A Project Report

on

HATE SPEECH DETECTION

Submitted in partial fulfillment of the

requirement for the award of the degree of

Bachelor of Computer Applications

Submitted By

Kanchi Katiyar- 21SCSE1040161

Bhawna Rani- 21SCSE1040125

Divya Kumari- 21SCSE1040140

IN

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

BACHELOR OF COMPUTER APPLICATION

Under The Supervision of

Name of SUPERVISOR: Ms. Richa Kumari

Designation: Assistant Professor, SCSE

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

CERTIFICATE

This is to certify that Project Report entitled "**Hate Speech Detection**" which is submitted by Kanchi Katiyar, Bhawna Rani, Divya Kumari in partial fulfillment of the requirement for the award of degree BCA in Department of School of Computer Application and Technology Galgotias University, Greater Noida, India is a record of the candidate own work carried out them under my supervision. The matter embodied in this project is original and has not been submitted for the award of any other degree

Signature of Examiner

Signature of Supervisor

Ms. Richa Kumari

Acknowledgement

We would like to express our sincere gratitude to each fellow group member for their unwavering support, collaboration, and dedication throughout the course of this project. Working together has been an enriching experience, and we are thankful for the collective effort, creativity, and diligence that each member brought to the team with their exceptional contributions and the seamless coordination that made this project a success.

We would also like to extend our appreciation to our project advisor Ms. Richa Kumari ma'am, for her invaluable guidance and insights. This project would not have been possible without the combined efforts of all involved, and we are grateful for the opportunity to work alongside each other.

Abstract

With the pervasive growth of online communication platforms, the rise of hate speech has become a significant concern, necessitating effective tools for its detection and mitigation. This project presents a comprehensive approach to hate speech detection leveraging machine learning techniques. The proposed system integrates natural language processing (NLP) and deep learning algorithms to analyze textual data and identify instances of hate speech.

The project begins by curating a diverse dataset that encompasses various forms of hate speech across different contexts and demographics. This dataset is used to train and fine-tune a machine learning model capable of discerning between hate speech, offensive language, and benign content. The model is designed to recognize subtle nuances in language, including context and sarcasm, enhancing its accuracy and robustness. Additionally, the system incorporates continuous learning mechanisms to adapt to evolving language patterns and emerging forms of hate speech.

Key components of the project include preprocessing techniques for text data, feature engineering, and the implementation of state-of-the-art deep learning architectures such as recurrent neural networks (RNNs) or transformer models. Evaluation metrics such as precision, recall, and F1-score are employed to assess the model's performance, ensuring a balance between identifying hate speech and minimizing false positives

The outcomes of this project hold the potential to enhance online safety, promote healthier online discourse, and contribute to the creation of more inclusive digital spaces. Continuous learning mechanisms enabled the model to dynamically update its understanding of evolving linguistic expressions, ensuring resilience against novel forms of hate speech.

The proposed hate speech detection project is a robust and scalable solution for safer digital environments, that not only contributes to the ongoing research in the field but also addresses the pressing need for scalable and accurate solutions to combat online hate speech.

Table of Contents

Title		Page No.
Bonafide Certificate		**I**
Acknowledgement		**II**
Abstract		**III**
Contents		**IV**
List of Figures		**V**
Acronyms		**VI**
Chapter 1	**Introduction**	
	1.1 Introduction	**8**
	1.2 Formulation of Problem	**11**
	1.2.1 Tools and Technology Used	**12**
Chapter 2	**Literature survey/Project Design**	**16**
Chapter 3	**Functionality**	**18**
Chapter 4	**Results and Discussion**	**23**
Chapter 5	**Conclusion and Future Scope**	
	5.1 Conclusion	**26**
	5.2 Future Scope	**27**
	References	**29**

List of Figures

S.No.	Title	Page No.
1	Variations in texts of dataset	10
2	Methodology	17
3	Test data result	23
4	Test data grouping	24

Acronyms

NLP	Natural Language Processing
ML	Machine Learning
NLTK	Natural Language Toolkit
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network

CHAPTER-1

INTRODUCTION

Internet communication platforms are now an essential part of human contact in this age of fast digitization and global connectedness. These platforms provide a dark side in the shape of hate speech, a toxic incarnation that undermines the fundamental underpinnings of inclusive discourse, even as they provide never-before-seen chances for discourse and information exchange. Open discourse and societal harmony are seriously threatened by hate speech, which is defined as any communication that denigrates, discriminates against, or incites violence against individuals or groups based on characteristics like race, religion, ethnicity, gender, or sexual orientation.

In this era of rapid digitization and worldwide connectivity, internet communication platforms have become an indispensable component of human interaction. These platforms offer never-before-seen opportunities for discourse and information exchange, but they also have a dark side in the form of hate speech, a toxic embodiment that threatens the foundations of inclusive discourse. Hate speech, which is defined as any communication that disparages, discriminates against, or incites violence against individuals or groups based on traits like race, religion, ethnicity, gender, or sexual orientation, poses a major threat to free speech and social harmony.

Internet communication platforms are becoming an essential part of human contact in this age of increasing digitization and global connection. These forums present previously unheard-of chances for dialogue and information sharing, but they also have a negative aspect in the shape of hate speech, a poisonous manifestation that jeopardizes the principles of inclusive discourse. Free speech and social harmony are seriously threatened by hate speech, which is defined as any communication that denigrates, targets, or encourages violence against individuals or groups because of characteristics such as race, religion, ethnicity, gender, or sexual orientation.

