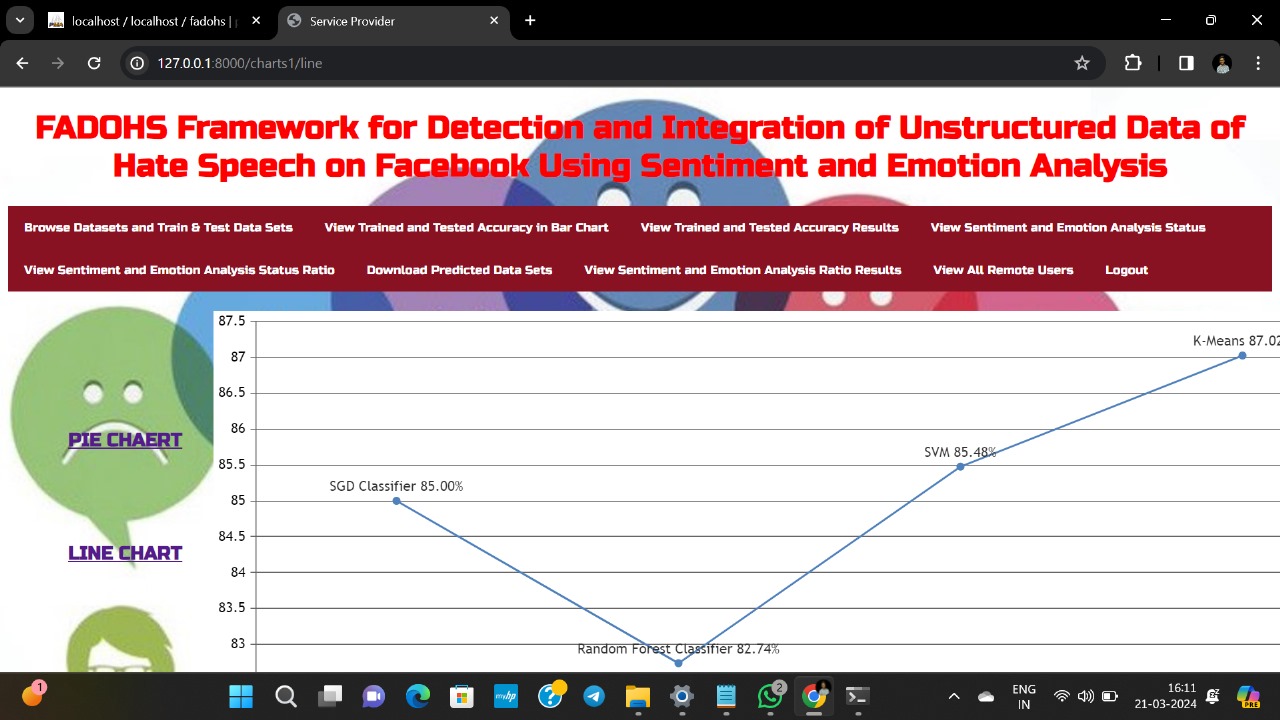
**Figure 10.11 Admin Login**



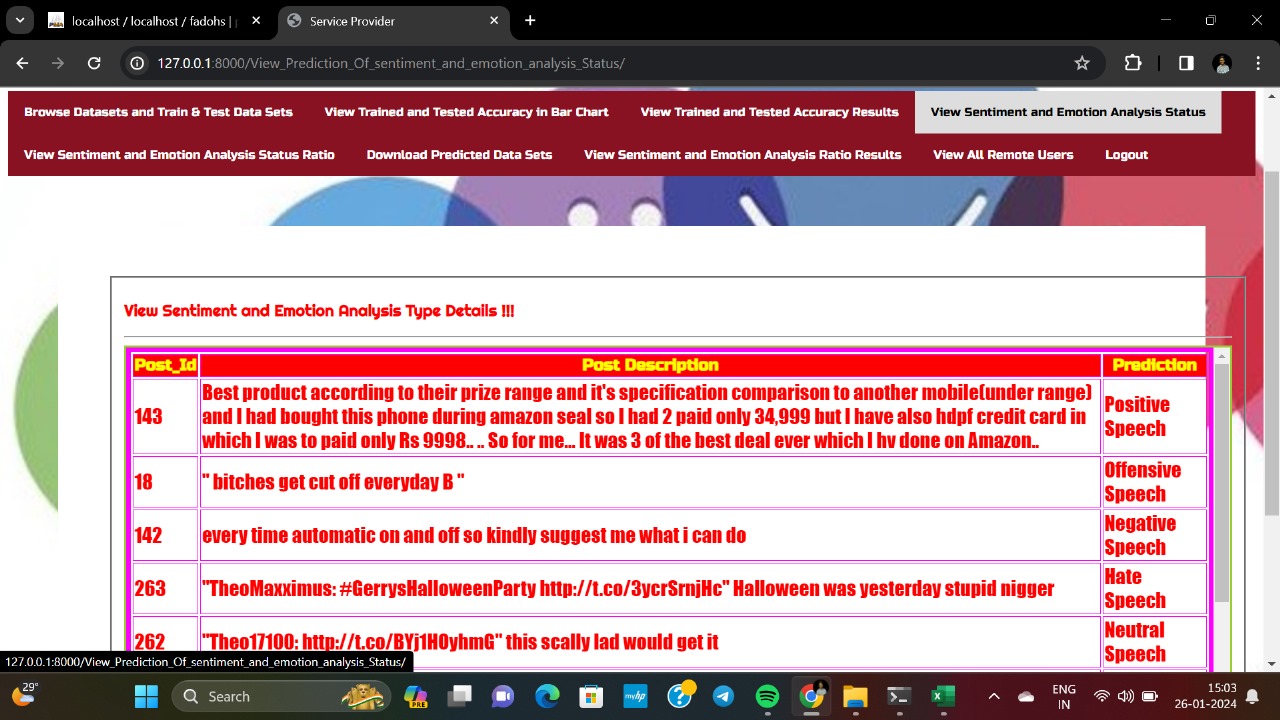
