Enhanced Hate Speech Detection in Social Media using Transformer-based Models.

by

Anika Tabasshum
19201106
Fairuz Tasnim Ashrafi
19201035
Sadia Afreen
19201105

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
January 2024

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Anika

Anika Tabasshum
19201106

Fairuz

Fairuz Tasnim Ashrafi
19201035

Sadia

Sadia Afreen
19201105

Approval

The thesis titled "Enhanced Hate Speech Detection in Social Media using Transformer-based Models." submitted by

1. Anika Tabasshum (19201106)
2. Fairuz Tasnim Ashrafi (19201035)
3. Sadia Afreen (19201105)

Of Fall, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 9, 2024.

Examining Committee:

Supervisor:
(Member)

Md Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
School of Data and Sciences
Brac University

Abstract

Hate speech on social media can escalate into ”cyber conflict,” detrimentally impacting social life. With the exponential growth of Internet users and media content, identifying abusive language in audio and video content has become increasingly challenging. The nuances of human communication mean that individuals might employ seemingly non-hateful language in derogatory ways, often accompanied by specific voice tones and gestures that aren’t captured when converting multimedia into text. This research delves deep into the realm of hate speech detection, aiming to automatically identify harmful content across various social media platforms. Initially focused on text, our study utilized remote supervision for automatically labeled dataset creation and employed word embeddings with a bias toward hate. We analyzed datasets from Twitter, testing various machine-learning models to gauge the representation of hate speech and abusive language. Any tweet or online post exhibiting racist or sexist sentiments was categorized as ”hate speech.” Our objective was to classify such messages for better content moderation systematically. With advancements in our research, we have extended our detection capabilities to audio content. By leveraging Simple Feed-forward Neural Networks, RNNs, and CNNs, we can now discern hate speech patterns in audio with enhanced accuracy. However, the vastness of content on social media platforms means not every piece can be manually moderated. This underscores the importance of our automated hate speech detection, especially when dealing with content in linguistically challenging languages. However, social media networks cannot control every piece of user content. Because of this, it is necessary to identify hate speech automatically. This desire is heightened when the content is written in challenging languages. Our study provides a unique transformer-based methodology for detecting hate speech in social media. The proposed model uses Natural Language Processing (NLP) approaches to assess text and audio input. To increase the accuracy of hate speech identification, we use sophisticated deep learning architectures such as attention methods and transformers. Our model is trained on a huge dataset of tweets and audio recordings, and its performance is measured using a variety of criteria. Our transformer-based approach beats existing state-of-the-art hate speech identification methods, according to the results. Our study makes an essential addition to the field of computer science and engineering by addressing the critical issue of hate speech on social media and proposing an effective solution based on modern machine learning techniques.

Keywords: Hate Speech; Offensive Language; Machine Learning; Neural Network; Social Media, Recurrent, Convolutional;

Acknowledgement

Firstly, we would like to express our gratitude to Almighty Allah for keeping us physically and mentally healthy enough to complete our thesis work, titled ”Enhanced Hate Speech Detection in Social Media using Transformer-based Models.” flawlessly and on time, as well as for providing us with the knowledge, skill, and resolve to complete our work.

We are thankful to our supervisor, Md. Golam Rabiul Alam, PhD, Professor, Computer Science and Engineering, for his valuable guidance, encouragement, supervision, motivation, and suggestions throughout our thesis procedure. His endless support helped us to go through our work and ideas properly.

The successful outcome of this study is not only the result of individual hard work but also the accumulation of many other people’s efforts. Without the general help of our Department of Computer Science, this study would not be complete. We would like to thank our parents, who have given hope to us and kept supporting us to pursue our careers, without which maybe we would not be able to complete our thesis work.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AUC Area Under the Curve

BLR Bayesian Logistic Regression

$BoWV$ Bag of Words Vectors

CNN Convolutional Neural Network (CNN)

CNN Convolutional Neural Network

$DCNN$ Deep Convolutional Neural Network automates

$DNNs$ Deep Neural Networks

$DRLN$ Deep Recurrent Learning Network

$ELMo$ Embeddings from Language Models

$GBDTs$ Gradient Boosted Decision Trees

HS Hate speech

$LSTM$ SVM and Long Short Term Memory

ML Machine Learning

$RFDT$ Random Forest Decision Tree

RNN Recurrent Neural Network

RNN Recurrent Neural Networks (RNNs)

ROC Receiver Operating Characteristic

$SMNs$ Social Media Networks

SVM Support Vector Machine

TF_IDF Term Frequency-Inverse Document Frequency

WoS Web of Science

Chapter 1

Introduction

”Hate speech” refers to any public discourse that displays hatred for or incites violence against a person or group based on their race, religion, sexual orientation, gender, or any other arbitrary category. The number of Web of Science (WoS)-indexed publications on hate speech (HS) increased from 42 to 162 between 2013 and 2018 due to the academic community’s growing interest in the topic. The author, Mara Antonia Paz, analyzed the amount of HS-related articles published in all domains between 1975 and 2019 [1].

Table 1.1: HS in WoS published paper between 1975-2019 [1]

Country	Documents
USA	431
England	169
Australia	51
Canada	39
Spain	35
Germany	34
South Africa	29
Netherlands	22
Brazil	21
Italy	18

Hate speech exploits stereotypes to instill hatred. The term ”hate speech” refers to ”any statement that criticizes a person or another group on the basis of a particular attribute that includes race, color, ethnicity, sexual orientation, gender, ethnicity, faith, or other characteristics.” Internet commenters utilize blog posts, discussion boards, Twitter, and Facebook to make use of their First Amendment entitlement to free expression, which protects the majority of hate speech in the United States. However, such hosted services usually restrict hate speech [2]. Yahoo! Terms of Service. 1 restricts the posting of ”information deemed unconstitutional, detrimental, dangerous, abusive, threatening, painful, defamatory, offensive, obscene, malicious, invasive of another individual’s privacy, diabolical, or racially, ethnically, or otherwise objectionable.” Facebook, on the other hand, forbids ”information that: is violent, threatening, or inappropriate; or incites violence.” No publicly accessible machine classifier can recognize hate speech; however, user inputs are usually checked for a list of harmful words [3][2].

1.1 Motivation

Social Network Sites promote damaging campaigns against specific groups and individuals. Massive online offensives cause cyberbullying, self-harm, and sexual predation. Group attacks can get violent. Using morpho-syntactical characteristics, sentiment polarity, and word embedding lexicons, the author designs and implements two Italian language classifiers using different learning algorithms: SVM and Long Short Term Memory (LSTM) from Recurrent Neural Network (RNN). They test these two learning algorithms for hate speech classification. Two categorization methods performed well on the first manually annotated Italian Hate Speech Corpus of social media text [4]. Online and offline hate speech has increased in recent years [7]. Several things cause this. According to [5], the anonymity provided by social media and the internet might lead to violent behavior. However, internet expression promotes hate speech [6]. Authorities and online communication platforms may profit from early identification and avoidance strategies, as biased speech is harmful to society. The author contributes to the solution of this challenge by evaluating field studies. They define, structure, and discover solutions. They critically evaluate theoretical and practical resources like datasets and other projects [7].

1.2 Problem Statement

Hate speech as a social issue is a well-established study topic in the fields of arts and the liberal arts, but it is new in the computing industry. To keep scientists informed, it is vital to keep them up to speed with the latest and most recent advances or advancements ([13]). Given the rapid growth of Internet users and media content, it is particularly difficult to identify the abusive language in audio and video. As humans may humorously utilize non-hateful language as hate speech and use different voice tones and display other gestures in the video than in writing, it is challenging to detect hate speech when converting audio, video, or motion into text. This study focuses on hate speech and seeks to automatically identify malicious content using data collected from multiple social media platforms. The remote supervision method will automatically generate tagged datasets and hate-polarized word embeddings.

hate speech recognition using audio and text is essential because it enables a more complete understanding of hate speech in social media. A deeper analysis may be undertaken to properly detect instances of hate speech by considering both the acoustic and textual features. The use of transformer models, like BERT, has yielded encouraging results in the identification of hate speech. These models can capture the context of the text, which is essential for recognizing the intricacies of hate speech. Furthermore, transformer-based models outperformed classical machine learning and deep learning models in terms of accuracy, precision, recall, and F-measure. They also outperform cross-domain datasets in terms of generalization. As a result, combining techniques with transformer models can dramatically improve hate speech detection capabilities. [53] [54]

1.3 Research Objective

Messages are delivered and received practically instantaneously on social media networks (SMNs), making them the quickest means of communication. Many people use social media networks for positive purposes, while others use them for negative connotations such as hate speech and trading. The focus is to look at machine learning (ML) techniques and algorithms for finding hate speech on social media (SM). Most of the time, hate speech is modeled as a text classification task [8]. ML systems have significantly contributed to hate speech identification and social media content analysis, in general, [9]. Over the past two decades, Hate Speech (HS) and cyberbullying have been the most explored topics in NLP [10]. Regarding SM data analysis for the discovery and classification of offensive comments, ML algorithms have been of significant use [11] in this respect. The advancements in ML algorithm research have had substantial effects in many domains of endeavor, leading to the development of crucial tools and models for analyzing massive quantities of data in real-world issues, such as SMNs content analysis [12].

Deep Learning (DL) is a branch of machine mastering that makes a specialty of coaching sellers how to make choices by interacting with their surroundings. It is not like conventional supervised mastering, in which agents are given categorized examples, DL agents learn through trial and mistake, receiving comments in the form of rewards or penalties for their movements [42].

- The agent's aim is to analyze top-of-the-line coverage that maximizes cumulative rewards over time. Inside the context of hate speech detection on social media, DL can be implemented to construct shrewd models able to classify content material as hate speech or non-hate speech. The agent perceives textual information, consisting of posts and remarks, as states, and takes movements by way of classifying them hence. The surroundings evaluate the agent's choices and give rewards or consequences based totally on the correctness of its classifications.

- Repeated interactions, the agent learns from its stories and adjusts its choice-making coverage to enhance detection accuracy [43]. DL's adaptability allows the model to correctly cope with dynamic and evolving patterns of hate speech, making it extra strong in figuring out harmful content. To utilize DL for hate speech detection, advanced natural language processing strategies and deep neural networks are hired to technique and constitute the textual records. The version is skilled on a huge dataset of categorized hate speech instances, getting to know how to distinguish between offensive and non-offensive content. While the dataset is trained, the DL-based hate speech detection model can be incorporated into social media structures to automatically experiment and filter out incoming content [44]. It identifies the ability of hate speech times and flags them for similar evaluation or removal via human moderators, contributing to safer and more inclusive online surroundings.

- Leveraging the interaction between the agent and the surroundings, DL allows the improvement of sophisticated hate speech detection systems that adapt and learn from real-world interactions. This proactive approach can play an essential function in curbing the unfolding of hate speech and fostering a more high-quality online network. However, making sure equity and addressing capacity biases within the education data are important aspects to keep in mind in deploying DL-primarily based hate speech detection structures. Continuous studies and refinement will be critical to harness the overall capability of DL in combatting hate speech efficiently [45].

1.4 Research Gaps

Throughout our research, we have gone into a variety of research issues that intimately connect to the fabric of our study, making it more complex, fruitful, and significant. Our investigations are methodically planned to uncover the nuances of hate speech on social media, inquiring into its impact, platform prevention tactics, and the incorporation of machine learning mitigation solutions.

The widespread use of the Internet and social media platforms, particularly among youth, has resulted in unparalleled connection as well as unexpected obstacles. Because 93 percent of youth use these platforms on a regular basis, exposure to hate speech and disinformation is a serious concern[14]. This section examines the consequences of hate speech in the digital age, investigating its origins in the period of free speech and its amplification under present social and political conditions[15].

In response to an onslaught of online vitriol, many social media sites have implemented anti-hate speech regulations. The varied nature and intricacy of these policies, however, provide a significant challenge[16][17]. This section of our research focuses on the regulatory environment of hate speech on social media, diving into content moderation teams' removal processes and providing ideas for a more uniform and successful approach[17][18].

As hate speech pervades the internet domain, technical solutions become increasingly necessary. Our research investigates a method to hate speech identification using Natural Language Processing (NLP) and Computer Vision[19]. We investigate the usefulness of machine learning in recognizing hate speech across multiple modalities [20] by applying sophisticated models such as Bidirectional Encoder Representations from Transformers (BERT) and A Lite BERT (ALBERT). This section delves into the specifics of our experiment, emphasizing the use of speech features and visual clues for a holistic approach to hate speech reduction.

This in-depth examination of the multiple features of hate speech on social media seeks to provide significant insights and ideas for a better-educated discussion of this vital societal problem. While hate speech detection has been a focal point in numerous research endeavors, a distinctive facet of our study lies in the incorporation of adversarial attacks. Adversarial attacks involve deliberately introducing subtle modifications to input data with the aim of deceiving machine learning models without significantly altering the human-perceivable content. The decision to employ adversarial attacks stems from the recognition of potential vulnerabilities in hate speech detection models. Many existing studies may not have explored this avenue, but our approach acknowledges the importance of assessing model robustness in the face of adversarial manipulations. Adversarial attacks simulate real-world scenarios where malicious actors may attempt to subvert hate speech detection systems for their benefit.

1.4.1 Generation of Adversarial Samples

- Adversarial samples were crafted by applying carefully designed perturbations to the input data, aiming to mislead the hate speech detection model.

- Modifications were constrained to be imperceptible to the human eye, ensuring that the adversarial nature of the samples was subtle.

1.4.2 Evaluation under Adversarial Conditions

- The hate speech detection model was rigorously evaluated using both original and adversarial samples.

- Performance metrics, including accuracy, precision, recall, and F1 score, were analyzed under normal and adversarial conditions.

1.4.3 Model Vulnerabilities

- Adversarial attacks revealed certain vulnerabilities in the hate speech detection model that were not apparent in conventional evaluations.

- Subtle manipulations in input data led to misclassifications, highlighting potential weak points in the model's discriminatory capabilities.

1.4.4 Robustness Enhancements

- Insights gained from adversarial attacks were used to enhance the model's robustness.

- Countermeasures, such as adversarial training and input preprocessing, were explored to mitigate the impact of adversarial manipulations.

1.5 Thesis Organization

The remainder of this work is arranged as follows. The relevant literature on the usage of the hate speech detection model is examined in Chapter 2. The approaches for the models used in this work are described in Chapter 3. The specifications are provided in Chapter 4. Chapter 5 illustrates the performance analysis and assessment process. Chapter 6 provides an explanation of the models. The report concludes in Chapter 7 with a discussion of limits and future research.

Chapter 2

Literature Review

Hate speech recognition on Twitter is critical for purposes that include disputed event extraction, AI chatterbot building, and sentiment research. This study investigated deep neural network architectures for detecting hate speech. They also experiment with other classifiers, such as Logistic Regression, Random Forest, SVMs, Gradient Boosted Decision Trees (GBDTs), and Deep Neural Networks (DNNs). Our paper's main contributions are: (1) Investigate deep learning approaches for identifying hate speech. (2) Investigate several twitter semantic embeddings, including character n-grams, speech TF-IDF standards, Bag of Words Vectors (BoWV) spanning Global Vectors for Word Representation (GloVe), and specific to the job embeddings trained with FastText, CNNs, and LSTMs. (3) The approaches vastly surpass current methods. Experiments on a standard dataset consisting of 16,000 annotated tweets show that these deep learning approaches beat state-of-the-art char/word n-gram algorithms by 18 F1 points [21].

The expanding usage of social media and knowledge sharing has significantly positive effects on mankind. This research aims to analyze the performance of three feature engineering approaches and eight machine learning algorithms using a publicly accessible dataset containing three separate classes. Three feature engineering techniques were Bigram, Word2vec, and Doc2vec. The eight machine learning algorithms were Machine Learning Classifiers, Naïve Bayes, Random Forest, Support Vector Machines, K Nearest Neighbor, Decision Tree, Adaptive Boosting, Multilayer Perceptron, and Logistic Regression. Hateful tweets are publicly available, and CrowdFlower created this dataset. Hate speech and offensive and non-offensive tweets are categorized in this dataset. It has 14509 tweets. The result shows that bigram features using the support vector machine method fared best with 79 percent accuracy. This baseline study on automated hate speech detection has practical applications. Different comparisons will also be employed as state-of-the-art methodologies to compare future research against present automated text categorization algorithms [22].