In this era of growing global connectivity and digitization, internet communication platforms are becoming an indispensable component of human interaction. These forums offer unprecedented opportunities for discussion and knowledge exchange, but they also have a drawback in the form of hate speech, a toxic incarnation that puts the values of inclusive discourse in jeopardy. Hate speech, which is defined as any communication that disparages, targets, or supports violence against individuals or groups because of qualities like race, religion, ethnicity, gender, or sexual orientation, poses a major threat to both free speech and societal harmony.

However, identifying hate speech is a difficult task. First, there are differences in opinion over the definition of hate speech. This implies that depending on how each definition is used, some content may be regarded as hate speech by some people but not by others. We begin by discussing various definitions, concentrating on the various elements that go

into making hate speech. We cannot claim to be exhaustive, as new definitions are always being developed. Our goal is only to show variations that draw attention to the challenges that result from them.

Conflicting definitions make it difficult to evaluate hate speech detection systems because different datasets exist that define hate speech differently, resulting in datasets that not only come from different sources but also capture different information. This can make it challenging to determine which characteristics of hate speech to look for directly. In the following section, we go over the different datasets that can be used to test and evaluate hate speech detection systems. Depending on the definition, additional difficulties in automatically identifying hate speech arise from linguistic nuances and subtleties.

Despite variations, some recent methods for identifying hate speech in textual content have shown encouraging results. Text is classified as hate speech using machine learning techniques in the suggested solutions. These methods have the drawback of sometimes making decisions that are hard for people to understand and why they were made. This is a real problem because speech censoring systems that operate automatically probably require a manual appeals procedure. In order to tackle this issue, we suggest a novel approach to classifying hate speech that enhances comprehension of the choices made and demonstrates its superior performance over current methods on select datasets.

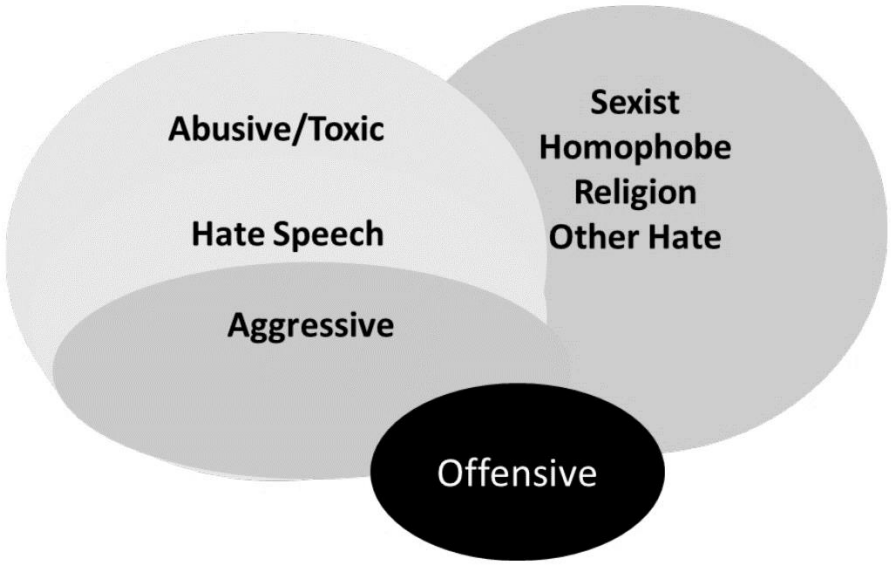


Fig 1

Defining the speech

Neither the definition of hate speech as a whole nor any one of its components is fully agreed upon. It was thought that a precise definition of hate speech would aid in the study of hate speech detection by simplifying the process of annotating hate speech and enhancing the accuracy of the annotations. Some are hesitant to precisely define hate speech because the boundary between it and appropriate free expression is hazy.

1.Encyclopedia of the American Constitution:

"Hate speech is speech that attacks a person or group on the basis of attributes such as race, religion, ethnic origin, national origin, sex, disability, sexual orientation, or gender identity," according to the Encyclopedia of the American Constitution.

2.Facebook:

"A direct attack on individuals based on what we refer to as protected characteristics—race, ethnicity, national origin, religion, sexual orientation, caste, sex, gender, gender identity, and a serious illness or disability—is what we consider hate speech. We also offer certain immigration status protections. Attacks are described as acts of violence or dehumanization, inferiority complex statements, or demands for isolation or segregation.

3.Twitter:

Hateful conduct is defined as follows on Twitter: "You may not incite violence against, directly attack, or threaten other people on the basis of your race, ethnicity, national origin, sexual orientation, gender, gender identity, religion, age, handicap, or serious illness."

4.Davidson et al.:

Language that is meant to be disparaging, demeaning, or insulting to members of a particular group, or that expresses hatred towards them, according to Davidson et al.

Datasets

It is difficult to gather and annotate data in order to train automatic classifiers to identify hate speech. In particular, it can be challenging to determine whether a given text qualifies as hate speech because, as was previously mentioned, there is no accepted definition of the term. According to research on the validity of hate speech annotations by Ross et al., annotators are not trustworthy.