**Figure 10.12 Bar chart Accuracy**



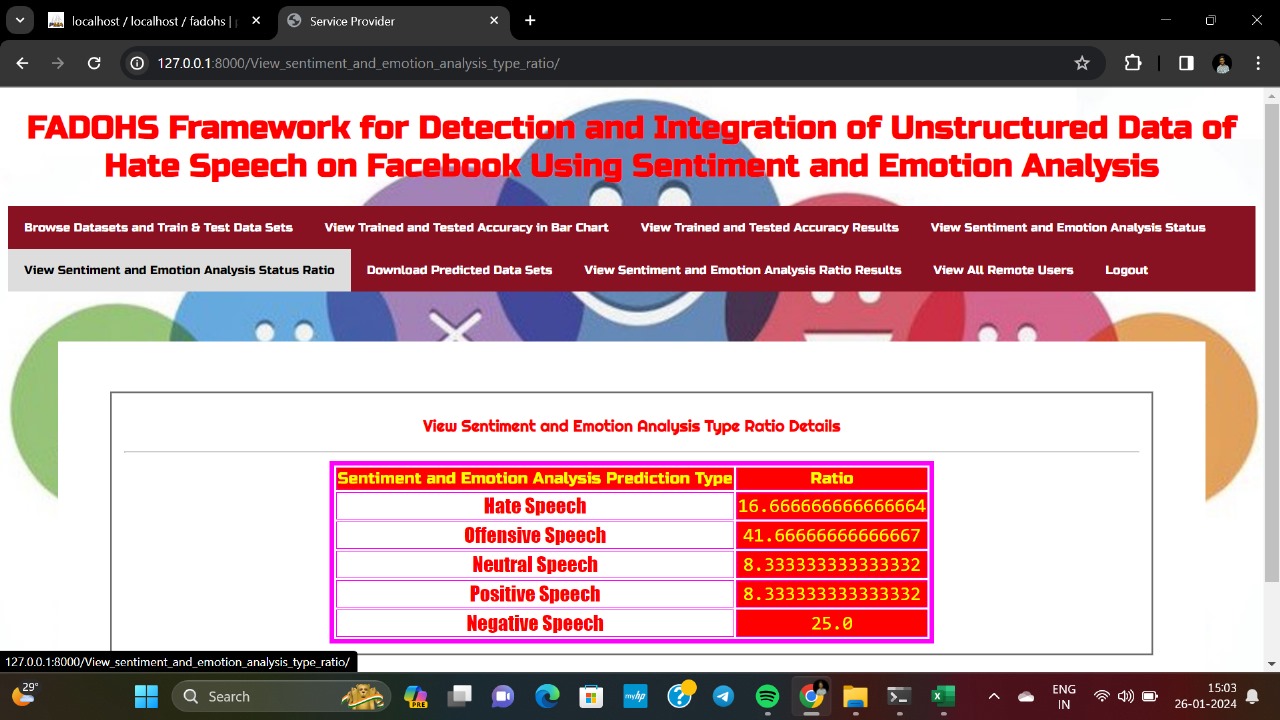
**Figure 10.13 pie chart representation**