Another blogger claims that hate speech has grown into a severe issue which is now prevalent on social media. As a result, this author is dedicated to developing more effective strategies to protect free expression on online platforms and in online communities, while simultaneously reducing illegal discrimination. As neural network approaches grow more advanced for text classification tasks, a strategy for improving hate speech categorization using neural networks is outlined. The approach makes use of a publicly available embedding model, which is tested against a hate speech dataset from Twitter. To ensure dependability, they compare the results to a well-known sentiment dataset. The authors are satisfied with the approximately 5-point improvement in F-measure (mean of sub-models, the standard deviation of sub-models, mean of ensembles, the standard deviation of ensembles, and best results from original author) between the suggested method and the previous investigation using a dataset that is freely accessible for evaluating hate speech [23]. This research demonstrated how the three types of text classification approaches, Embedded systems from a Language Model (ELMo), Bidirectional Encoder Representation from Transformers (BERT), and Convolutional Neural Network (CNN), function and then applied them to identify hate speech. The performance was then enhanced by combining the findings of ELMo, BERT, CNN, and three CNN-based classifiers with varying learning rates. The fifth assignment of SemEval 2019 requires you to apply these approaches to the data. Then, using fusion procedures, merge the classifiers to increase overall classification performance. The findings indicate that the categorization is significantly more precise and has a higher F1 score [24].

The terms unigram and bigram, the number of hostile words and terrible words, and the quantity of words having an adverse disposition were used in the research on identifying hate speech aimed toward religions in the Indonesian language. NB and SVM were selected as the two approaches to be compared. This program also created a hate speech lexicon to determine the quantity of hate speech-related terminology and phrases. The resultant vocabulary was of poor quality due to an uneven quantity of tweets on religion that were not classified as non-hate speech against religion. This made the dictionary more appropriate as a dictionary of terms and phrases linked to religion than as a vocabulary of words associated with hate speech [25][26].

This research aims to locate instances of hate speech in Indonesian and to compile a new dataset that contains all forms of hate speech, including speech that is hostile toward religion, racial or ethnic groups, gender, or sexual orientation. In addition, we carried out some exploratory research utilizing a technique known as "machine learning." They examined the effectiveness of a variety of characteristics and machine learning techniques when it came to locating hate speech. Some of the qualities that were eliminated are word n-grams with n=1 and n=2, character n-grams with n=3 and n=4, and negative emotion. The data was sorted using many different algorithms, including Naive Bayes, (SVM), Bayesian, Random Forest Decision Tree, and Logistic Regression. An F-measure of 91.75 percent was attained using the Random Forest Decision Tree method in conjunction with the word n-gram feature. In addition, the results demonstrate that the character n-gram feature did not perform as well as the word n-gram feature. In addition, their findings differed from the findings of two other research that investigated the detection of hate speech in English. It was stated that character n-grams were superior to word n-grams, however, our research showed that the contrary is really the case. It was claimed that Random Forest Decision Tree(RFDT), Bayesian Logistic Regression(BLR), and SVM were all similarly effective in identifying hate speech; however, our research revealed that SVM was much less effective than RFDT and BLR [26].

Hate speech is typically defined as poking fun of an individual or group because of their ethnic background, color, ethnic background, sexual orientation, gender, nationality, faith, or another characteristic. This report presents a survey on how to detect hate speech automatically. Simple Surface characteristics, Word Generalizing, sentiment evaluation, Lexical Assets, Morphological Features, Based on knowledge Features, Meta-Information, and Information are all distinct sets of characteristics examined in diverse publications. The classification methods mainly focus on supervised learning. Different kinds of surveys show that their data comes from social media, different types of comments, and many different kinds of videos. Hate speech is found through supervised learning, which looks at all these data to find text or speech. It's hard to say how effective many of the complex features are as a whole because they are usually only judged on individual data sets, most of which are not available to the public and often only deal with one type of hate speech, like bullying of certain ethnic minorities. They say there should be a benchmark data set for detecting hate speech so that different features and methods can be better compared [27].

The majority of attempts to discover messages of hatred these days have centered on English text, with little attention paid to Arabic. In this study, the author created an ordinary Arabic dataset that may be used to identify hate speech and abuse. This work provided an OSN sample for Arabic expressions of hatred detection. Three Arabic annotators painstakingly classified twenty thousand Instagram, YouTube, Facebook, and Twitter posts, comments, and tweets into two distinct balanced categories: hate and non-hate. This could be the first collection of Arabic hate speech from many platforms. Twelve machine learning algorithms and two neural networks (CNN and RNN) were employed to assess the dataset's performance. Twelve machine learning algorithms and two deep learning designs (CNN and RNN) were employed to assess the dataset's performance. Complement NB outperformed other machine learning algorithms, achieving 97.59 percent accuracy. RNN outperformed other neural network architectures, getting 98.70. In the future, we want to expand the dataset to include user actions, likes, sentiments, and answers. Collect data from many Arabic places to encompass languages and cultures, and identify disability hate speech [28].

There was another author, and together they gave a detailed survey of a wide range of problems that arise when detecting hate speech. The research has been done by putting these problems into three main groups: the level of data, the level of models, and the level of people. These categories are broken down into more specific subcategories, which are then looked at by giving examples. The study shows that the problem of how hate speech spreads and how to find it is still difficult and needs to be dealt with in the right way to get good results [29].

Cyberbullying, hate speech, and other issues arose as Internet use increased. This article discusses Twitter's hate speech issues. Hate speech appears to anger individuals and promote hate ideology through misconceptions. Hate speech targets protected categories like race, religion, handicap, and gender. Hate speech may depress people and lead to bad behavior. So, monitor user posts and eliminate hate speech before it spreads. Twitter receives almost 600 tweets each second and 500 million per day. Hand-filtering such a massive volume of incoming traffic is relatively easy. Here, the (DCNN). The proposed DCNN model leverages the Twitter text and GloVe embedding vector to determine tweet content via convolution. The best accuracy, recall, and F1-score values were 0.93, 0.81, and 0.90, better than the previous models[30].

Hate speech on social media is a grave and urgent challenge, encompassing conversation that incites violence, discrimination, or hostility toward individuals based on race, ethnicity, faith, gender, sexual orientation, incapacity, or different traits. Different factors make contributions to its prevalence, such as anonymity, the fast dissemination of content material, inadequate moderation, algorithmic polarization, and ingrained societal biases. The repercussions of hate speech are well-sized, main to actual-international harm, cyberbullying, and even radicalization. Addressing this issue requires a multifaceted approach. Robust content material moderation and network hints are crucial to filtering hateful content and fostering more secure online surroundings. Person reporting mechanisms empower the network to flag and report offensive material, ensuring a collective effort in fighting hate speech. Furthermore, leveraging superior technology like Deep Reinforcement gaining knowledge of (DRL), and system mastering is essential in proactively identifying and disposing of hateful content material [31] [32]. The ultimate purpose is to create a digital area that is both safe and inclusive. Preventing hate speech isn't always about stifling loose speech or silencing various views, however instead it is about protecting individuals from dangerous and offensive content. By way of striking a balance between freedom of expression and accountable content moderation, we are able to foster an environment where all and sundry can freely specify their reviews without selling hatred or causing harm to others. The collaborative efforts of platform directors, content material moderators, customers, and advanced technology can collectively paint toward accomplishing this intention.
A set of rules and device learning strategies offer precious solutions to deal with the problem of hate speech on social media. The ones era allowed the development of automated detection structures that could hastily discover and flag likely dangerous content material. Natural language processing algorithms study text data, even as devices getting to know models, like deep learning networks, can be trained on full-size datasets to apprehend patterns and characteristics of hate speech. Keyword filtering enables block offensive phrases and terms, stopping the without delay spread of dangerous content material fabric. Sentiment evaluation algorithms decide the emotional tone of posts, pinpointing people with competitive or terrible sentiments that could propose hate speech. Real-time monitoring allows non-prevent scanning of social media, permitting quick responses to growing hate speech tendencies. Additionally, network reporting mechanisms empower customers to report hate speech, prompting in addition human evaluation. By combining algorithmic strategies with human moderation, structures can paintings closer to growing more secure and extra-inclusive online areas, putting a balance among freedom of expression and curtailing the dissemination of hate speech.

Deep Reinforcement studying (DRL) has emerged as an effective and promising technique for addressing complicated and dynamic problems in numerous domains [33]. In recent years, it has received interest for its potential to detect hate speech on social media systems, where harmful and offensive content can unfold swiftly, main to tremendous actual-world results. DRL, a subfield of gadget-gaining knowledge, includes an agent interacting with an environment to learn optimum choice-making policies. Inside the context of hate speech detection, the environment consists of textual data, inclusive of posts, remarks, and messages on social media structures [34] [35]. The agent's objective is to categorize these textual inputs into classes: hate speech and non-hate speech. The agent perceives the textual information as states, in which every nation corresponds to a bit of textual content. It takes action by means of classifying the text as either hate speech or non-hate speech. Primarily based on its classifications, the agent receives rewards or penalties from the environment. Those rewards function remarks to guide the agent in getting to know a policy that maximizes its cumulative praise through the years. Gaining knowledge of procedures in DRL entails an iterative technique. Through non-stop interactions with the surroundings, the agent refines its choice-making coverage to enhance its hate speech detection accuracy. This pliability permits the version to be examined from its studies, making it sturdy in addressing new and evolving patterns of dangerous content material. To allow DRL for hate speech detection, advanced herbal language processing techniques are used to preprocess and constitute the textual information [36]. Neural networks, especially deep getting-to-know architectures, are employed as function approximators to seize complex patterns and relationships inside the statistics. One common DRL technique for hate speech detection is the Deep Q community (DQN). In a DQN, a deep neural network is used to approximate the Q-feature, which predicts the predicted cumulative praise for each viable motion in a given state [38]. The agent selects movements with the very best Q-values, balancing exploration and exploitation to gain higher detection performance. Policy Gradient methods are another popular choice in DRL for hate speech detection. These techniques at once parameterize the coverage and optimize it with the usage of gradient-based total techniques, looking to maximize the anticipated cumulative reward. Policy Gradient methods can manage continuous action areas and provide good pattern performance. The implementation of DRL for hate speech detection involves schooling the agent on a huge dataset of classified textual information, wherein human moderators have already identified hate speech times. The version is fine-tuned through reinforcement gaining knowledge of, where it interacts with the environment and learns from the rewards received.

As soon as educated, the DRL-based totally hate speech detection model can be deployed on social media platforms to experiment and filter incoming content material in actual time. It identifies potentially dangerous content and flags it for similarly human evaluation or elimination, ensuring a more secure and extra inclusive online environment [37].

Text-to-audio generation, commonly known as Text-to-Speech (TTS), is a transformative technology that converts written text into spoken words. At its core, TTS systems model the intricate processes of human speech production, aiming to generate natural and intelligible voice outputs. Early TTS systems relied on concatenative synthesis, where pre-recorded speech fragments were stitched together to produce the final audio [51]. However, the advent of deep learning has revolutionized TTS, leading to the development of models like WaveNet and Tacotron that generate speech directly from text using neural networks [52]. These models have significantly improved the naturalness and fluency of synthesized speech, making it nearly indistinguishable from human voices in some cases. As TTS technology continues to evolve, its applications span diverse domains, from assistive technologies for visually impaired individuals to voice assistants and multimedia content generation.

Feature extraction from audio is a pivotal process in audio signal processing, aiming to distill the raw, complex waveform into a more concise representation that captures its essential characteristics. One of the most prominent features extracted from audio is the Mel-Frequency Cepstral Coefficients (MFCCs), which represent the short-term power spectrum of sound and are particularly influential in speech and audio recognition tasks [48]. Another significant feature is the Spectral Contrast, which gauges the amplitude difference between peaks and valleys in a sound spectrum, proving useful for tasks like music genre classification [49]. Chroma features, which pertain to the twelve distinct pitch classes, have found applications in chord recognition and music analysis. Other features like the Zero-Crossing Rate, which measures signal polarity changes, and Spectral Roll-off, indicating the frequency below which most spectral energy is contained, further enrich the feature set used in distinguishing between different audio content types and detecting events within audio streams [50]. In essence, feature extraction transforms the intricate audio signals into a format that's more amenable to analysis, with the chosen features often tailored to the specific application and nature of the audio data.

In the initial phase, it's crucial to preprocess the audio. This preprocessing can involve converting the audio to a standard format, such as WAV, removing any background noise, and normalizing the volume. This ensures that the audio is in the best possible state before being fed into the ASR system [46].

When it comes to ASR systems, there are several options available. Commercial solutions like Google Cloud Speech-to-Text and IBM Watson Speech to Text offer cloud-based recognition with support for multiple languages. On the other hand, open-source solutions like Mozilla's DeepSpeech, which is based on deep learning, have gained popularity due to their flexibility and adaptability. DeepSpeech, in particular, leverages neural networks to improve its accuracy, a technique that has been discussed extensively in recent literature[47]. Once the audio is transcribed using the chosen ASR system, the resulting text can be searched for the desired word or phrase. It's worth noting that the accuracy of the transcription can vary based on the quality of the audio and the ASR system's proficiency. Therefore, post-processing might be required to correct potential transcription errors or refine the search results.

Despite its capacity, DRL for hate speech detection faces challenges consisting of biased education facts and antagonistic assaults. Making sure equity and robustness inside the version's decision-making is crucial to keep away from unintended biases and malicious manipulations. DRL offers a promising method to detecting hate speech on social media systems. By way of leveraging the interplay among an agent and the environment, DRL models can learn to differentiate among hate speech and non-hate speech efficiently [39]. Their adaptive nature and potential to seize complicated patterns cause them to nicely-perfect for addressing the dynamic and evolving nature of dangerous content material on social media. Integrating DRL-based totally hate speech detection systems into social media structures can play a crucial position in proactively figuring out and removing offensive content, growing a more secure and more inclusive online area for users. But, cautious attention to biases and opposed robustness is important to make certain fair and reliable hate speech detection. Persisted studies and improvement on this discipline might be essential to harness the entire capability of DRL in fighting hate speech and fostering a more fit online environment. Deep Reinforcement Learning (DRL) is a framework that combines Markov Decision Processes (MDPs) with deep neural networks. In this framework, an agent interacts with an environment modeled as an MDP, which consists of states, actions, transition probabilities, and immediate rewards [40] [32]. The agent observes the current state, selects an action based on its policy, and receives a reward from the environment. The aim of the agent is to learn a policy that maximizes the expected cumulative reward over time. DRL utilizes deep neural networks, such as Deep Q Networks (DQNs) or Policy Gradient networks, to approximate the action-value function or policy function. The parameters of these neural networks are updated using algorithms like Q-learning or Policy Gradient algorithms, enabling the agent to learn and adapt its decision-making policy based on observed rewards and state transitions [41]. By continuously interacting with the environment and using deep neural networks, DRL can effectively handle complex and high-dimensional problems, making it a powerful approach for applications like hate speech detection on social media platforms.

Several research studies have looked at hate speech detection using audio and text input. Deep learning algorithms have been proposed by researchers to integrate audio and textual components for identifying hate speech in languages such as Amharic [55] [56]. They collected audio data from YouTube videos and used the Google Speech-to-Text API to convert it to text. Word2vec was used to extract textual features, while Mel-Frequency Cepstral Coefficient (MFCC) was used to extract acoustic data. A multi-modal model was created using four deep learning algorithms: LSTM, BILSTM, GRU, and BIGRU. The results revealed that the multi-modal model with BILSTM outperformed other tests, detecting Amharic hate speech with an accuracy of 88.15% [57]. Efforts have also been made to discover hostile memes using analysis, which combines visual and linguistic clues utilizing lightweight architectures and classification models [58] [59].

Chapter 3

Methodology

Hate speech detection with machine learning entails building a model to identify and classify hostile or offensive text or speech. The data sets and the planned study are introduced and described in detail. Explain how you got your hands on this info and what you did with it. A comprehensive overview of the paper's data sets and proposed work is provided. Describe the data collection process and the data set. Below is a detailed diagram of the workflow.