In addition, hate speech thrives on social media platforms, despite the fact that many of them have very stringent guidelines for data distribution and usage. Because of this, there are only a few datasets that are available for public analysis, the majority of which are from Twitter (which has a more lax policy regarding data usage). Though the Twitter resources are useful, the character limit makes them less applicable to a wider audience; this leads to brief, to-the-point text. Posts from other platforms, on the other hand, are usually longer and can contribute to a broader conversation about a particular subject. This adds more context, which could change the text's meaning.

Twitter Hatebase :.

A collection of 24,802 tweets from Davidson et al. [9] is one Twitter dataset. They followed the following process to create the dataset. Initially, they obtained a list of terms associated with hate speech from Hatebase [16] and used it to search tweets from roughly 33,000 users. After that, they extracted a timeline from each of these users, yielding a collection of about 85 million Tweets. A randomized sample of 25,000 tweets containing terms from the lexicon was selected from the dataset comprising approximately 85 million tweets. They annotated each tweet using crowdsourcing, classifying it as either offensive (but not hate speech), hate speech, or neither. The tweet was disregarded if the annotators' agreement was too low.

Kaggle:

A shared task on identifying offensive comments was hosted by Kaggle.com [21]. 8,832 social media comments that have been classified as offensive or not make up the dataset. Insulting text may be a sign of hate speech, even though it's not always the case.

TRAC:

A shared task aimed at identifying aggressive text in Hindi and English was organized by the 2018 Workshop on Trolling, Aggression, and Cyberbullying (TRAC) [19]. Hate speech often includes aggressive text. The task's dataset, which comprises 15,869 Facebook comments classified as overtly aggressive, covertly aggressive, or non-aggressive, is open to the public. The same labels are also present in a tiny Twitter dataset that consists of 1,253 tweets.

—

Formulation of problem

The creation of a scalable and efficient hate speech detection system is the main issue this project attempts to solve. This system needs to be able to accurately distinguish hate speech from benign or offensive content while maintaining the right balance between protecting people's right to free speech and protecting communities from the negative effects of hate speech.

Key facets of this problem formulation include:

- Diverse Hate Speech Manifestations:
  Hate speech can take many different forms, differing in context, subtlety, and intensity. Finding such a variety of forms—from obvious to subtle—is extremely difficult.

- Context Sensitivity:
  Since hate speech frequently makes use of sarcasm, cultural allusions, and contextual cues, it is critical that the detection system comprehend and take into account the context in which the content is presented.

- Changing Linguistic Patterns:
  Language is always changing, and hate speech is no different. To continue to be effective over time, the hate speech detection system must constantly adjust to changing linguistic trends, slang, and new forms of hate speech.

- Mitigation Without Overreach:
  It's important to strike a balance when it comes to identifying hate speech and avoiding overzealous censorship. The system ought to uphold the principles of free speech and reduce the number of false positives.

- Generalizability:
  To ensure its application in a variety of online environments, the solution must be able to identify hate speech across linguistic, cultural, and demographic divides.

- Scalability:
  The system needs to be scalable in order to efficiently process and analyze massive volumes of text, as online platforms host vast amounts of content on a daily basis.

- This project aims to address these intricate challenges by harnessing cutting-edge natural language processing (NLP) and machine learning techniques, ultimately contributing to the creation of safer, more inclusive digital spaces.

Tools and technology used

Machine learning (ML) and natural language processing (NLP) methods are combined in hate speech detection projects. An efficient hate speech detection system's development and implementation depend greatly on the technologies and tools selected. The following technologies and tools are essential to this project:

- **Programming Language: Python**

Python is a well-liked language for natural language processing and machine learning. Essential functionalities for tasks related to text processing and machine learning are provided by libraries like scikit-learn, spaCy, and NLTK (Natural Language Toolkit).

The most popular high-level language is Python, which Guido van Rossum created and initially made available on February 20, 1991. It is an interpreted programming language with a syntax that is easy to read. It offers a number of frameworks and libraries that make machine learning development easier. Python is a great language for machine learning projects because of its adaptability and vibrant community. It also supports object-oriented programming, which is the most popular kind of general-purpose programming. Python finds application in a multitude of fields, including Artificial Intelligence, Data Science, Machine Learning, Deep Learning, Networking, Game Development, Web Development, Web Scraping, and more.

Indeed, a subset of artificial intelligence is machine learning. Because of its design, computers are able to recognize patterns, anticipate outcomes, and make decisions based on information, all without the need for explicit programming from programmers. This allows computers to learn from their experiences and become more intelligent. Here, Python, a flexible programming language, has emerged as a solid option for beginners and benefits a lot of machine learning enthusiasts because of its ease of use, extensive library, and multitude of applications.

The Use of Python in Machine Learning:

Python's contribution to machine learning is significant because of its libraries, which include NumPy, Pandas, Scikit-learn, TensorFlow, and Keras. These libraries provide the tools and features needed for analyzing, manipulating, and creating machine learning models. It has a strong reputation for being readable and providing platform independence. It is the ideal language for machine learning because of all of these factors.

- **Libraries: NLTK, Pandas, NumPy, sklearn**

Natural Language Toolkit, or NLTK, is an effective library for handling data related to human language. Tokenization, stemming, tagging, parsing, and semantic reasoning are among the functions it offers.