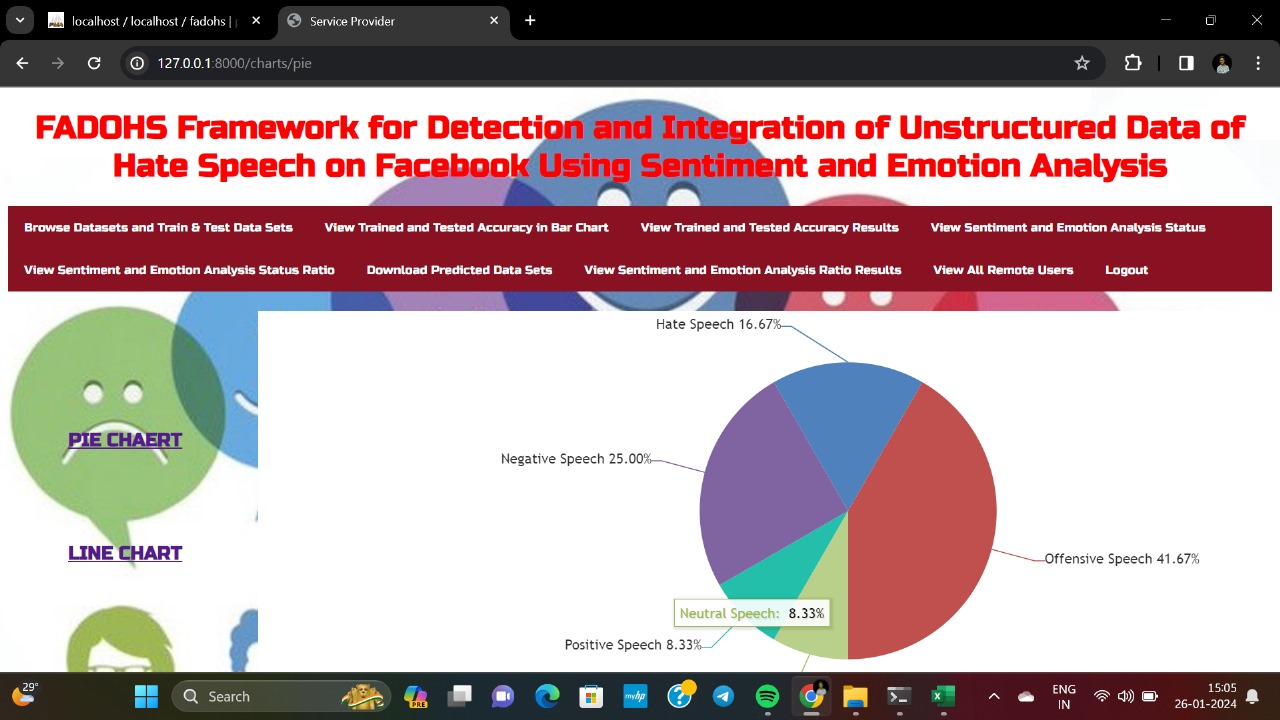
**Figure 10.14 Line Chart Representation**



**Figure 10.15 Status of sentiment and emotional analysis**



**Figure 10.16 Ratio of sentiment and emotional analysis**



**Figure 10.17 Pie chart representation of prediction type**

# 11.TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

## TYPES OF TESTS

### Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach: Field testing will be performed manually and functional tests will be written in detail.