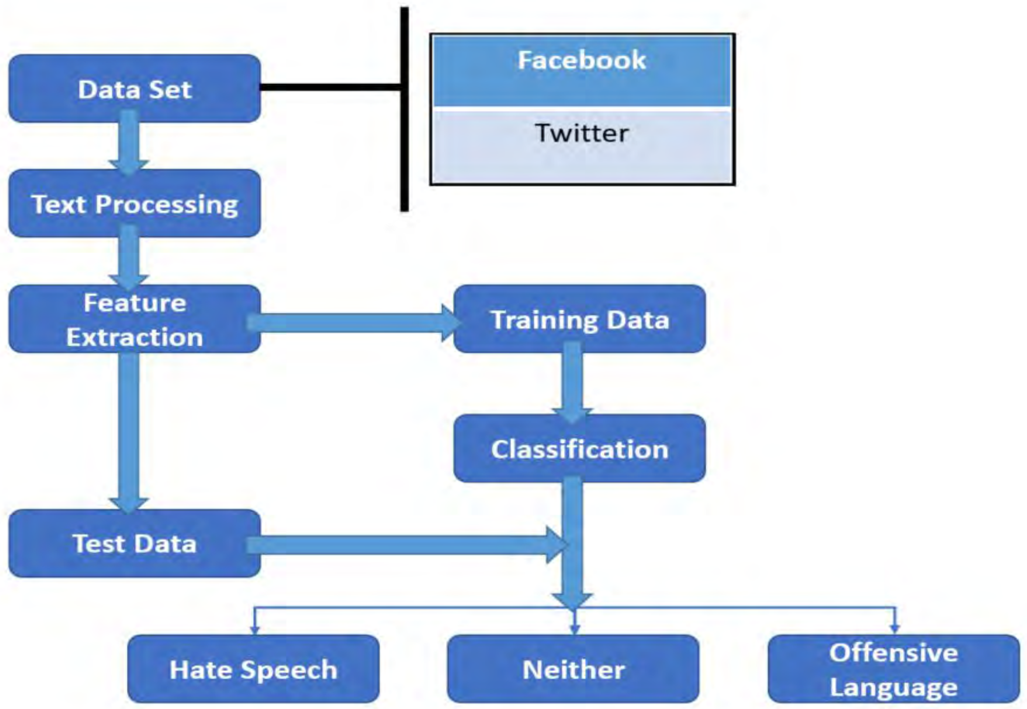

Figure 3.1: Workflow for Text-based Hate Speech Detection

3.1 Data Set

In this section, we delineate the comprehensive data collection process undertaken to construct the datasets utilized in this study. Our efforts encompassed the acquisition of textual data from social media platforms, specifically Facebook and Twitter, and the subsequent conversion of this audio information into text files.

3.1.1 Text Data Collection

The objective of this phase is to curate a dataset containing instances of hate speech, with a focus on identifying content that may exhibit racist or sexist sentiments. We conducted the data collection in adherence to a structured approach outlined below:

Data Source Selection

We constructed a dataset by combining three frequently used hate speech detection datasets: the OLID dataset [61], the White Supremacy Forum [60], and the AHSD dataset [62]. Our preparation included removing texts that included no terms from the provided list, marked as L. The resulting dataset contains 27,368 messages, 4,818 of which are classified as normal and 22,550 as hate speech. Following that, we executed a 4:1 random split into training and test sets. Each experiment was iterated five times to ensure robustness, using different random seeds for variety.

Data Crawling and Sampling

We used data sourcing methods to gather the information, mostly depending on GitHub rather than web scraping techniques such as the Facebook Graph API and other similar tools because of the inherited meaning of various words. Our data-gathering technique was designed to include posts and comments from a variety of topical topics, guaranteeing a complete representation of user-generated material.

Data Processing for Hate Speech

Several main functions are used in the data processing step to prepare and supplement the dataset for hate speech identification. The following are the major functions:

- Estimating Probability - The get_prob_dict function was used to retrieve the probability distribution of each word in the dataset. specified the whole dataset, this function computes the probability (P(h')) of each word in a specified list. It uses a Counter object to count word occurrences and then computes the probability.

- Data Augmentation: - We used the get_augmented_tweets_with_prob function to enrich the data. This method substitutes words in a given tweet with terms from a list provided. The replacement is carried out based on the predicted probabilities, ensuring that the adjustments are diverse and contextually relevant. Furthermore, the function allows for word misspellings, which adds to the variety of the supplemented dataset.

- Misspelling Function - The misspell_hw function creates misspellings for hate words, adding spelling variances to improve the model's resilience.

- Misspelling Dictionary - The get_misspell_dict method generates a dictionary that maps original hate words to misspelled variants, easing the misspelling augmentation process.

- Replacement and Misspelling of Test Data - The replace_test_hw and misspell_test_hw routines replace or misspell hate words in the test dataset. These features imitate real-world settings in which hate speech may include misspellings or the usage of synonyms.

We want to improve the diversity and complexity of our dataset by using these data processing methods, which will provide a solid foundation for training and assessing our hate speech detection model. The use of probability-based word substitution and misspelling provides subtle variations, which improves the model's capacity to generalize across various forms of hate speech in both text and audio modes.

Privacy and Ethical Considerations

Adhering to stringent privacy and ethical standards, we took measures to safeguard user identities. Both commenter and original poster names were meticulously removed from the dataset, ensuring the privacy and anonymity of the individuals involved. These steps underscore our unwavering commitment to data integrity and privacy protection throughout this study.

3.1.2 Audio-to-Text Conversion for Hate Speech Detection:

An important part of our technology for hate speech identification is the translation of audio data into text. The approach is critical for seamlessly integrating both textual and aural data for a thorough study of hate speech. Our methodology is outlined by the approaches listed below:

Speech Recognition Library Integration

We used the SpeechRecognition package to make it easier to convert audio files to text. This sophisticated library provides a variety speech recognition engines, and we chose the Google Web Speech API for its strong performance.

Audio-to-Text Conversion Function

The audio-to-text conversion mechanism is encapsulated in the convert_audio_to_text function. This method handles ambient noise with the SpeechRecognition recognizer, records the audio, and then uses the Google Web Speech API for transcription.

Integration with Google Drive

The solution offers capabilities to download audio files from Google Drive to the local Colab environment, improving accessibility and integrating audio data seamlessly into the hate speech detection pipeline.

Audio File Processing with FFmpeg

To ensure compatibility and efficient audio file processing, the code involves the installation of FFmpeg, a multimedia processing program.

Formatted Path and Text Conversion

The path to the converted audio file is prepared for further examination. To retrieve the transcribed text from the audio, the convert_audio_to_text function is called. Finally, the identified text is presented, bringing the audio-to-text conversion process to a close. This transcribed material is incorporated into our dataset for hate speech detection.

Quality Assurance

To maintain the highest standards of data quality, we performed periodic checks on the converted audio files. This involved verifying that the audio accurately reflected the corresponding text content, with particular attention to any subtle variations or nuances that might carry additional meaning.

3.1.3 Data Consistency

Efforts were made to ensure that the audio data maintained consistency with the text data. Each audio file corresponded to a specific text record, facilitating a cohesive analysis of hate speech and offensive language across both modalities.
This robust and efficient audio-to-text conversion process guarantees that audio data is seamlessly integrated into our hate speech detection framework, boosting the model's capacity to evaluate and contextualize hate speech across multiple modalities.

3.2 Proposed Work

In our research, we aim to address the pervasive issue of hate speech, which manifests in both textual and auditory forms. Our approach is bifurcated into two primary sections: text-based and audio-based hate speech detection. Each of these methods is designed to tackle the unique challenges posed by their respective mediums. Then observing the complex analysis we try to build a multi-modal transformer-based Hate Speech Detection method.

3.2.1 Text-Based Hate Speech Detection

The plan of action drawn out to conduct the research included a number of steps. First, we will conduct an analysis of how hate speech was communicated in a general sense as well as what material we ought to concentrate on. After conducting research, we discovered that there are three primary modes by which hate speech is communicated. The following tweets and posts from Facebook and Twitter that contain mixed language, inflammatory language, and hate speech were then collected. A dataset is created by combining all of them together. We compiled the dataset, and then we turned it into a variety of other sorts of data. From each distinct kind of dataset, we extracted the determining elements and attributes. The next step was to collect features, after which we used a number of different machine learning algorithms and deep learning methods to determine whether or not the content was hateful. We used the results from each of these algorithms and methods in a unique way and then combined them to get the final result. The figure shows the whole process of prediction of the data set:

3.2.2 Audio-Based Hate Speech Detection

The process for audio-based hate speech detection is more intricate due to the complexities of auditory data. Initially, we embarked on a data collection phase, sourcing audio files that potentially contained hate speech. This was followed by a preprocessing step, where the audio files were cleaned, normalized, and readied for feature extraction. In the feature extraction phase, key auditory signatures indicative of hate speech, such as tone, pitch, and specific phonetic patterns, were identified.
Once the features were extracted, we proceeded to the model training phase. Here, the extracted features were fed into machine learning algorithms to train them to recognize hate speech patterns. Post-training, the models underwent a rigorous evaluation phase to ascertain their accuracy and reliability. Successful models were then deployed in real-world scenarios for hate speech detection.
The deployment phase is crucial. Here, the models actively scan and analyze audio data in real-time or batch-processing modes to detect hate speech. Any detected hate speech triggers the feedback loop, where the model's decision is either reinforced or corrected based on the accuracy of its detection. This continuous feedback ensures that the model remains updated and evolves with changing patterns of hate speech.

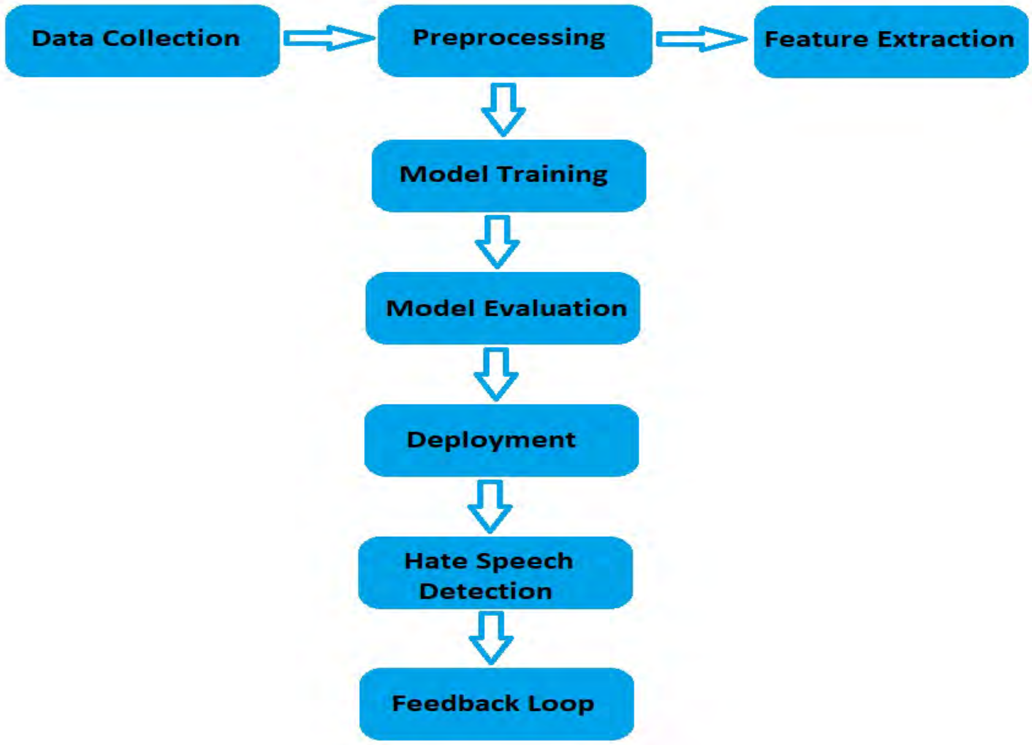

Figure 3.2: Workflow for Audio-based Hate Speech Detection

3.2.3 Data Processing for Transformer based model

The process incorporates downloading an audio file from Google Drive, loading it, adjusting for ambient noise, recording the adjusted audio, converting it to text using the SpeechRecognition library, and displaying the transcribed text output, indicating the successful audio-to-text conversion process. The script then uses the Google Web Speech API to convert the audio into text. The workflow diagram illustrates the execution of a process for detecting hate speech. It starts by calculating the probability of each word's occurrence in the dataset, P(h'), which is crucial for data augmentation. The script then augments tweets by replacing words with random words, introducing diversity and variability. A misspelling dictionary is generated for hate words, enhancing the model's robustness. The code then replaces hate words in test data with random words, showcasing variations in hate speech expressions.

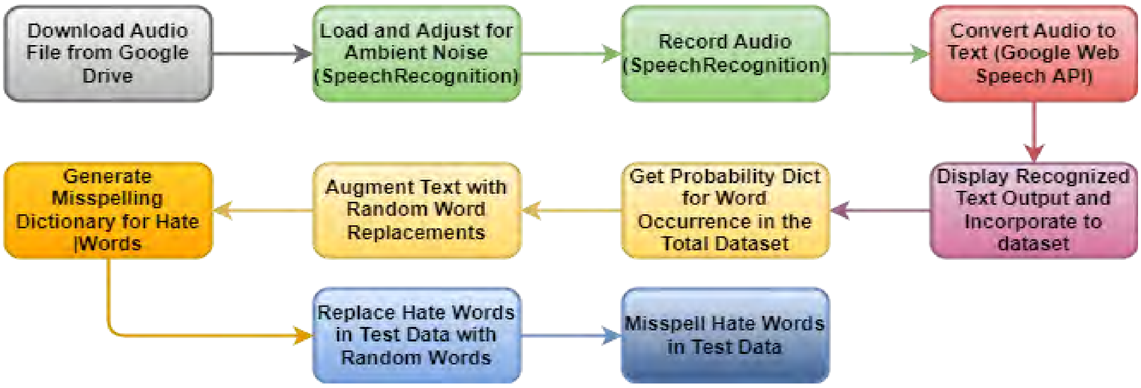
Download Audio File from Google Drive
Load and Adjust for Ambient Noise (SpeechRecognition)
Record Audio (SpeechRecognition)
Convert Audio to Text (Google Web Speech API)
Display Recognized Text Output and Incorporate to dataset
Get Probability Dict for Word Occurrence in the Total Dataset
Augment Text with Random Word Replacements
Generate Misspelling Dictionary for Hate |Words
Replace Hate Words in Test Data with Random Words
Misspell Hate Words in Test Data

Figure 3.3: Data Processing for Transformer based model.

Chapter 4

Model Specification

4.1 Machine Learning

Machine learning (ML) is a subfield of artificial intelligence that studies strategies that allow computers to acquire knowledge from data requiring explicitly programming it. In basic terms, an ML model detects similarities in data to generate accurate predictions or judgments. The procedure includes conditioning the model using data, assessing its efficacy, and using variables that were provided (features) to foresee an output (goal). There are three forms of machine learning: supervised training (using labeled data), unsupervised training (using unlabeled data), and reinforcement programming (in which the model learns via interaction and feedback). The mathematical foundation of machine learning includes likelihood, linear algebra, data analysis, calculus, as well as data theory. These fundamentals are critical for optimizing and understanding algorithm behavior. In the setting of discriminatory language detection, the goal is to determine if a particular piece of writing or audio includes hate speech. Audio data is frequently converted into characteristics such as spectrograms prior to analysis. Dan Jurafsky and James H. Martin's "Speech and Language Processing" provides insights into natural language processing, whereas Ian Goodfellow, Yoshua Bengio, and Aaron Courville's "Deep Learning" explores the complexities of deep learning. The work "Automated Hate Speech Detection and the Problem of Offensive Language" by Davidson et al. (2017) addresses the difficulty of identifying hate speech. It's imperative to approach hate speech detection with an awareness of ethical considerations, as biases in training data can lead to skewed or unjust classifications.

4.2 Text-based Detection Experiment

4.2.1 Supervised Learning

Supervised training is a fundamental technique to machine learning in which a model is trained on a dataset that includes both the input information as well as the expected outputs or labels. The model's goal in learning from this labeled data is to develop a connection among both inputs and outputs, permitting it to make projections on previously unknown data. The basic concept of supervised learning is iteratively modifying a model's inner parameters in order to reduce the discrepancy between predictions and actual labels within the set used for training.