Pandas:

For working with data sets, use the Pandas Python library. Its features include data manipulation, cleaning, analysis, and exploration.
Pandas is a powerful and versatile library that simplifies tasks of data manipulation in Python . Pandas is built on top of the NumPy library and is particularly well-suited for working with tabular data, such as spreadsheets or SQL tables. Its versatility and ease of use make it an essential tool for data analysts, scientists, and engineers working with structured data in Python.

Have you ever wondered why pandas are commonly used in data science applications? This is due to the fact that pandas is utilized in conjunction with additional data science-related libraries. Because Pandas is built on top of the NumPy library, many of NumPy's structures are either replicated or used. Pandas generates data that is frequently used as input for SciPy's statistical analysis, Matplotlib's plotting functions, and Scikit-learn's machine learning algorithms. A list of things we can accomplish with Pandas is provided here.

NumPy:

Numpy is a library that extends the capabilities of the Python programming language by enabling the manipulation of sizable multi-dimensional arrays and matrices. It also provides a vast array of sophisticated mathematical functions that can be applied to these arrays.

NumPy, an acronym for numeric python, is a Python package designed for processing and calculating both single- and multidimensional array elements.
In order to create the NumPy package, Travis Oliphant injected the features of the ancestor module Numeric into the Numarray module in 2005.
It is a Python extension module that was primarily written in C. It offers a number of functions that can quickly complete numerical computations.

Numerous strong data structures, including multi-dimensional arrays and matrices, are implemented by NumPy. These data structures are employed in the best possible calculations involving matrices and arrays.
We will go through the NumPy numerical Python library in this tutorial.

Sklearn:

Scikit-Learn, a Python library for statistical modeling and machine learning, is also referred to as sklearn. We can implement different machine learning models for regression, classification, and clustering using scikit-learn, and we can use statistical tools to analyze these models.
The most reliable and practical Python machine learning library is called Scikit-learn, or Sklearn. Through a Python consistency interface, it offers a range of effective tools for statistical modeling and machine learning, such as regression, clustering, classification, and dimensionality reduction. This library is based on NumPy, SciPy, and Matplotlib and is mostly written in Python.

The application of Sklearn:

The NumPy library is extensively used in Scikit-learn's mostly Python code to perform extremely effective array and linear algebra computations. To improve this library's efficiency, a few basic algorithms are also implemented in Cython. Wrappers written in Python for LIBSVM and LIBLINEAR are used to execute support vector machines, logistic regression, and linear SVMs, respectively. In such cases, expanding these

routines using Python might not be practical.

Numerous other Python packages, such as SciPy, Pandas data frames, NumPy for array vectorization, Matplotlib, seaborn and plotly for graph plotting, and many more, integrate well with scikit-learn.

Need for hate speech detection:

Over the past ten years, social media has grown astronomically in terms of both its scope and significance as a communication tool. Due to the nature of social media, anyone can post anything they want and express any viewpoint, regardless of how offensive, enlightening, or in between. Such posts may be visible to millions of people, depending on the forum. Although different forums define inappropriate content differently and have different procedures for doing so, automated methods play a significant role in this task due to the size of the medium. One significant component of this unsuitable content is hate speech.
However, there is no one definition for hate speech because it is a subjective and complex term. Regardless of the definition of the term or the nature of the issue, it is evident that in certain situations automated techniques for identifying hate speech are required. It is crucial in these situations that the techniques used are precise, successful, and efficient.

For the purpose of detecting hate speech, a range of techniques have been investigated: conventional classifiers, deep learning-based classifiers, or a mix of the two. However, a number of dataset benchmarks, including Davidson, Founta, and Twitter Sentiment Analysis (TSA), have been developed and made available for the assessment of the effectiveness of these techniques.

CHAPTER-2

LITERATURE REVIEW

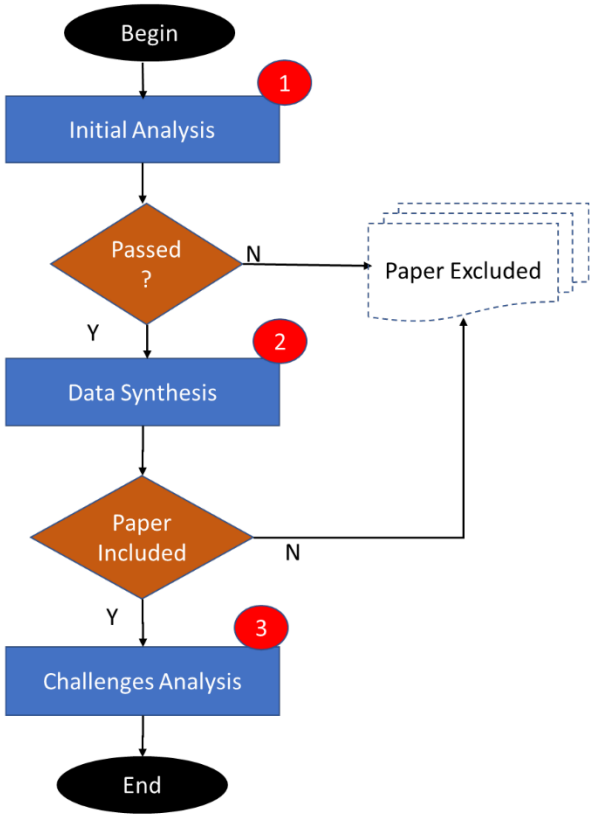
The goal of this project was to use natural language processing to identify hate speech. Prior to applying different techniques and understanding the concept behind Natural Language Processing, it was necessary to define hate speech and review relevant literature. Three labels—hate, offensive language, and neither—were applied to tweets in order to analyze them using a deep learning technique called a Convolutional Neural Network (CNN). The findings demonstrated that while the CNN architecture produced good results, some non-hate speech was mistakenly classified as hate speech. Nonetheless, CNNs have a great deal of potential to perform well if datasets are larger in terms of both size and quality.