Test objectives:

* + - 1. All field entries must work properly.
      2. Pages must be activated from the identified link.
      3. The entry screen, messages and responses must not be delayed.

features to be tested:

1. Verify that the entries are of the correct format.
2. No duplicate entries should be allowed.
3. All links should take the user to the correct page.

### Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

### Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

* + - 1. Valid Input : identified classes of valid input must be accepted. Invalid Input : identified classes of invalid input must be rejected. Functions : identified functions must be exercised.
      2. Output : identified classes of application outputs must be exercised.
      3. Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identifying Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

### System testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation

### USER TRAINING

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

### MAINTENANCE

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user’s requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible

extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

### TESTING STRATEGY

A strategy for system testing integrates system test cases and design techniques into a well-planned series of steps that results in the successful construction of software. The testing strategy must cooperate with test planning, test case design, test execution, and the resultant data collection and evaluation.

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code .

Segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

# 12. CONCLUSION

In this Project, we propose FADOHS, which identifies and integrates unstructured data from Facebook pages to promote hate speech, allowing us to identify the most commonly discussed topics.. The proposed framework offers a novel approach for clustering posts and comments, detecting highly discussed topics that generate hate speech, and identifying hate speech.

FADOHS uses dictionaries, sentiment/emotion analysis, and clustering methods to group and analyse posts that may contain hate speech. To properly address the hate speech issue, we begin our analysis with a small set of pages known to discuss sensitive topics that contains hateful comments. We use predetermined dictionaries, sentiment, and emotion analysis to identify posts with a high level of negativity in the comments. The results allow us to confidently conclude that unstructured data can be identified and integrated from hate speech-promoting pages.

The next critical phase is to categorize these data, which is accomplished by using the K-means clustering algorithm. We then manually analyse the posts in each group and label each cluster. By comparing the manual label to the cluster centroids, we can conclude that both variables are identical, confirming the efficiency of our method.

This work is a clear example of taking unstructured data, such as Facebook posts, and applying a framework to achieve meaningful analysis. According to the experimental results, the proposed FADOHS framework outperforms the state-of-the-art method in terms of precision, recall, and F1 scores by about 10%. Our framework had the maximum Precision, recall and F1 score as 54,61 and 49 respectively in recognizing the hate speech in social, according to the fact-finding investigation.

In future research, we intend to expand our framework to include not only comments but also replies in order to accurately identify individuals suspected of promoting hate speech. Long-term advantages include the ability to detect cyber bullies and cyber terrorists. We would also like to conduct a more in-depth analysis of the emotion filtering and clustering findings to determine the most reliable setup for optimizing results.

# 13. REFERENCES

[1] A. Matamoros-Fernández and J. Farkas, ``Racism, hate speech, and social media: A systematic review and critique,'' *Telev. New Media*, vol. 22, no. 2, pp. 205\_224, Feb. 2021.

[2] F. Del Vigna, A. Cimino, F. Dell-TOrletta, M. Petrocchi, and M. Tesconi,``Hate me, hate me not: Hate speech detection on Facebook,'' in *Proc. 1st* *Italian Conf. Cybersecur. (ITASEC)*, Venice, Italy, 2017, pp. 86\_95.

[3] D.Kucukusta, M. Perelygina, andW. S. Lam, ``CSR communication strategies and stakeholder engagement of upscale hotels in social media,'' *Int.* *J. Contemp. Hospitality Manage.*, vol. 31, no. 5, pp. 2129\_2148, May 2019.

[4] A. Rodriguez, C. Argueta, and Y.-L. Chen, ``Automatic detection of hate speech on Facebook using sentiment and emotion analysis,'' in *Proc. Int.* *Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Feb. 2019, pp. 169\_174.