Given a dataset with m examples, each input is represented as a vector $\mathbf{x}^{(i)}$ and has a corresponding label $y^{(i)}$. The model's predictions are given by the hypothesis function $h_\theta(\mathbf{x}^{(i)})$. The objective is to find parameters θ that minimize the overall error or loss, often represented as:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} L(y^{(i)}, h_\theta(\mathbf{x}^{(i)})) \tag{4.1}$$

where L is a suitable loss function.
When applying supervised learning to hate speech detection, the task becomes a binary classification problem: categorizing text or audio as either hate speech or not. The labeled dataset provides examples of both categories, and the model learns the distinguishing features and patterns associated with hate speech. Feature extraction becomes pivotal, transforming raw text or audio into a numerical format that can be processed by the model. Once features are extracted, the model is trained to recognize and differentiate between hate speech and non-hate speech patterns.
For a text with n unique words or features, its representation can be a vector $\mathbf{x} \in R^n$. The label, indicating hate speech, is y, where $y \in \{0, 1\}$ (0 for non-hate speech and 1 for hate speech). The loss function for binary classification, such as logistic regression, can be:

$$L(y, h_\theta(\mathbf{x})) = -y \log(h_\theta(\mathbf{x})) - (1 - y) \log(1 - h_\theta(\mathbf{x})) \tag{4.2}$$

Optimization techniques, like gradient descent, adjust the model's parameters θ to minimize this loss, with the update rule:

$$\theta_{\text{new}} = \theta_{\text{old}} - \alpha \nabla_\theta L \tag{4.3}$$

where α is the learning rate.

4.2.2 Support Vector Machine (SVM)

Support Vector Machines (SVMs) are a collection of supervised learning techniques used for regression and classification. At their heart, SVMs seek to identify the region of space that best separates a dataset into classifications. The ”support variables” are the data values closest to the hyperplane and the most troublesome to categorize, impacting the hyperplane's location and orientation. The capacity of SVMs to convert the input space into a space with more dimensions using kernel functions enables non-linear classification.
Given a dataset with data points $\mathbf{x}_i$ and labels $y_i \in \{-1, 1\}$, the objective of SVM is to find the optimal hyperplane defined by weights $\mathbf{w}$ and bias b that maximizes the margin between the two classes. The decision function is given by:

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b \tag{4.4}$$

The optimization problem can be formulated as:

$$\min_{\mathbf{w},b} \frac{1}{2} \|\mathbf{w}\|^2 \tag{4.5}$$

subject to the constraints:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \forall i \tag{4.6}$$

where the constraint ensures that all data points are correctly classified outside the margin.
When applying SVMs to hate speech detection, the goal is to classify text data as either hate speech or not. Each piece of text is transformed into a feature vector, often using techniques like TF-IDF or word embeddings. The SVM then learns the optimal hyperplane that separates the hate speech examples from the non-hate speech ones in this feature space. Given the high-dimensional nature of text data and the potential non-linear boundaries between hate speech and non-hate speech, kernelized SVMs, which implicitly map data to a higher-dimensional space, are often preferred for this task.
For non-linear classification, SVMs employ kernel functions to implicitly map the input data into a higher-dimensional space. A popular choice is the Radial Basis Function (RBF) kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2) \tag{4.7}$$

where γ is a parameter controlling the shape of the decision boundary. In the transformed space, the decision function becomes:

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{4.8}$$

where α_i are the Lagrange multipliers obtained from solving the dual optimization problem, and m is the number of training examples.

4.2.3 Logistic Regression

Logical regression is a method of statistical analysis that is commonly used for problems related to binary classification. Unlike the method of linear regression, whose predicts values that are continuous, logistical regression predicts the likelihood that a given occurrence falls into a certain category. This probability is estimated by the model using a logistic function, which ensures that the outcome is between 0 and 1. Typically, the selection boundary is set at a value of 0.49 with examples having a probability larger than this threshold assigned to one class and those with a probability less than this level assigned to another.
Given an input feature vector $\mathbf{x}$, the logistic regression model computes a weighted sum of the features, $z = \mathbf{w}^T\mathbf{x} + b$, where $\mathbf{w}$ is the weight vector and b is the bias term. This sum is then passed through the logistic (sigmoid) function to produce the probability p that the instance belongs to the positive class:

$$p = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{4.9}$$

The model is trained by adjusting $\mathbf{w}$ and b to maximize the likelihood of the observed data, which is equivalent to minimizing the logistic loss.
In the context of hate speech detection using Logistic Regression, the text data is first transformed into a numerical representation, such as TF-IDF scores or word

embeddings. Each piece of text is then associated with a probability of being hate speech. By setting a threshold, typically 0.5, the model classifies texts with probabilities above the threshold as hate speech and those below as non-hate speech. The strength of logistic regression lies in its simplicity and interpretability, making it a popular choice for binary classification tasks like hate speech detection.
For a given text represented by $\mathbf{x}$, the probability p that it is hate speech is given by:

$$p = \sigma(\mathbf{w}^T \mathbf{x} + b) \tag{4.10}$$

The model's parameters $\mathbf{w}$ and b are learned by minimizing the logistic loss over the training data:

$$L(y, p) = -y \log(p) - (1 - y) \log(1 - p) \tag{4.11}$$

where y is the true label (1 for hate speech and 0 for non-hate speech). The optimization is typically performed using methods like gradient descent.

4.2.4 Naive Bayes

The naive Bayes method is a stochastic method of categorization that uses Bayes' theorem and makes the assumption that features are independent. This ”naivety” relates to the technique's simplistic assumption that each variable contributes separately to the likelihood of an outcome, no matter the other features' values. Despite its straightforward nature and naïve assumptions, naïve Bayes may be quite successful, particularly in text classification applications, because to its capacity to handle a huge number of features.
Given a set of features $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and a class label C, Bayes' theorem states:

$$P(C|\mathbf{x}) = \frac{P(\mathbf{x}|C) \times P(C)}{P(\mathbf{x})} \tag{4.12}$$

In the context of classification, we're interested in finding the class C that maximizes $P(C|\mathbf{x})$. Using the naive independence assumption, the likelihood term $P(\mathbf{x}|C)$ can be decomposed as:

$$P(\mathbf{x}|C) = \prod_{i=1}^{n} P(x_i|C) \tag{4.13}$$

When applying Naive Bayes to hate speech detection, the algorithm calculates the probability of a text being hate speech based on the occurrence of individual words or n-grams within it. Each word or n-gram contributes independently to the overall probability. Texts are then classified as hate speech or non-hate speech based on which category has a higher posterior probability. Due to its efficiency and scalability, Naive Bayes is particularly suited for high-dimensional datasets, like those encountered in text classification tasks.
Given a text represented by a set of words $\mathbf{w} = (w_1, w_2, \ldots, w_m)$, the probability that it belongs to a class C (e.g., hate speech) is proportional to:

$$P(C|\mathbf{w}) \propto P(C) \times \prod_{j=1}^{m} P(w_j|C) \tag{4.14}$$

To classify the text, we compare $P(C|\mathbf{w})$ for each possible class and choose the one with the highest probability. In practice, to avoid numerical underflow due to multiplying many small probabilities, the computations are often done in the logarithmic domain:

$$\log P(C|\mathbf{w}) \propto \log P(C) + \sum_{j=1}^{m} \log P(w_j|C) \tag{4.15}$$

4.2.5 Gradient Boosting

Gradient enhancement is an aggregate machine learning approach that creates a powerful predictive model by integrating the outputs of numerous weak learners, most often decision trees. The basic principle underlying the booster of gradients is to repeatedly add branches to the model, with each new tree correcting the mistakes committed by the preceding trees. This allows the model to continue to improve its predictions. The term ”gradient” in the context of gradient boosting alludes to the technique’s use of slope descent in order to reduce the loss function, directing the formation of new branches to locations where they will be most useful.

Given a dataset with n samples $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$, the prediction of the model after adding m trees is:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \alpha \cdot h_m(\mathbf{x}) \tag{4.16}$$

where $F_{m-1}(\mathbf{x})$ is the prediction of the model after $m-1$ trees, $h_m(\mathbf{x})$ is the prediction of the m-th tree, and α is the learning rate. The new tree h_m is trained to approximate the negative gradient of the loss function with respect to the model’s predictions.

In the context of hate speech detection using Gradient Boosting, the algorithm is trained to classify text data as either hate speech or not. Each piece of text is transformed into a feature vector, often using techniques like TF-IDF or word embeddings. The gradient boosting model then learns to differentiate between hate speech and non-hate speech by iteratively adding trees that correct the misclassifications of the previous trees. The ensemble nature of gradient boosting, combined with its ability to focus on hard-to-classify instances, makes it a powerful tool for tasks like hate speech detection where the decision boundary might be complex.

Given a text represented by a feature vector $\mathbf{x}$, the gradient boosting model updates its prediction as:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \alpha \cdot h_m(\mathbf{x}) \tag{4.17}$$

The new tree h_m is trained to fit the residuals, which are the differences between the true labels and the predictions of the model after $m-1$ trees. The residuals are given by:

$$r_{i,m} = y_i - F_{m-1}(\mathbf{x}_i) \tag{4.18}$$

for each sample i. The tree h_m is then fit to these residuals, effectively guiding the model to focus on the samples it currently misclassifies.

4.2.6 Random Forest

Random forest modeling is a method of collaborative learning that uses several decision trees to create a more precise and robust model. Each tree in the natural environment is built using a portion of the data used for training and a selected number of the features, adding unpredictability and variety to the model. When producing forecasts, the random forest combines the outputs of all individual trees, usually utilizing majority voting for tasks such as classification. This ensemble technique aids in preventing overfitting, addressing missing data, and assigning priority ratings to features.
Given a dataset with n samples, a Random Forest with B trees selects a bootstrap sample of size n (with replacement) for each tree. For each split in a tree, a random subset of k features is chosen, and the best split among those features is used. The final prediction for a new instance $\mathbf{x}$ is given by:

$$\hat{y}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} T_b(\mathbf{x}) \tag{4.19}$$

where $T_b(\mathbf{x})$ is the prediction of the b-th tree for instance $\mathbf{x}$.
In the case of recognizing hateful speech with a random forest approach, the method classifies text input as hate communication or not depending on the combined choices of several decision trees. Every single piece of information is first turned into the vector of features, which is frequently achieved by techniques such as TF-IDF or word embedding. The system of random forests then assesses each text by routing it through all of the trees in the canopy of the forest. The final categorization is decided by a majority vote of each tree. Random Forest's capacity to capture complicated decision boundaries and natural resistance to overestimation make it an appropriate candidate for hate speech identification.
Given a text represented by a feature vector $\mathbf{x}$, the Random Forest model aggregates the predictions of all its trees to make a final decision. If we denote the decision of the b-th tree as $D_b(\mathbf{x})$ (with 1 indicating hate speech and 0 indicating non-hate speech), the overall prediction is:

$$\hat{y}(\mathbf{x}) = \begin{cases} 1 & \text{if } \frac{1}{B}\sum_{b=1}^{B} D_b(\mathbf{x}) > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{4.20}$$

This formula indicates that the text is classified as hate speech if more than half of the trees in the forest vote in favor of it being hate speech.

4.2.7 K-Nearest Neighbors (KNN)

The k-nearest-neighbors algorithm (KNN) is a parametric in nature, instance-based learning technique used in classification and regression problems. KNN is based on the principle that data points that are connected ought to carry similar labels. When predicting an unknown data the point, the algorithm examines the dataset used for training for the k training instances closest to the point and delivers the most frequent output value between them for categorization or a standard deviation for regression. The "distance" among data points may be calculated in a variety of methods, the distance calculated using Euclid constituting the one that is the most used.

Given a dataset $\mathcal{D}$ with n samples and an unseen data point $\mathbf{x}$, the KNN algorithm identifies the k samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$ from $\mathcal{D}$ that are closest to $\mathbf{x}$ based on a distance metric, typically the Euclidean distance:

$$d(\mathbf{x}, \mathbf{x}_i) = \sqrt{\sum_{j=1}^{m} (x_j - x_{ij})^2} \tag{4.21}$$

where m is the number of features. For classification, the predicted label y for $\mathbf{x}$ is the mode of the labels of the k nearest neighbors.

When applying KNN to hate speech detection, the algorithm classifies text data based on the labels of its neighboring texts in the feature space. Each piece of text is first transformed into a feature vector, often using techniques like TF-IDF or word embeddings. The KNN algorithm then determines the classification of a given text by examining the labels of its k closest texts in this feature space. If the majority of these neighbors are labeled as hate speech, the text in question is also classified as such. The simplicity of KNN, combined with its ability to make decisions based on local data structures, can make it effective for tasks like hate speech detection, especially when the decision boundary is complex.

For a given text represented by a feature vector $\mathbf{x}$, the KNN algorithm finds the k training examples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$ that are closest to $\mathbf{x}$ based on the chosen distance metric. The predicted label y for $\mathbf{x}$ is then determined as:

$$y = \text{mode}\{y_1, y_2, \ldots, y_k\} \tag{4.22}$$

where y_i is the label of the i-th nearest neighbor. In the context of hate speech detection, $y = 1$ might indicate hate speech, while $y = 0$ indicates non-hate speech.

4.2.8 Decision Trees

Decision Trees are a common machine learning approach for classification and regression applications. They function by recursively partitioning data according to attribute values, producing a decision tree-like model. At every branch of the tree, the choice takes place based on a characteristic value, going down through the leaf nodes, which reflect the ultimate predictions. The choice to divide at all nodes is often made based on metrics such as knowledge gained or Gini contaminants, with the goal of maximizing class separation or reducing variation.

The decision to split the data at a node based on a feature f and threshold t can be determined using the Information Gain (IG) criterion, which is defined as:

$$\text{IG}(f, t) = \text{Entropy}(D) - \sum_{v \in \{L,R\}} \frac{|D_v|}{|D|} \text{Entropy}(D_v) \tag{4.23}$$

where D is the dataset at the current node, D_L and D_R are the datasets that result from splitting D based on the threshold t for feature f, and the entropy is given by:

$$\text{Entropy}(D) = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{4.24}$$

where p_i is the proportion of samples in D that belong to class i, and c is the number of classes.

In the context of hate speech detection using Decision Trees, the algorithm classifies text data based on decisions made on its features, which can be word frequencies, TF-IDF scores, or even embeddings. Each decision node in the tree evaluates a feature of the text and decides which subsequent node to proceed to, eventually leading to a leaf node that provides the classification. Decision Trees offer a transparent and interpretable model, making it easier to understand which features play a significant role in classifying a text as hate speech or not. However, they can be prone to overfitting, especially when the tree is deep.

For a given text represented by a feature vector $\mathbf{x}$, the Decision Tree provides a classification by traversing from the root to a leaf node based on the decisions at each node. The decision at each node n can be represented as:

$$\text{Decision}(n) = \begin{cases} \text{Go to Left Child} & \text{if } \mathbf{x}[f_n] \leq t_n \\ \text{Go to Right Child} & \text{otherwise} \end{cases} \tag{4.25}$$

where f_n is the feature being evaluated at node n and t_n is the threshold for that feature at node n. The traversal continues until a leaf node is reached, which provides the final classification.

4.3 Audio-based Deep Learning Experiment

4.3.1 Simple Feed-forward Neural Network

A Simple Feed-forward Neural Network (FFNN), sometimes known as a Feed-forward Neural Network, is a sort of artificial neural network in which the connections between the nodes (neurons) do not form a cycle. It is made up of three layers: an input layer, one or more hidden layers, and an output layer. Each neuron in one layer is linked to every neuron in the next layer. When input is supplied into the network, it goes through each layer, going through a sequence of weighted summations and activations until it reaches the output layer, where it produces a prediction. The output of a neuron in the first hidden layer may be written as follows given an input vector $\mathbf{x}$:

$$h_1 = \sigma(\mathbf{w}_1 \cdot \mathbf{x} + b_1) \tag{4.26}$$

where $\mathbf{w}_1$ is the weight vector, b_1 is the bias, and σ is the activation function (e.g., sigmoid, ReLU). This process is repeated for each layer, with the output of one layer serving as the input to the next, until the final output layer is reached.

In the context of hate speech detection using a Simple Feed-forward Neural Network, the network is trained to classify text data as either hate speech or not. Text data is first transformed into a numerical representation, often using techniques like word embeddings or TF-IDF. The FFNN then processes this numerical data, layer by layer, to produce a final prediction. The power of FFNNs lies in their ability to learn complex, non-linear decision boundaries, making them suitable for tasks like hate speech detection where the relationship between features and labels might be intricate.