The use of convolutional neural networks (CNNs) in hate speech detection is examined in this review of the literature. The review examined a number of studies that were released between 2016 and 2021 and emphasized how well various CNN-based models performed in identifying hate speech across a range of languages. According to the review, CNNs have outperformed more conventional machine learning methods like Naive Bayes and Support Vector Machines and have demonstrated significant promise in the identification of hate speech. Furthermore, in order to enhance the effectiveness of hate speech detection models, researchers have suggested a number of changes to CNN architectures, including the use of attention mechanisms and pre-trained word embeddings.

Character-level Convolutional Neural Network (Char CNN), Hierarchical Attention Network (HAN), and Convolutional Neural Network with Attention (CNN-Attention) are a few of the noteworthy CNN-based models covered in this literature review. These models have demonstrated a high degree of accuracy in recognizing hate speech in Arabic, Spanish, and English, among other languages. All things considered, this literature review offers a thorough summary of the most recent developments in CNN-based hate speech detection and emphasizes the room for more advancements in this field.

Methodology

Given how common this phenomenon is, this study focuses on machine learning classification models, hate speech datasets, and the key components of hate speech models. Using a survey and content analysis approach, we were able to locate and examine pertinent research using the Tranfield et al. methodology. Tranfield et al.'s methodology is comprised of four stages: the formulation of the research question, the definition of the review criteria, the final literature review criteria, and the extraction and combination of data. Finding the prevalent methods for addressing hate speech and the important datasets associated with it that are frequently utilized by the academic community is the main research question in this work. The review criteria are as follows: choosing a list of terms associated with hate speech and combining them with machine



learning techniques. Terms like "hate," "cyberhate," "racist," "offensive language," "online harm," "poisonous speech," "religion hate," "ethnicity, nationality, race, gender hate," "aggressive, abusive, misogynistic, sexist speech," and "homophones" are among those included in the selection;

fig 2

CHAPTER-3

FUNCTONALITY

This section explains the proposed system which we have employed to classify the text into three different classes namely, “hate speech, offensive speech, and no hate speech and offensive speech”. The methodology used in this project has 5 major steps which have been discussed below.

1. **Data Collection**
   You can find the dataset we used on www.kaggle.com to build our hate speech detection model. The purpose of this dataset is to conduct research on hate speech detection using data from Twitter. Hate speech, offensive language, and no hate speech or offensive speech have been assigned to the text or tweets in the dataset. Please be aware that this dataset contains text that may be offensive in general and racist, sexist, or homophobic due to the nature of the study. Seven columns make up the hate speech detection dataset: tweet, class, hate speech, offensive language, hate speech, count, and index.

2. **Data Preprocessing**
   While building a machine learning model, data preprocessing is the first and most important step because the data isn't always clean and well-formatted. Additionally, cleaning and formatting data is a must for any operation involving it. Thus, we employ the data preprocessing task for this. This step involves preparing the raw data so that a machine learning model can use it.

   Thus, stopword and stemmer—two crucial terms in natural language processing—are present in our dataset. In natural language processing, stopwords are terms that are useless. Those words can be removed from the input. The process of creating morphological alterations from a root word is known as stemming. To make better and simpler predictions, we must identify the stem word for every text.

3. **Splitting Data**
   Our dataset needs to be split into training and testing datasets. Three-quarters of the dataset will be used for testing, and the remaining portion will be used for training. The classification model is trained with the train data in order to learn how to classify the dataset. Test data is used in the interim to determine the accuracy of our classification model.

4. **Building the ML Model**
   Once we have separated our data, we need to find an algorithm that works well with our model so that we can create our machine learning model. We can construct our hate speech detector using a decision tree classifier. They belong to the category of supervised machine learning that is frequently applied to classification issues.

5. **Result Evaluation**

Using the testing set, our ML model will predict in this step whether the unlabeled text or tweet from our dataset belongs to the class of "hate speech, offensive speech, or no hate speech and offensive speech." This allows us to gauge the accuracy of our model.

Decision Tree

For classification problems, decision trees are a well-liked and frequently applied machine learning algorithm. They are an effective tool for identifying hate speech because they are easy to comprehend and analyze. Recursively dividing the data into smaller subsets according to the most informative features results in the construction of decision trees, continuing until a stopping criterion is satisfied. The outcome is a structure that resembles a tree and can be used to categorize new occurrences.

Several algorithms for the detection of hate speech were assessed during the experimentation phase. Potential candidates included SVM, KNN, random forest, naïve Bayes, and decision trees. Although some algorithms outperformed naïve Bayes, their computational demands presented difficulties for large-scale datasets, which limited their usefulness for this study.

Some common terminologies in decision tree:

Root Node: The decision tree originates at the root node. It depicts the complete dataset, which is then split up into two or more sets of similar data.