[5] G. C. Santia and J. R. Williams, ``BuzzFace: A news veracity dataset with Facebook user commentary and egos,'' in *Proc. 12th Int. AAAI Conf. Web* *Social Media*. Palo Alto, CA, USA, 2018, pp. 531\_540.

[6] A. Chopra, A. Dimri, and S. Rawat, ``Comparative analysis of statistical classi\_ers for predicting news popularity on social web,'' in *Proc. Int. Conf.* *Comput. Commun. Informat. (ICCCI)*, Jan. 2019, pp. 1\_8.

[7] B. Lin, F. Zampetti, G. Bavota, M. Di Penta, M. Lanza, and R. Oliveto,``Sentiment analysis for software engineering: How far can we go?'' in *Proc. IEEE/ACM 40th Int. Conf. Softw. Eng. (ICSE)*. Gothenburg, Sweden: IEEE, 2018, pp. 94\_104.

[8] V. Franzoni, Y. Li, and P. Mengoni, ``A path-based model for emotion abstraction on

Facebook using sentiment analysis and taxonomy knowledge,'' in *Proc. Int. Conf. Web Intell.*, Aug. 2017, pp. 947\_952.

[9] S. Al Mansoori, A. Almansoori, M. Alshamsi, S. A. Salloum, and K. Shaalan, ``Suspicious activity detection of Twitter and Facebook using sentimental analysis,'' *TEM J.*, vol. 9, no. 4, p. 1313, 2020.

[10] S. Sadiq, M. Umer, S. Ullah, S. Mirjalili, V. Rupapara, and M. Nappi, ``Discrepancy detection between actual user reviews and numeric ratings of Google app store using deep learning,'' *Expert Syst. Appl.*, vol. 181, p. 115111, 2021.

[11] M. Shad Akhtar, D. Ghosal, A. Ekbal, P. Bhattacharyya, and S. Kurohashi, ``A multi-task ensemble framework for emotion, sentiment and intensity prediction,'' 2018, *arXiv:1808.01216*.

[12] A. Hussain, A. Tahir, Z. Hussain, Z. Sheikh, M. Gogate, K. Dashtipour, A. Ali, and A. Sheikh, ``Arti\_cial Intelligence\_Enabled analysis of public attitudes on Facebook and Twitter toward COVID-19 vaccines in the united kingdom and the united states: Observational study,'' *J. Med. Internet Res.*, vol. 23, no. 4, Apr. 2021, Art. no. e26627.

[13] U. Bhaumik and D. K. Yadav, ``Sentiment analysis using Twitter,'' in *Computational Intelligence and Machine Learning*. Singapore: Springer, 2021, pp. 59\_66.

[14] N. Dey, M. S. Rahman, M. S. Mredula, A. S. M. S. Hosen, and I.-H. Ra,

``Using machine learning to detect events on the basis of Bengali and banglish Facebook posts,'' *Electronics*, vol. 10, no. 19, p. 2367, Sep. 2021.

[15] P. Mehta and S. Pandya, ``A review on sentiment analysis methodologies, practices and applications,'' *Int. J. Sci. Technol. Res.*, vol. 4, pp. 601\_609, Dec. 2020.

[16] M. N. Moghadasi, Z. Safari, and Y. Zhuang, ``A sentimental and semantical analysis on Face book comments to detect latent patterns,'' in *Proc. IEEE* *Int. Conf. Big Data (Big Data)*, Dec. 2020, pp. 4665\_4671.

[17] R. G. Almonte, C. R. Malizon, M. R. F. Montano, and J. Olimpiada, ‘‘Sentiment analysis of local college in the Philippines using Facebook posts towards good governance: A framework proposal,’’ in Proc. 4th Int. Conf. Inf. Comput. Technol. (ICICT), 2021, pp. 47–51.

[18] T. Reddy, ‘‘An analysis of the popularity of Facebook news posts,’’ Projects Appl. Data Sci., vol. 4, p. 1, Oct. 2019.