For a given text represented by a feature vector $\mathbf{x}$, the FFNN processes it through its layers to produce a prediction. If we consider a network with one hidden layer, the output from the hidden layer is:

$$\mathbf{h} = \sigma(\mathbf{W}_h\mathbf{x} + \mathbf{b}_h) \tag{4.27}$$

where $\mathbf{W}_h$ is the weight matrix for the hidden layer and $\mathbf{b}_h$ is the bias vector. The final prediction $\hat{y}$ from the output layer can be represented as:

$$\hat{y} = \sigma(\mathbf{W}_o\mathbf{h} + b_o) \tag{4.28}$$

where $\mathbf{W}_o$ is the weight matrix for the output layer and b_o is the bias for the output. The activation function σ can be a sigmoid function for binary classification, ensuring the output is between 0 and 1.

4.3.2 Convolutional Neural Network (CNN)

Convolutional neural networks, more commonly are a form of deep neural network designed primarily to interpret grid-like inputs such as images and sequences. They excel at recognizing images and conversational processing. CNNs consist of layers using convolution, layers using pooling, and fully interconnected layers. Convolutional layers scan incoming data with filters (kernels) to detect local patterns. Layer pooling decreases the dimension of space, but fully connected layers generate classification results.

In a CNN, a convolutional layer performs convolution operations on the input data using learnable filters. Given an input tensor I and a filter tensor K, the output tensor O is computed by sliding the filter across the input and computing element-wise multiplications and summations:

$$O(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n) \cdot K(m, n) \tag{4.29}$$

This process is repeated for multiple filters, producing feature maps. Pooling layers typically use operations like max-pooling to reduce the spatial dimensions of feature maps.

When CNNs are used to identify hate speech, the network is trained to classify text input as hate speech or non-hate speech. CNNs are created for image data, but they may be adapted to text data by considering text as sequences of discrete symbols such as words or letters. 1D convolutions are employed in text-based CNNs to scan over sequences, collecting local patterns of words or characters. This enables CNNs to learn hierarchical text representations, from individual letters to higher-level language structures, making them useful for hate speech identification when context and patterns are essential.

In a text-based CNN, given a sequence of word embeddings or characters $x_1, x_2, \ldots, x_n$, a 1D convolution operation is applied using filters of various sizes. For each filter, the convolution is computed as:

$$c_i = \text{ReLU}(x_{i:i+k-1} * K) \tag{4.30}$$

where $x_{i:i+k-1}$ is a subsequence of x of length k starting at position i, K is the filter, and ReLU is the rectified linear unit activation function. This operation captures local patterns in the text. The resulting feature maps are then processed by fully connected layers for classification.

4.3.3 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks, also referred to as RNNs are neural networks that handle sequential data. RNNs, unlike feedforward neural networks, include interconnections that continually loop over themselves, allowing them to maintain a hidden state or memory of previous inputs. RNNs are ideal for applications that need patterns, like the processing of natural language and time sequence analysis, because of their ability to detect temporal correlations. RNNs go through information one phase at a time, modifying their internal concealed state based on its current input and previous hidden condition at each step.

At each time step t in an RNN, the hidden state h_t is updated based on the input x_t and the previous hidden state h_{t-1} using a set of learnable parameters W and U as well as an activation function ϕ:

$$h_t = \phi(W \cdot x_t + U \cdot h_{t-1}) \tag{4.31}$$

The output y_t at each time step can be computed based on h_t and is often used for various tasks such as sequence prediction or classification.

When utilized for hate speech identification, RNNs may be used to process and evaluate text input sequentially. The RNN treats text data as a sequence of words or characters, and it processes each word in the sequence while updating its hidden state. This enables the RNN to record word dependencies, which is critical for interpreting context and recognizing hate speech. Traditional RNNs, on the other hand, might suffer from the vanishing gradient problem, which restricts their capacity to capture long-term relationships.

In the context of hate speech detection, given a sequence of word embeddings or characters $x_1, x_2, \ldots, x_t$, the hidden state h_t at each time step t is updated as follows:

$$h_t = \phi(W \cdot x_t + U \cdot h_{t-1}) \tag{4.32}$$

Classification may be performed using the output y_t at each time step. To overcome the problem of vanishing gradients, RNN versions that utilize LSTM (Long Short-Term Memory) and the Gated Recurrent Unit, or GRU, have been created to better capture long-range relationships in sequential data.

4.4 Transformer Based Proposed Model Architecture

The BERT model architecture (Bidirectional Encoder Representations from Transformers) is a multi-layer bidirectional Transformer encoder. It is based on the original implementation reported by Vaswani et al. (2017) and is available in the tensor2tensor library,[object Object]. The architecture is made up of several layers (Transformer blocks) labeled as L, a hidden size denoted as H, and a number of self-attention heads denoted as A,[object Object]. Unlike some other language representation models, BERT's architecture is consistent across tasks, with only minor differences between the pre-trained and final downstream architectures,[object Object]. Because of its uniform design, BERT can be fine-tuned with only one extra output layer to generate cutting-edge models for a wide range of tasks without requiring significant task-specific architectural modifications,[object Object].

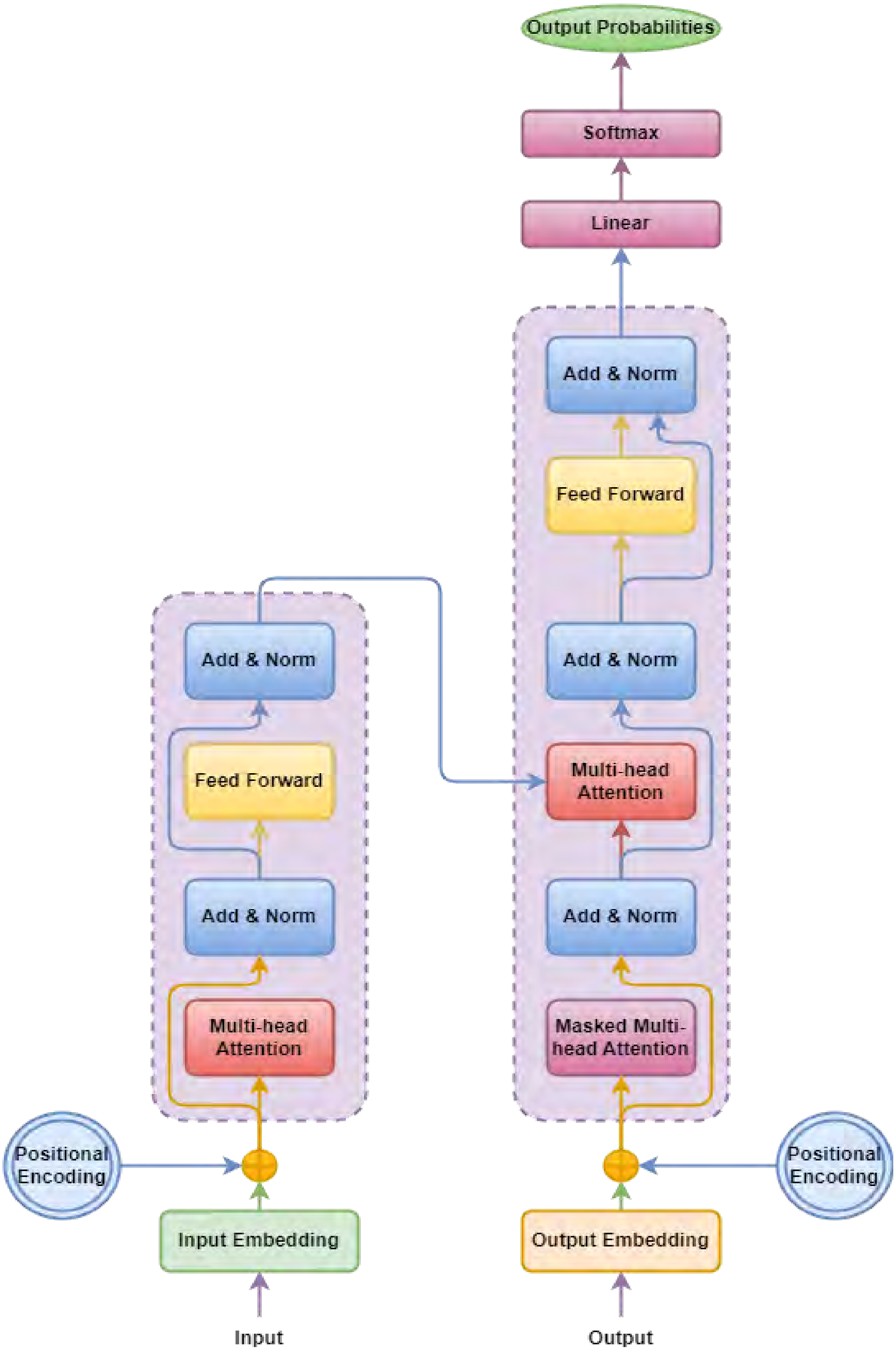

Figure 4.1: Presented a detailed overview of our proposed model

A hate speech detection model may be thought of as a functional projection from a set of input texts (T) to a set of target labels (Y), with each input text (t) corresponding to a specific label (y). The softmax probabilities for forecasting each class (k) are commonly represented by the model output, indicated as fk(t; θ) = P(Y = $yk\|t$), where denotes the model parameters. We assume the presence of a specified collection of target words (H), which often includes hostile or sentimental expressions. Let X represent the rest of the text after removing the words from H, i.e., T = X, H. Adversarial cases are instances in which detection model inputs are deliberately perturbed on H to cause errors in the model's predictions.

As Pearl [63] pioneered, causal graphs are frequently used techniques for depicting causal links among variables. These graphs are basically directed acyclic graphs (DAGs), represented by the formula G = V, E, where V represents a set of variables and E denotes the causal connections between them.

As shown in Fig. 1, we propose a causal network to explain hate speech detection in our technique. This graph contains variables X, H, and Y, and introduces I to represent a user's hatred intent. Recognizing the inherent difficulty in determining a user's actual intent, we regard I as a latent variable represented by the dashed circle in the graph.

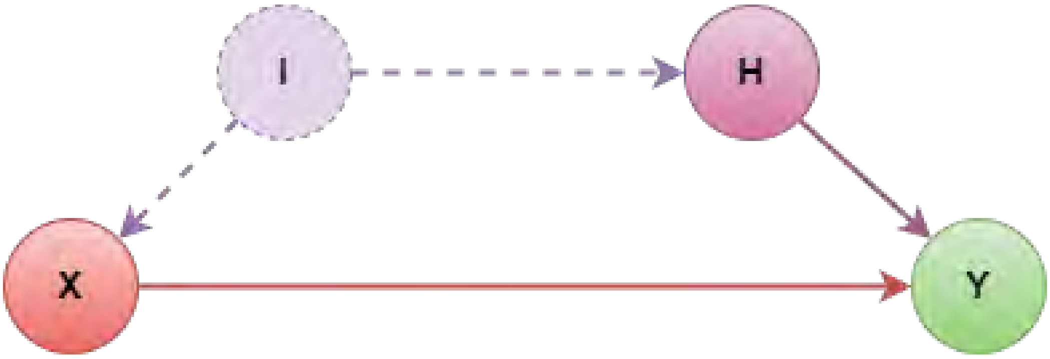

Figure 4.2: Casual Graph for Hate Speech Detection

The graph's causal links may be explained as follows: when a user is motivated to distribute hateful information, they choose target words (possibly vulnerable to subsequent changes) to communicate the hostile meaning within the remaining text. As a result, I is the parent variable of both H and X, which in turn are the parents of Y. To illustrate, given a text T, such as "We don't want more [religious group] in this country. Enough with those MAGGOTS," H corresponds to the term "MAGGOTS," and X signifies the leftover text.

We identify a key explanation for the lack of resilience in vanilla detection models against adversarial assaults using the causal network. The problem is that these models make predictions based not just on the semantic meanings of texts, but also on the accidental association between X and Y via H (i.e., X - I - H - Y). The inclusion of target words has a significant impact on this association. For example, if some target phrases, such as explicit language, show a significant association with the hate label during training, the model may become unduly reliant on predicting hatred based on the mere appearance of these words, ignoring the larger context of the entire text. As a result, when adversarial assaults interfere to disrupt such correlations, such as by eliminating the target words, the detection model becomes manipulable and prone to producing incorrect predictions.

To improve the detection model's resilience to perturbations, it is critical to prevent the model from learning erroneous correlations. The penalty we suggest for the causal influence of H on Y during training effectively prevents the emergence of such misleading correlations. The job of inferring causal impacts in machine learning is extremely difficult. We argue in this study for the use of the idea of causal strength, as established in [64]. This measure seeks to assess the impact of a causal graph intervention that removes certain arrows. In this case, we want to test how deleting the link between target terms and hate labels by changing the target words affects the outcome. The causal strength of the arrow from H to Y (H Y) reflects this.

$$L = LCE + \lambda CH \rightarrow Y = (1 - \lambda)LCE + \lambda LI \tag{4.33}$$

In the equation above, we go deeper into the understanding of the word LI. As previously stated, the artificial correlation between X and Y makes standard detection methods vulnerable to adversarial assaults. The backdoor adjustment technique, a well-established strategy for removing false correlations [63], is used to solve this issue. This method has found effective applications in a variety of tasks, including picture captioning and question answering, adding to model robustness enhancement.

The interventional distribution P(Y |do(X)) is determined via backdoor adjustment, as shown in the causal graph in Fig. 4.1.

$$P(y|do(x)) = \sum_{h'} P(h')P(y|x, h') \tag{4.34}$$

When the two formulations are compared, a striking likeness appears. This resemblance stems from the fact that both the ”arrow cutting” and the backdoor adjustment procedures disturb the route X - I - H - Y. However, using the interventional distribution P(Y ||do(X)) directly for predictions is difficult since the model's utility is dependent on the closeness of P(Y ||do(X)) to the real distribution, which is beyond of the user's control. As a result, our loss formulation in the previous equation may be considered as an extension of backdoor adjustment-based techniques that are solidly founded on the causal strength theorem.

Chapter 5

Performance Analysis

5.1 Data Analysis

Hate speech, which is defined as any kind of discourse that discriminates or promotes assault against people or groups based on characteristics such as race, faith, sexual orientation, or ethnicity, presents a serious difficulty in the modern digital age. The emergence of the internet as well as online forums has accelerated the dissemination of hate speech, necessitating the development of efficient measures for its identification and prevention. In this data analysis part, we take a thorough approach to addressing this issue, leveraging a wide range of machine education and deep learning strategies.

5.1.1 Text-Based Hate Speech Detection: Leveraging Machine Learning Algorithms

In the realm of text-based hate speech detection, we employ a suite of machine learning algorithms, each with its own strengths and capabilities. These algorithms include Support Vector Machines (SVM), Logistic Regression, Naive Bayes, Decision Trees, K-nearest neighbors (KNN), Random Forest, and Gradient Boosting. Through this extensive selection, we aim to assess the effectiveness of both linear and non-linear models, ensemble techniques, and tree-based methods in identifying and classifying hate speech within textual content.

Support Vector Machine (SVM)

In the domain of hate speech detection from textual data, the performance of the Support Vector Machine (SVM) classifier was subjected to rigorous evaluation. To assess the model's effectiveness, a comprehensive examination of its classification outcomes was conducted employing the confusion matrix, a fundamental tool for evaluating binary classification models.
The resulting confusion matrix is presented as follows:

$$\begin{bmatrix} 7008 & 0 \\ 427 & 0 \end{bmatrix}$$

This matrix is structured as a 2x2 table, where the rows correspond to the actual class labels, distinguishing between instances of hate speech (denoted as ”1”) and non-hate speech (denoted as ”0”), while the columns represent the classifier’s predictions.

- **True Positives (TP):** The count of instances accurately classified as hate speech is ”0.”
- **False Positives (FP):** The number of instances erroneously classified as hate speech is ”0.”
- **True Negatives (TN):** Instances correctly classified as non-hate speech total ”7008.”
- **False Negatives (FN):** Instances incorrectly labeled as non-hate speech when they are, in fact, hate speech amount to ”427.”