Leaf Node: After obtaining a leaf node, the tree cannot be further divided; leaf nodes are the ultimate output nodes.

Splitting: The process of splitting the decision node/root node into sub-nodes in accordance with the specified parameters is known as splitting.

Sub-tree: A tree created by slicing another tree into a branch or subtree.

Pruning: Removing undesirable branches from a tree is the process of pruning.

Parent/Child node: Nodes in a tree are referred to as parent and child nodes, respectively. The parent node is the root node.

Python implementation of decision trees:

We will now use Python to implement the decision tree. The dataset "user_data.csv," which we have utilized in earlier classification models, will be used for this. We can compare the decision tree classifier with other classification models, like KNN SVM, Logistic Regression, etc., by using the same dataset.

The following steps, which are listed below, will also not change:

- Step of pre-processing data
- Algorithm Fitting for Decision Trees to the Training Set
- Estimating the outcome of the test
- Test the result's accuracy (Creation of Confusion matrix).
- Displaying the test set's outcome.

Machine learning for hate speech detection

Following the preparation of the text for machine reading, the detection task can be carried out by classification algorithms. There are three types of machine learning approaches for classifiers: supervised, semi-supervised, and unsupervised.

Guided education:

Because this method depends on manually labeling a significant amount of text, it is domain dependent. Although labeling tasks take a lot of time and effort, they are more effective for events that are domain-dependent. The majority of techniques used for tasks involving the detection of hate speech are supervised techniques. For example, Burnap and Williams employed multiple supervised classifiers to identify hate speech on Twitter; their findings demonstrated that while all classifiers performed similarly, the model's accuracy varied depending on the feature settings. As a result, the features that can be taken out of the corpus determine which classifier should be used.

Semi-supervised:

This paradigm involves using both labeled and unlabeled data to train algorithms. The Hua et al. model demonstrates how labeled data combined with unlabeled data can effectively improve performance. They maintained that the capacity of unsupervised learning to deal with small-scale occurrences is constrained.

Conversely, supervised learning can capture small-scale events efficiently, but the model's scalability is reduced by the requirement to manually label the data set. A semi-supervised approach was recommended by the authors as a means of striking the correct balance between these two scenarios. Furthermore, Xiang et al. claimed that their method could be a good substitute for the expensive supervised approaches to detect hate speech

because it substituted an automatically generated feature for the costly manual annotation.

Unsupervised education:

It is a scalable method that is domain-independent and able to manage a wide range of content. Instead of using human labor to label a sizable training set, it dynamically extracts key terms associated with the domain. Gitari et al., used a bootstrapping strategy to develop their lexicon, beginning with a tiny hate verb seed and gradually adding to it. They found that adding them-based features and semantic hate to their model produced the best results.

Machine Learning Model

This task can be classified as domain-dependent when it involves working with a particular language (like Arabic) and geographic area. Therefore, the best options for this task are supervised approaches. But now that deep learning and deep neural networks have revolutionized NLP tasks, we'll think about limiting our options to these reliable models. Since RNN and CNN are the two most widely used deep learning models for NLP tasks, Yin et al. [72] conducted a comparative study between them. RNN essentially comes in two varieties: GRU and LSTM, and it can support sequential architectures.

CNN, on the other hand, is structured hierarchically. RNN are well suited for the long-range context dependencies, as demonstrated by Yin's experiments. CNN, however, does a better job of capturing regional features. When it comes to text size, the results of GRU and CNN can be compared; GRU performs better with slightly longer sentences. Ultimately, they came to the conclusion that hyperparameter tuning greatly influences the performance of deep neural networks.

Deep Learning

Deep learning models appear to have a bright future in text mining applications. It is totally dependent on artificial neural networks, albeit with more nuance. In an attempt to learn how to recognize patterns in the given text, it mimics the event in layers of neurons. Deep learning techniques are not always superior to conventional supervised techniques, though. Deep learning performance is dependent on selecting the appropriate algorithm, number of hidden layers, and feature representation method.

By contrasting the performance of Recurrent Neural Network (RNN) and Support Vector Machine (SVM), Al-Smadi et al., demonstrated the validity of the preceding assumption. They found that, for a given set of features, SVM performed better than RNN. Thus, they recommended that for their next work, they employ (LSTM) and an alternative embedding algorithm. Pitsilis et al. [34] broke through the barrier of language dependency in word embedding approach by using an RNN model with word frequency

vectorization to implement the features instead of word embedding for hate speech detection. Their findings outperformed the most advanced deep learning techniques available today for the detection of hate speech.

Neural networks, which are nothing more than a collection of decision-making networks that have been trained to perform a specific function, are the foundation upon which deep learning algorithms are dynamically built. Subsequently, every one of them goes through basic layered representations before proceeding to the subsequent layer. Nonetheless, the majority of machine learning is designed to function fairly well with datasets containing hundreds of features or columns. The main reason machine learning fails to classify a data set as structured or unstructured is that it cannot identify a basic 800x1000 RGB image. Such depths become very impractical for a traditional machine learning algorithm to handle. Here's where deep learning comes in.

Current Status of Related Concepts and Hate Speech Detection:

Some studies (e.g., cyberbullying, abusive language, radicalization detection) have examined various related terminologies that serve similar related concepts to the phenomenon of hate speech. The examination of these disparate terminologies will undoubtedly aid in gaining understanding of the current situation from various angles and will also help in identifying and recognizing the connections between these terminologies.