[19] D. A. J. Daniel and M. J. Meena, ‘‘A novel sentiment analysis for Amazon data with TSA based feature selection,’’ Scalable Comput., Pract. Exp., vol. 22, no. 1, pp. 53–66, Feb. 2021.

[20] A. Oboler and K. Connelly, ‘‘Hate speech: A quality of service challenge,’’ in Proc. IEEE Conf. E-Learn., E-Manage. E-Services (IC3e), Dec. 2014, pp. 117–121.

[21] A. Matamoros-Fernández, ‘‘Platformed racism: The mediation and circulation of an Australian race-based controversy on Twitter, Facebook and Youtube,’’ Inf., Commun. Soc., vol. 20, no. 6, pp. 930–946, Jun. 2017.

[22] K. Müller and C. Schwarz, ‘‘Fanning the flames of hate: Social media and hate crime,’’ J. Eur. Econ. Assoc., vol. 19, no. 4, pp. 2131–2167, 2021.

[23] P. Fortuna and S. Nunes, ‘‘A survey on automatic detection of hate speech in text,’’ ACM Comput. Surv., vol. 51, no. 4, pp. 1–30, 2018.

[24] C. Baydogan and B. Alatas, ‘‘Metaheuristic ant lion and moth flame optimization-based novel approach for automatic detection of hate speech in online social networks,’’ IEEE Access, vol. 9, pp. 110047–110062, 2021.

[25] OpenAI. (2019). Better Language Models Their Implications. Accessed: Apr. 3, 2020. [Online]. Available: https://openai.com/blog/betterlanguage-models/.

[26] T. Klein and M. Nabi, ‘‘Learning to answer by learning to ask: Getting the best of GPT-2 and BERT worlds,’’ 2019, arXiv:1911.02365.

[27] K. Ethayarajh, ‘‘How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings,’’ 2019, arXiv:1909.00512.

[28] P. Budzianowski and I. Vulià, ‘‘Hello, It’s GPT-2 - how can i help you? Towards the use of pretrained language models for task-oriented dialogue systems,’’ 2019, arXiv:1907.05774.

[29] E. Karampotsis, K. Boulas, and Tzanetos, ‘‘Computational intelligence techniques for modelling the critical flashover voltage of insulators: From accuracy to comprehensibility,’’ in Proc. Int. Conf. Ind., Eng. Other Appl. Appl. Intell. Syst., 2017, pp. 295–301.

[30] M. Onat Topal, A. Bas, and I. van Heerden, ‘‘Exploring transformers in natural language generation: GPT, BERT, and XLNet,’’ 2021, arXiv:2102.08036.

[31] S. Levy, M. Saxon, and W. Y. Wang, ‘‘Investigating memorization of conspiracy theories in text generation,’’ in Proc. Findings Assoc. Comput. Linguistics. Ithaca, NY, USA, 2021, pp. 4718–4729.

[32] S. H. An and O. R. Jeong, ‘‘A study on the psychological counseling ai chatbot system

based on sentiment analysis,’’ J. Inf. Technol. Services, vol. 20, no. 3, pp. 75–86, 2021.

[33] S. S. Tu, ‘‘Limits of using artificial intelligence and GPT-3 in patent prosecution,’’ SSRN, 2021.

[34] K. Zhao, H. Ding, K. Ye, X. Cui, and Z. Fu, ‘‘A latent variable model with hierarchical structure and GPT-2 for long text generation,’’ in Proc. Int. Conf. Artif. Neural Netw. Cham, Switzerland: Springer, 2021, pp. 297–308.

[35] S. Ludwig, C. Mayer, C. Hansen, K. Eilers, and S. Brandt, ‘‘Automated essay scoring using transformer models,’’ Psych, vol. 3, no. 4, p. 915, Dec. 2021.

[36] R. Lawson, ‘‘Fine-tuning Daria: Exploring the implications of temperature, epochs, & corpus size on GPT-2 screenplay generation,’’ 2021.

[37] J.-S. Lee and J. Hsiang, ‘‘Patent claim generation by fine-tuning OpenAI GPT-2,’’ 2019, arXiv:1907.02052.