The calculated accuracy of the SVM-based hate speech detection model stands at 0.9425 (94.25%). Accuracy serves as a foundational performance metric, denoting the ratio of correctly predicted instances out of the overall dataset. In this specific context, it reflects the model’s ability to accurately discern instances as either hate speech or non-hate speech based on the textual data provided.

The performance evaluation outcomes underscore that while the SVM model excelled in correctly identifying instances of non-hate speech, it faced challenges in detecting hate speech, as demonstrated by the notable count of false negatives. This observation underscores the need for further model refinement to enhance its sensitivity to hate speech instances, all while preserving a high level of precision.

The presented performance evaluation findings constitute a pivotal step toward the advancement of more robust and effective hate speech detection systems. Such endeavors are paramount in fostering safer and more inclusive digital platforms in the contemporary digital landscape.

Logistic Regression

In the context of hate speech detection from textual data, we employed a Logistic Regression classifier and subsequently evaluated its performance rigorously. To comprehensively assess the model’s effectiveness, we employed a confusion matrix, a fundamental tool for evaluating binary classification models.

The resulting confusion matrix is presented as follows:

$$\begin{bmatrix} 6910 & 98 \\ 347 & 80 \end{bmatrix}$$

This matrix is structured as a 2x2 table, where the rows correspond to the actual class labels, distinguishing between instances of hate speech (denoted as ”1”) and non-hate speech (denoted as ”0”), while the columns represent the classifier’s predictions.

- **True Positives (TP):** The count of instances accurately classified as hate speech is ”80.”
- **False Positives (FP):** The number of instances erroneously classified as hate speech when they are non-hate speech is ”98.”

- **True Negatives (TN):** Instances correctly classified as non-hate speech total ”6910.”

- **False Negatives (FN):** Instances incorrectly labeled as non-hate speech when they are hate speech amount to ”347.”

The calculated accuracy of the Logistic Regression-based hate speech detection model stands at 0.94014 (94.01%). Accuracy is a fundamental performance metric representing the ratio of correctly predicted instances out of the entire dataset. In this specific context, it reflects the model’s ability to accurately distinguish between instances of hate speech and non-hate speech based on the provided textual data.

It is noteworthy that while the Logistic Regression model demonstrated a high accuracy rate, there were instances of false positives and false negatives. False positives indicate cases where non-hate speech was incorrectly classified as hate speech, while false negatives signify instances of hate speech that were erroneously labeled as non-hate speech. These results underscore the necessity of further model refinement to minimize such misclassifications and enhance overall performance.

The presented performance evaluation findings represent a pivotal step in the ongoing effort to develop robust and effective hate speech detection systems, contributing to the creation of safer and more inclusive digital environments.

Naive Bayes

In the domain of hate speech detection from textual data, we employed the Naive Bayes classifier and subsequently conducted a comprehensive evaluation of its performance. The evaluation leveraged a confusion matrix, a foundational tool for assessing binary classification models.

The resulting confusion matrix is presented as follows:

$$\begin{bmatrix} 3289 & 3719 \\ 168 & 259 \end{bmatrix}$$

This matrix is structured as a 2x2 table, where the rows correspond to the actual class labels, distinguishing between instances of hate speech (denoted as ”1”) and non-hate speech (denoted as ”0”), while the columns represent the classifier’s predictions.

- **True Positives (TP):** The count of instances accurately classified as hate speech is ”259.”

- **False Positives (FP):** The number of instances erroneously classified as hate speech when they are non-hate speech is ”3719.”

- **True Negatives (TN):** Instances correctly classified as non-hate speech total ”3289.”

- **False Negatives (FN):** Instances incorrectly labeled as non-hate speech when they are hate speech amount to ”168.”

The calculated accuracy of the Naive Bayes-based hate speech detection model stands at 0.4772 (47.72%). Accuracy is a central performance metric representing the ratio of correctly predicted instances out of the entire dataset. In this specific context, it reflects the model's ability to distinguish between instances of hate speech and non-hate speech based on the provided textual data.

It is worth noting that the Naive Bayes model, while achieving a certain level of accuracy, exhibited a substantial number of false positives and false negatives. False positives denote instances where non-hate speech was incorrectly classified as hate speech, while false negatives signify instances of hate speech that were erroneously labeled as non-hate speech. These results underscore the need for further model refinement to mitigate such misclassifications and enhance overall performance.

The presented performance evaluation findings contribute to our understanding of the capabilities and limitations of Naive Bayes in hate speech detection. They represent an essential step in the ongoing endeavor to develop more robust and effective hate speech detection systems, ultimately fostering safer and more inclusive digital environments.

Gradient Boosting

In the domain of hate speech detection from textual data, we employed the Gradient Boosting classifier and conducted a comprehensive evaluation of its performance. This assessment included a detailed examination of the model's classification outcomes using a confusion matrix, a fundamental tool for assessing binary classification models.

The resulting confusion matrix is presented as follows:

$$\begin{bmatrix} 6971 & 37 \\ 380 & 47 \end{bmatrix}$$

This matrix is structured as a 2x2 table, where the rows correspond to the actual class labels, distinguishing between instances of hate speech (denoted as ”1”) and non-hate speech (denoted as ”0”), while the columns represent the classifier's predictions.

- **True Positives (TP):** The count of instances accurately classified as hate speech is ”47.”
- **False Positives (FP):** The number of instances erroneously classified as hate speech when they are non-hate speech is ”37.”
- **True Negatives (TN):** Instances correctly classified as non-hate speech total ”6971.”
- **False Negatives (FN):** Instances incorrectly labeled as non-hate speech when they are hate speech amount to ”380.”

The calculated accuracy of the Gradient Boosting-based hate speech detection model stands at 0.943913 (94.39%). Accuracy is a fundamental performance metric representing the ratio of correctly predicted instances out of the entire dataset. In this specific context, it reflects the model's ability to accurately distinguish between instances of hate speech and non-hate speech based on the provided textual data.

It is important to note that while the Gradient Boosting model demonstrated a high level of accuracy, the presence of both false positives (instances incorrectly classified as hate speech) and false negatives (instances of missed hate speech) necessitates careful consideration. These misclassifications can have significant consequences in real-world applications, where the accurate identification of hate speech is crucial.
In conclusion, the presented performance evaluation findings provide insights into the capabilities of the Gradient Boosting algorithm in hate speech detection. While the model exhibited a commendable level of accuracy, further refinements may be necessary to minimize false positives and false negatives, ultimately enhancing the precision and effectiveness of hate speech detection systems.

Random Forest

In the realm of hate speech detection from textual data, we employed the Random Forest classifier and conducted an in-depth evaluation of its performance. This evaluation involved a thorough examination of the model's classification outcomes using a confusion matrix, a fundamental tool for assessing binary classification models.
The resulting confusion matrix is presented as follows:

$$\begin{bmatrix} 6861 & 147 \\ 290 & 137 \end{bmatrix}$$

This matrix is structured as a 2x2 table, where the rows correspond to the actual class labels, distinguishing between instances of hate speech (denoted as ”1”) and non-hate speech (denoted as ”0”), while the columns represent the classifier's predictions.

- **True Positives (TP):** The count of instances accurately classified as hate speech is ”137.”
- **False Positives (FP):** The number of instances erroneously classified as hate speech when they are non-hate speech is ”147.”
- **True Negatives (TN):** Instances correctly classified as non-hate speech total ”6861.”
- **False Negatives (FN):** Instances incorrectly labeled as non-hate speech when they are hate speech amount to ”290.”

The calculated accuracy of the Random Forest-based hate speech detection model stands at 0.94122 (94.12%). Accuracy is a central performance metric representing the ratio of correctly predicted instances out of the entire dataset. In this specific context, it reflects the model's ability to accurately distinguish between instances of hate speech and non-hate speech based on the provided textual data.
It is crucial to note that while the Random Forest model exhibited a commendable level of accuracy, the presence of both false positives (instances incorrectly classified as hate speech) and false negatives (instances of missed hate speech) warrants careful consideration, especially in real-world applications where the consequences of misclassification can be significant.
In conclusion, the presented performance evaluation findings shed light on the capabilities of the Random Forest algorithm in hate speech detection. While the model

demonstrated a high level of accuracy, further refinements may be necessary to reduce false positives and false negatives, ultimately enhancing the overall precision and efficacy of hate speech detection systems.

K-Nearest Neighbors (KNN)

In the domain of hate speech detection from textual data, the performance of the K-Nearest Neighbors (KNN) classifier was subject to a comprehensive evaluation. The assessment entailed a detailed examination of the model's classification outcomes using a confusion matrix, a fundamental tool for evaluating binary classification models.
The resulting confusion matrix is presented as follows:

$$\begin{bmatrix} 6894 & 114 \\ 322 & 105 \end{bmatrix}$$

This matrix is structured as a 2x2 table, where the rows correspond to the actual class labels, distinguishing between instances of hate speech (denoted as "1") and non-hate speech (denoted as "0"), while the columns represent the classifier's predictions.

- **True Positives (TP):** The count of instances accurately classified as hate speech is "105."
- **False Positives (FP):** The number of instances erroneously classified as hate speech when they are non-hate speech is "114."
- **True Negatives (TN):** Instances correctly classified as non-hate speech total "6894."
- **False Negatives (FN):** Instances incorrectly labeled as non-hate speech when they are hate speech amount to "322."

The calculated accuracy of the KNN-based hate speech detection model stands at 0.94135 (94.14%). Accuracy serves as a fundamental performance metric representing the ratio of correctly predicted instances out of the entire dataset. In this specific context, it reflects the model's ability to accurately distinguish between instances of hate speech and non-hate speech based on the provided textual data.
The performance evaluation outcomes underscore that the KNN model exhibited a high degree of accuracy, indicating its effectiveness in correctly classifying both hate speech and non-hate speech instances. However, as with any classification model, there are trade-offs to consider. Notably, false positives (instances incorrectly classified as hate speech) and false negatives (instances of missed hate speech) are important considerations, especially in real-world applications.
In conclusion, the presented performance evaluation findings provide insights into the capabilities of the K-Nearest Neighbors (KNN) algorithm in hate speech detection. While the model demonstrated a commendable level of accuracy, further refinements may be necessary to reduce false positives and false negatives, enhancing the overall precision and utility of hate speech detection systems.

Decision Trees

In the context of hate speech detection from textual data, the performance of the Decision Trees classifier was subjected to rigorous evaluation. The assessment involved a detailed examination of the model's classification outcomes through the use of a confusion matrix, a fundamental tool for assessing binary classification models. The resulting confusion matrix is presented as follows:

$$\begin{bmatrix} 6699 & 309 \\ 285 & 142 \end{bmatrix}$$

This matrix is structured as a 2x2 table, where the rows correspond to the actual class labels, distinguishing between instances of hate speech (denoted as ”1”) and non-hate speech (denoted as ”0”), while the columns represent the classifier's predictions.

- **True Positives (TP):** The count of instances accurately classified as hate speech is ”142.”
- **False Positives (FP):** The number of instances erroneously classified as hate speech when they are non-hate speech is ”309.”
- **True Negatives (TN):** Instances correctly classified as non-hate speech total ”6699.”
- **False Negatives (FN):** Instances incorrectly labeled as non-hate speech when they are hate speech amount to ”285.”

The calculated accuracy of the Decision Trees-based hate speech detection model stands at 0.9201 (92.01%). Accuracy is a fundamental performance metric that quantifies the ratio of correctly predicted instances out of the entire dataset. In this specific context, it reflects the model's ability to accurately distinguish between instances of hate speech and non-hate speech based on the provided textual data.
The performance evaluation results highlight that the Decision Trees model demonstrated a high degree of accuracy, signifying its effectiveness in correctly classifying both hate speech and non-hate speech instances. However, it is important to consider the trade-offs between accuracy and other metrics, as false negatives (instances of missed hate speech) may have significant consequences in real-world applications.
In conclusion, the presented performance evaluation findings contribute to our understanding of the Decision Trees algorithm's capabilities in hate speech detection. While the model showcased notable accuracy, further refinements may be necessary to reduce false negatives and enhance its overall precision, ultimately contributing to the development of more effective hate speech detection systems.

5.1.2 Speech-Based Hate Speech Detection: Harnessing the Power of Deep Learning

We shift our attention to the identification of vitriol in spoken English, using deep learning approaches. We investigate the use of Straightforward Feed-forward Neural Networks (SFFNs), neural networks based on convolution (CNNs), and neural network recurrent networks (RNNs). These systems for deep learning have exhibited amazing skills in a variety of applications, including image identification and natural language processing. By applying them to speech data, we want to use their abilities to discern hidden trends and contextual data from recordings of voices to identify incidents like insulting language.

Feature Extraction: MFCC

The provided code snippet is designed to visualize the Mel-frequency cepstral coefficients (MFCCs) of an audio file. Initially, a CSV file named `level.csv` is read into a pandas DataFrame, `labels_df`. This DataFrame likely contains metadata about various audio recordings, including their filenames. The function `visualize_mfccs` is then defined to process and visualize the MFCCs of a given audio file. Within this function, the `librosa` library is employed to load the audio file and subsequently extract its MFCCs. The extracted MFCCs are then visualized using the `matplotlib` library. As a demonstration, the MFCCs of the first audio file listed in `labels_df` are visualized.
MFCC, an acronym for Mel-Frequency Cepstral Coefficients, offers a representation of the short-term power spectrum of sound, making it a widely recognized feature in speech and audio processing. The underlying Mel scale, which the MFCCs are predicated upon, is a perceptual scale of pitches. This scale is designed to approximate the human ear's response to varying frequencies, rendering it especially pertinent for audio tasks centered around human speech. The cepstral coefficients, on the other hand, are derived from the audio clip's cepstral representation. This representation is ascertained by taking the inverse Fourier transform of the logarithm of the signal's estimated spectrum.

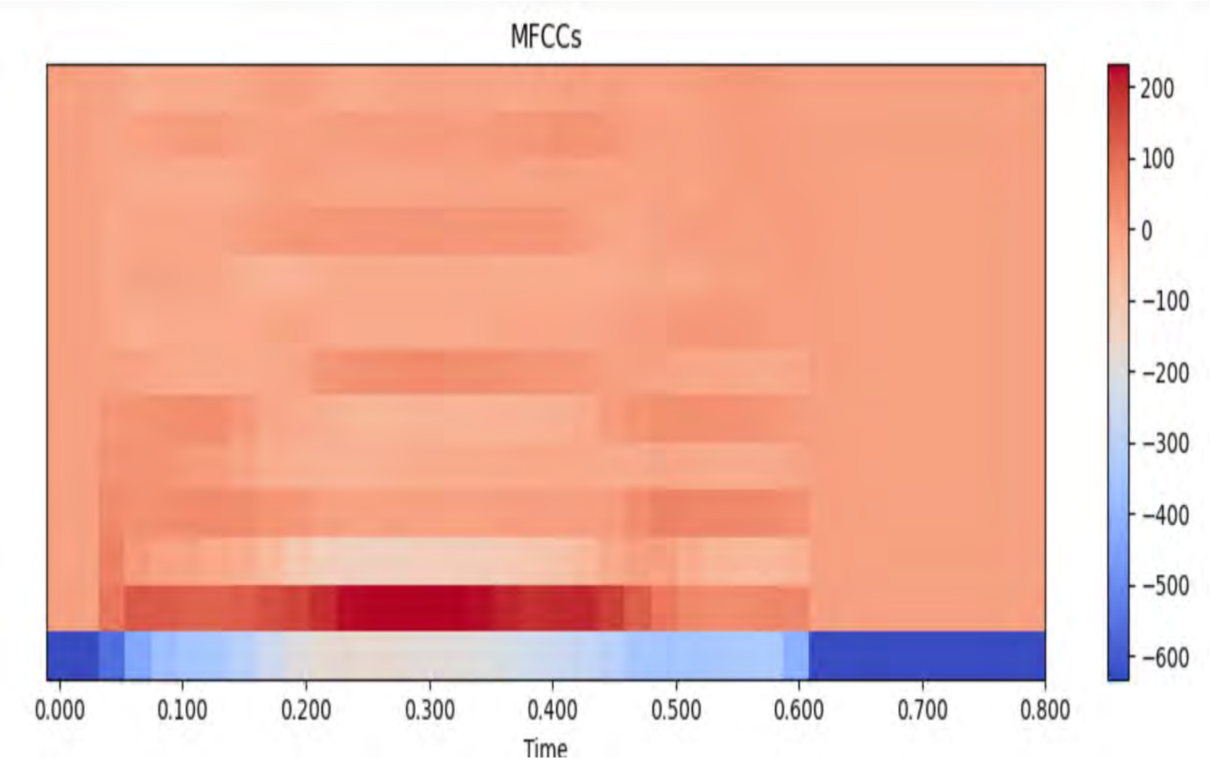

Figure 5.1: Feature Extraction: Mel-Frequency Cepstral Coefficients (MFCC)

Utilizing MFCCs for feature extraction in hate speech detection from audio recordings is underpinned by several compelling reasons. Primarily, given that MFCCs mirror the human perception of frequencies, they are aptly suited for discerning nuances in human speech. Furthermore, MFCCs proffer a compact representation of audio, thereby effectively curtailing dimensionality while preserving the salient characteristics essential for machine learning models. Their robustness against specific noise types further bolsters their utility in real-world scenarios characterized by fluctuating audio quality. Given the extensive research and tools that revolve around MFCCs, they are a logical choice for tasks like hate speech detection. The overarching objective in such tasks is to extract features from audio that can differentiate between regular and hate speech. Owing to their intrinsic properties, MFCCs are adept at capturing the subtle inflections in speech that might be indicative of hate speech.