CHAPTER-4

RESULTS AND DISCUSSION

In this section, you can see the results that were obtained through out Hate Speech Detection model on being tested, so that we can know whether it can be trusted or not.

Accuracy of the model- this represents the accuracy of the analysis done by the ML model on the dataset. 89% of the test results came out to be correct while the rest were incorrect. Through this we can conclude that our model is pretty accurate in predicting the nature of the tweets or texts.

```
In [21]: from sklearn. metrics import accuracy_score
         print (accuracy_score (y_test,y_pred))

         0.890695684068957
```

fig 1

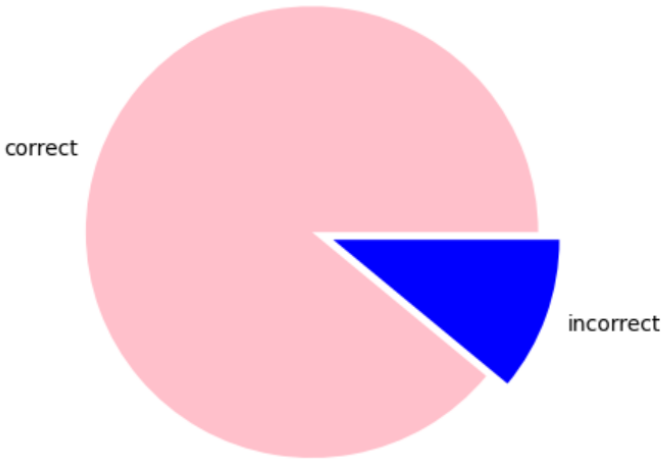


fig 3

Upon being tested on the test dataset, our ML model gave the following results:

```
In [20]: model. fit(X_train,y_train)
         y_pred = model. predict (X_test)
         y_pred

Out[20]: array(['Offensive Speech', 'Offensive Speech', 'Offensive Speech', ...,
                'No Hate and Offensive Speech', 'No Hate and Offensive Speech',
                'Offensive Speech'], dtype=object)
```

To represent the dataset into the three categories of speeches we first need to organize all the tweets/text into their respective categories to know how many texts belong to each one. Below is the code used to do the same:

```
df1= df.groupby('result').count()
df1
```

result	code
Hate Speech	346
No Hate and Offensive Speech	1417
Offensive Speech	6416

Finally we have created a bar plot to represent the dataset based on their categories that we grouped them as earlier. Below is the code and the bar plot for the same:

```
df1.unstack().plot(kind='bar')
plt.title("Text Analysis")
plt.xlabel("Type of speech")
plt.ylabel("Frequency of occurence")
plt.show()
```

The bar plot visualizes the distribution of our dataset. From this we can see that most of the tweets contain offensive speech while the hate speech is comparatively low in the dataset we have used. We can also see that the amount of speech with no hate or offensive language is relatively moderate.

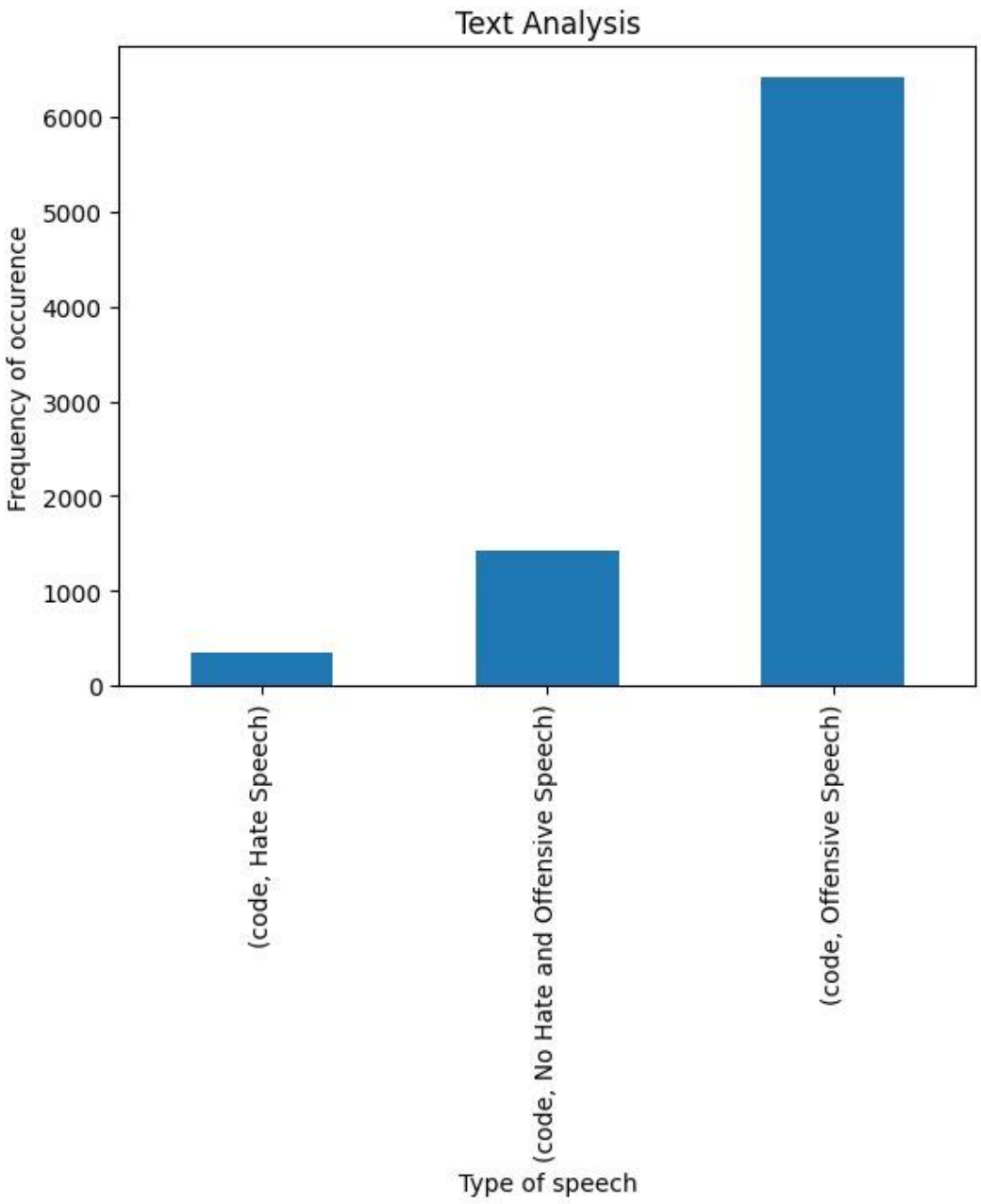

Text Analysis
Frequency of occurence
6000
5000
4000
3000
2000
1000
0
(code, Hate Speech)
(code, No Hate and Offensive Speech)
(code, Offensive Speech)
Type of speech

Fig 4

CHAPTER-5

CONCLUSION

In conclusion, this project has not only equipped us with a deeper understanding of the harmful impact of hate speech in our digital age but has also provided us with valuable insights into the complexities of developing effective automated detection systems. Through collaborative effort and the utilization of cutting-edge technology, we have taken significant steps toward countering hate speech and fostering more inclusive online spaces. While our project represents a vital contribution to this ongoing societal challenge, it is important to acknowledge that hate speech remains a dynamic and ever-evolving issue. As such, our work is far from finished, and we must continue to refine and expand our detection methods to ensure a safer and more tolerant online environment for all. This project has underscored the importance of vigilance and unity in the fight against hate speech, and we are determined to persist in our efforts to make a positive impact in the digital world.

FUTURE SCOPE

The future scope of hate speech detection holds significant promise and challenges. As technology continues to advance, we can expect the development of more sophisticated machine learning and natural language processing models capable of detecting hate speech in various languages and across diverse cultural contexts. Multimodal approaches, incorporating text, images, and audio data, will likely become more prevalent, enabling a comprehensive understanding of online content. Additionally, there is a growing emphasis on addressing bias and fairness in hate speech detection algorithms to minimize false positives and ensure ethical and equitable outcomes. The integration of explainable AI techniques will also be crucial in providing transparency and understanding of model decisions. Furthermore, interdisciplinary collaboration between researchers, social scientists, and policymakers is likely to become more essential in addressing the broader societal impact of hate speech and its detection. The future of hate speech detection will continue to evolve to meet the challenges posed by the dynamic and complex online communication landscape while striving to create safer and more inclusive digital spaces.

APPENDIX

List of reviewed datasets

No.	Dataset Name	Size (# of Tweets)	Categories of the Dataset
1	Waseem and Hovy	16,000	Racism, sexism, neither
2	Davidson et al.	24,783	Hate, offensive, neither
3	Waseem	6909	Racism, sexism, neither, both
4	SemEval Task 6 (OLID)	14,000 tweets	Level A: offensive, not offensive Level B: targeted insult, untargeted Level C: individual, group, other
5	SemEval Task 5 (HatEval)	19,600, 13,000 in English, 6600 in Spanish	Subtask A: hate, non-hate Subtask B: individual target, group target Subtask C: aggressive, non-aggressive
6	Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC)	5335 for the English dataset of HASOC 20207005 for HASOC 2019	Subtask A: hate and not offensive Subtask B: hate speech, offensive, and profanity
7	ElSherief et al.	25,278 hate instigators 22,857 targets 27,330 tweets	Archaic, class, disability, ethnicity, gender, nationality, religion, sexual orientation
8	Founta et al.	80,000 (Size Doesn't Guarantee Diversity [137])	Offensive, abusive, hateful speech, aggressive, cyberbullying, spam, normal

REFERENCES

Facebook, YouTube, Twitter, Microsoft, and Hern, A. sign the EU hate speech code. 31. The Guardian, 2016.

J. Rosa, Y. Bonilla, and Decolonizing diversity, deprovincializing Trump, and upending anthropology. 2017; American Ethnologist 44(2): 201-208.

Travis, A., Hate crimes against Muslims rise in the wake of the Manchester and London Bridge attacks. The Guardian, September 14, 2017.

Hate speech detection: Problems and solutions, S. MacAvaney et al.

S. Nunes and P. Fortuna, A survey on textual hate speech detection done automatically. 51(4), p. 85, ACM Computing Surveys (CSUR), 2018.

Tranfield, D.; Denyer, D.; Smart, P. Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review. Br. J. Manag. 2003, 14, 207–222. [Google Scholar] [CrossRef]