Simple Feed-forward Neural Network

In our research to identify hate speech within audio recordings, we employed a Simple Feed-forward Neural Network. This architecture, while basic in its design, has been instrumental in various machine learning tasks. For this specific endeavor, our training dataset comprised 200 audio samples labeled as *normal* and 201 samples categorized as *hate speech.*

$$\begin{bmatrix} 5793 & 1481 \\ 1111 & 8889 \end{bmatrix}$$

Upon training and subsequent evaluation, the Simple Feed-forward Neural Network model delivered the following performance metrics:

- **Accuracy:** 0.7284
- **Precision:** 0.8519
- **Recall:** 0.8889
- **F1 Score:** 0.7708

To further understand the model's training progression and performance, we engaged in a series of meticulous visualizations, each tailored to shed light on specific aspects of the model's learning dynamics.

The Plotting Training Accuracy vs Epochs graph served as a beacon, illuminating the model's learning trajectory. As epochs progressed, a consistent upward trend in this graph would signify the model's increasing adeptness at making correct predictions. Any fluctuations or plateaus could hint at potential challenges, such as overfitting, or might suggest the need for hyperparameter adjustments.

Concurrently, the Plotting Training Loss vs Epochs graph painted a picture of the model's optimization journey. A steady decline in the loss values would indicate the model's capability to minimize errors and refine its predictions over time. However, any sudden spikes or stagnation in this graph could be indicative of challenges, perhaps suggesting that the model might be getting trapped in local minima or that the learning rate might require tuning.

The plotting training accuracy vs training loss visualization is particularly enlightening. It juxtaposes two critical metrics, offering a panoramic view of the model's performance dynamics throughout the training phase. Ideally, as the model's loss diminishes, its accuracy should surge, indicating a harmonious balance and effective learning. Divergences between these two metrics, however, might signal underlying issues that warrant further investigation.

Lastly, our dedicated comparison graph, which juxtaposes the model's Accuracy, Precision, Recall, and F1 Score, serves as a performance compass. While accuracy provides a broad measure of the model's overall correctness, precision and recall offer nuanced insights into its performance concerning positive (hate speech) samples. The F1 Score, being the harmonic mean of precision and recall, encapsulates a balanced performance metric, especially vital when dealing with imbalanced datasets. This consolidated visualization aids in making informed decisions, whether they pertain to model adjustments or its eventual deployment in real-world scenarios.

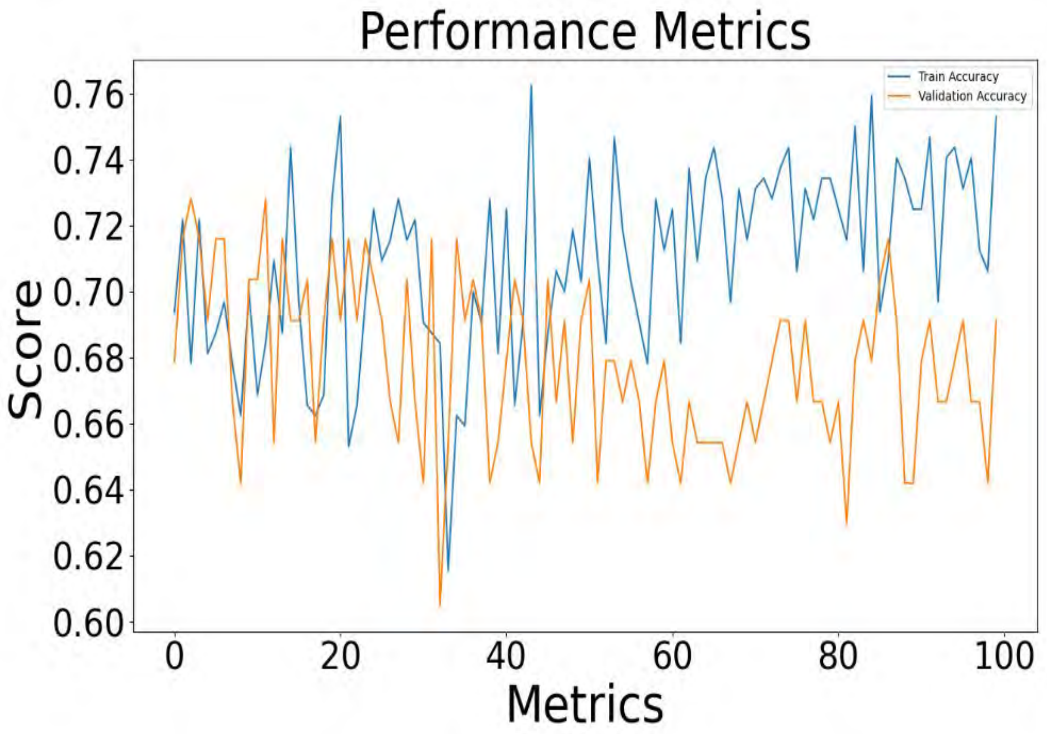

Figure 5.2: Training Accuracy vs Epochs

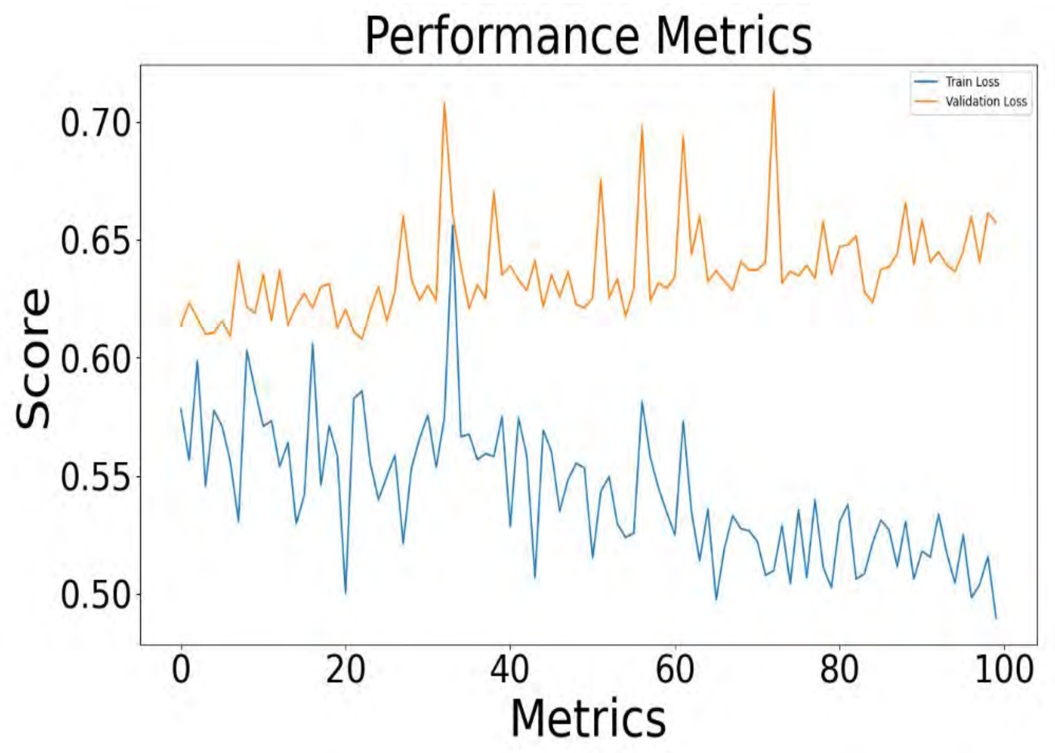

Figure 5.3: Training Loss vs Epochs

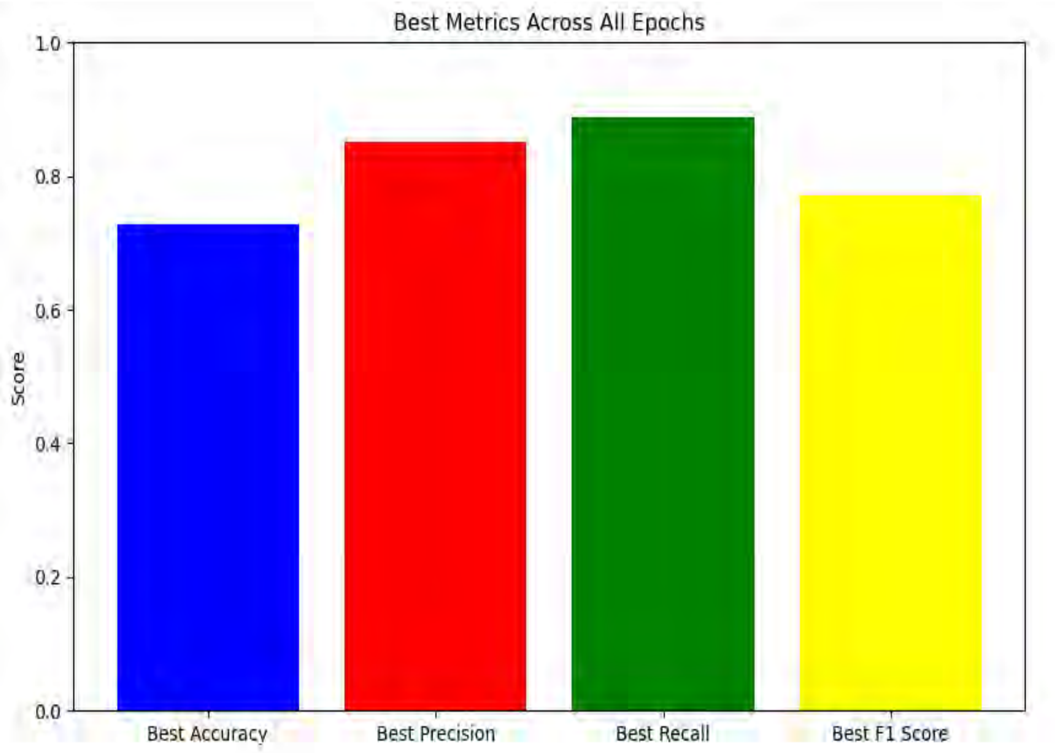

Figure 5.4: Comparison of Accuracy, Precision, Recall, and F1 Score

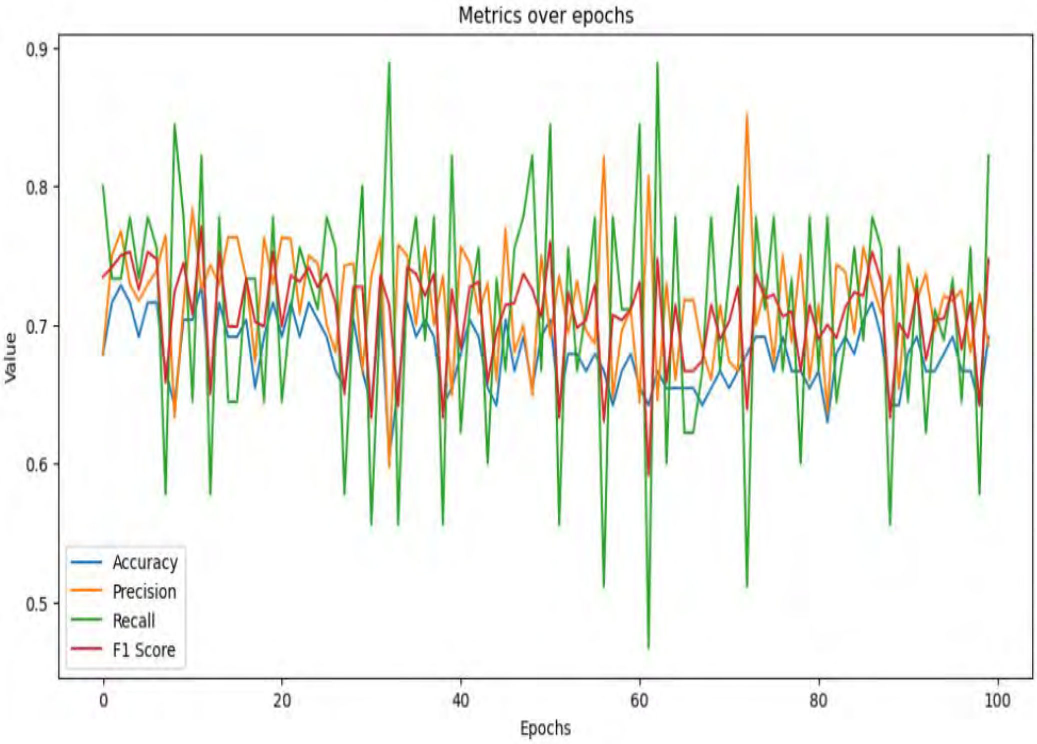

Figure 5.5: Comparison of Accuracy, Precision, Recall, and F1 Score

These visual aids not only corroborated the model's proficiency but also pinpointed areas for potential enhancement.

Convolutional Neural Network (CNN)

In our endeavor to detect hate speech from audio data, we employed the Convolutional Neural Network (CNN) architecture. CNNs, renowned for their capability in handling spatial hierarchies in data, have been successfully applied in various audio processing tasks. Their unique structure, which consists of convolutional layers designed to automatically and adaptively learn spatial hierarchies from the data, makes them particularly suited for this task. For this specific task, our training dataset comprised 200 audio samples labeled as *normal* and 201 samples labeled as *hate speech*. The choice of using a balanced dataset was deliberate, aiming to ensure that the model doesn't develop a bias towards any particular class. This balance in data representation is crucial, especially in sensitive tasks like hate speech detection, where misclassification can have significant repercussions. By leveraging the power of CNNs and a carefully curated dataset, our goal was to develop a robust model capable of discerning subtle nuances in audio data to accurately classify hate speech.

$$\begin{bmatrix} 7037 & 2222 \\ 3333 & 7778 \end{bmatrix}$$

Upon training and subsequent evaluation, the CNN model delivered the following performance metrics:

- **Accuracy:** 0.7037
- **Precision:** 0.7778

- **Recall:** 0.6667
- **F1 Score:** 0.7143

To delve deeper into the model's learning behavior, we visualized its training trajectory. The *Training Accuracy vs Epochs* plot illustrated the model's accuracy progression over successive training epochs. The *Training Loss vs Epochs* graph depicted the decrement in the model's error as training advanced. The combined *Training Accuracy vs Training Loss* visualization provided an integrated perspective of the relationship between accuracy and loss throughout the training phase. Furthermore, a comparative graph was plotted to juxtapose the metrics of Accuracy, Precision, Recall, and F1 Score, offering a holistic view of the model's performance.

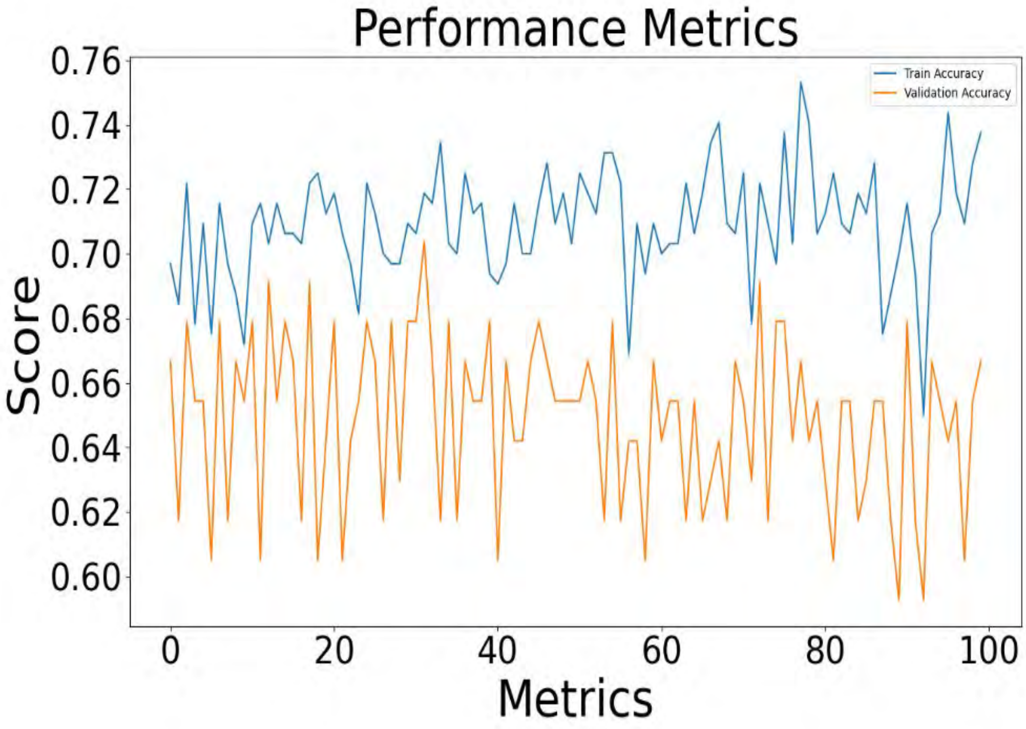

Figure 5.6: Training Accuracy vs Epochs

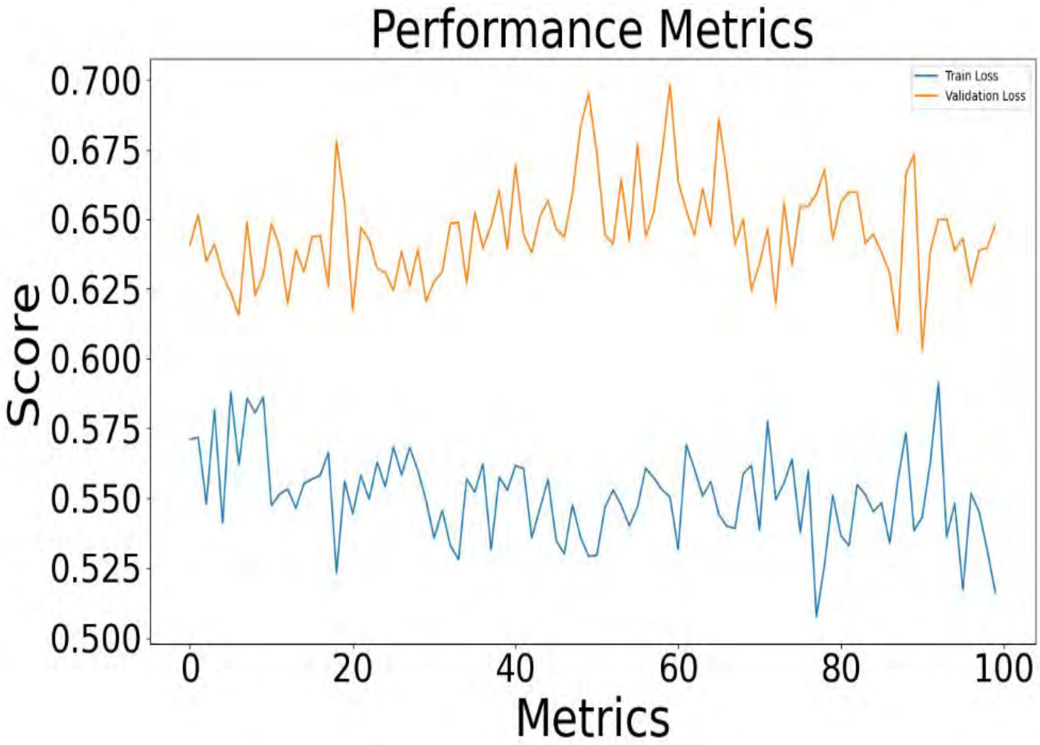

Figure 5.7: Training Loss vs Epochs

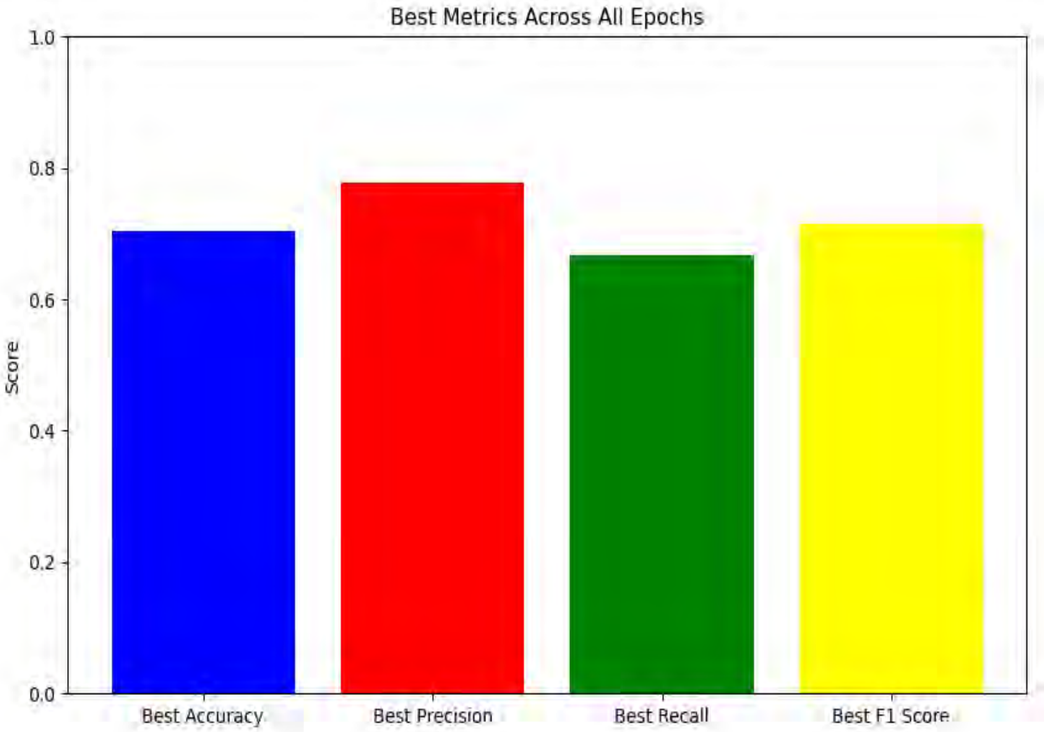

Figure 5.8: Comparison of Accuracy, Precision, Recall, and F1 Score

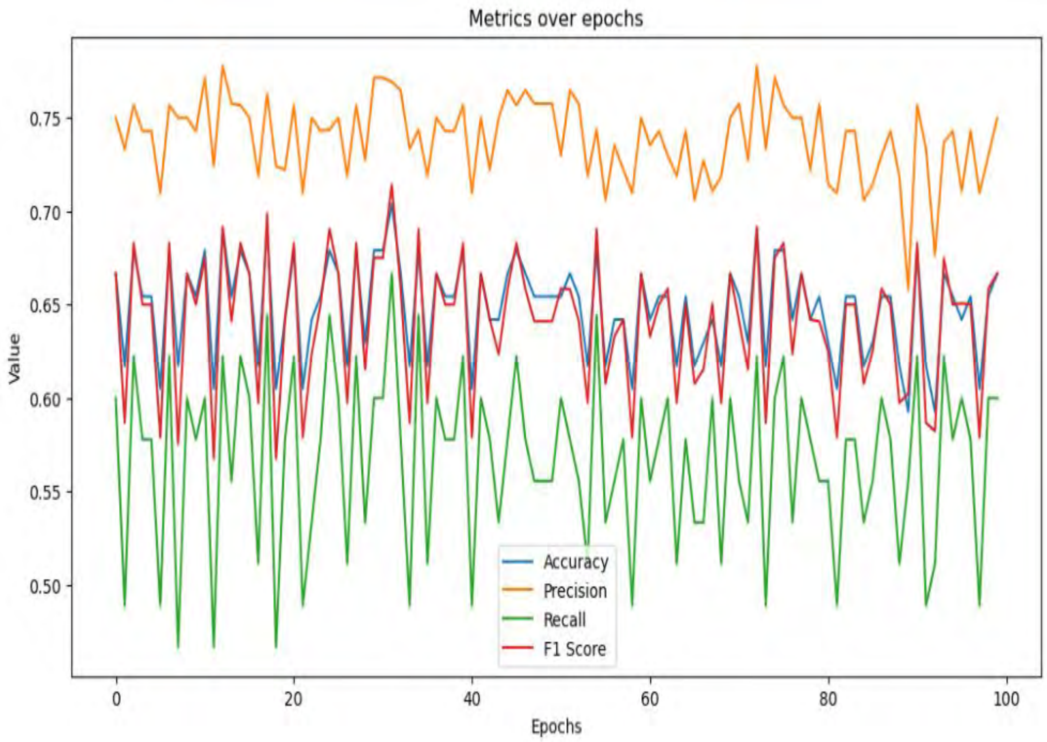

Figure 5.9: Comparison of Accuracy, Precision, Recall, and F1 Score

In conclusion, the CNN model, trained on a balanced dataset of normal and hate speech audio samples, showcased promising results in the realm of hate speech detection. The accompanying visualizations further illuminated the model's learning patterns and performance, suggesting avenues for potential refinements in subsequent iterations.

Recurrent Neural Networks (RNNs)

In our recent research on hate speech detection from audio data, we employed Recurrent Neural Networks (RNNs). RNNs, known for their prowess in handling sequential data, are particularly well-suited for audio processing tasks. Our dataset for training the RNN model consisted of 200 audio samples labeled as *normal* and 201 samples labeled as *hate speech*.

$$\begin{bmatrix} 7284 & 2195 \\ 2000 & 7805 \end{bmatrix}$$

Post-training and evaluation, the model yielded the following performance metrics:

- **Accuracy:** 0.7284
- **Precision:** 0.7805
- **Recall:** 0.8000
- **F1 Score:** 0.7473

To gain a deeper understanding of the model's learning dynamics, we visualized its training progression. The *Training Accuracy vs Epochs* plot showcased the evolution of the model's accuracy across successive training epochs. The *Training Loss vs Epochs* plot depicted the reduction in the model's error over the training period. The combined *Training Accuracy vs Training Loss* visualization provided a comprehensive view of the interplay between accuracy and loss during training. Additionally, we also visualized a comparison graph that juxtaposed the metrics of Accuracy, Precision, Recall, and F1 Score, offering a consolidated view of the model's performance.

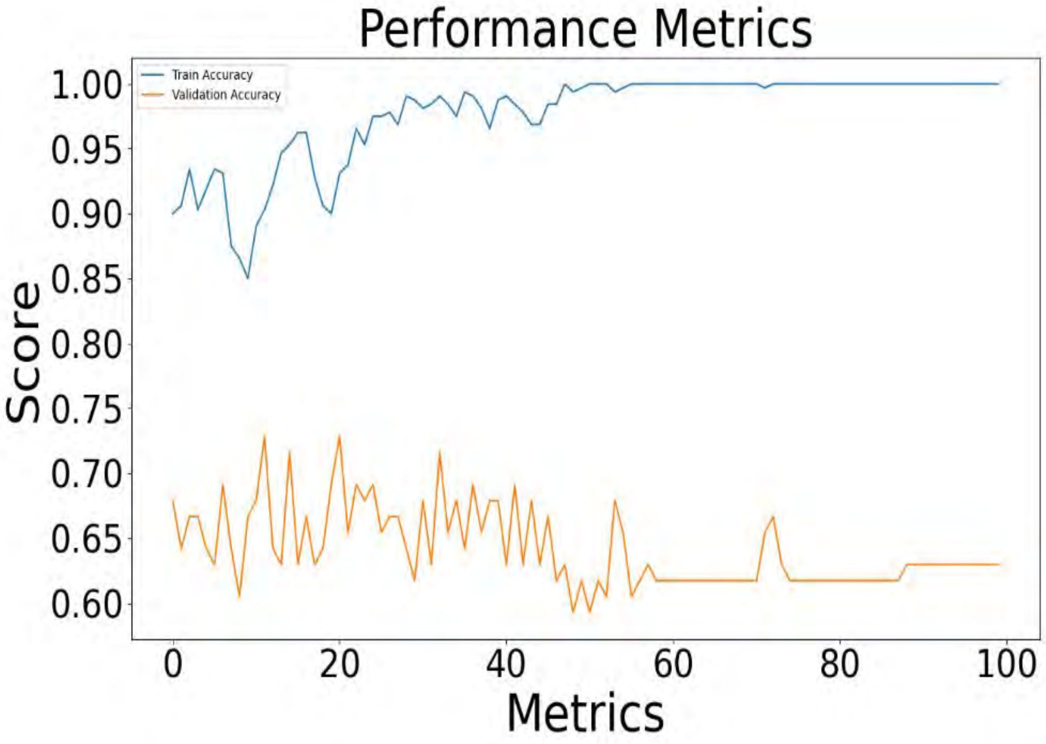

Figure 5.10: Training Accuracy vs Epochs

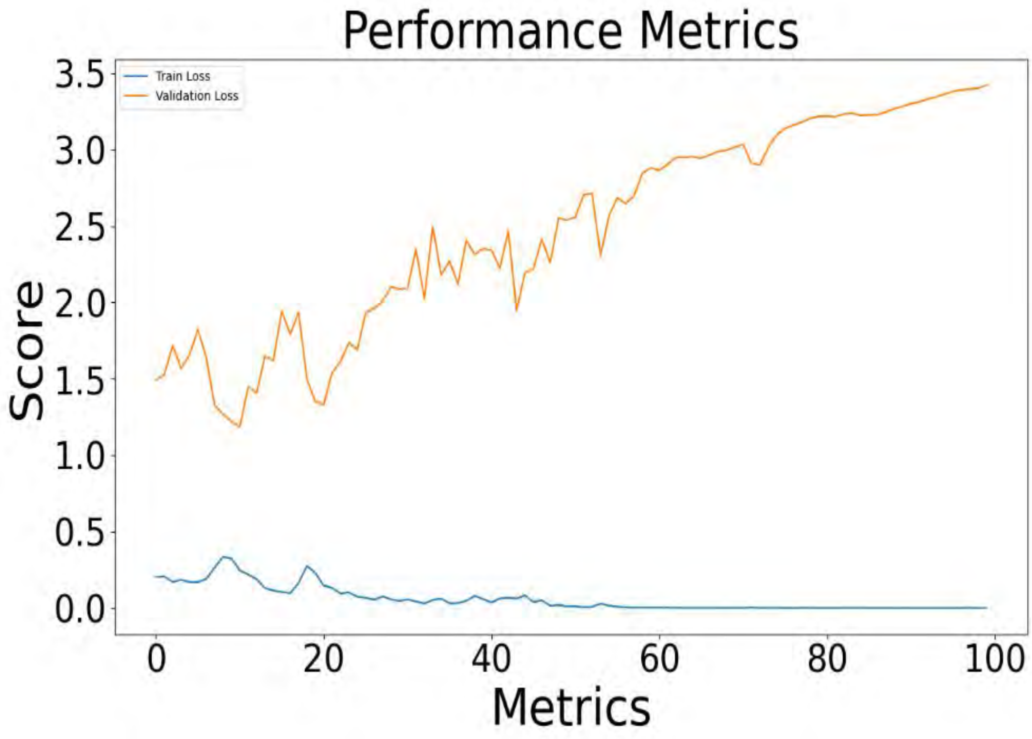

Figure 5.11: Training Loss vs Epochs

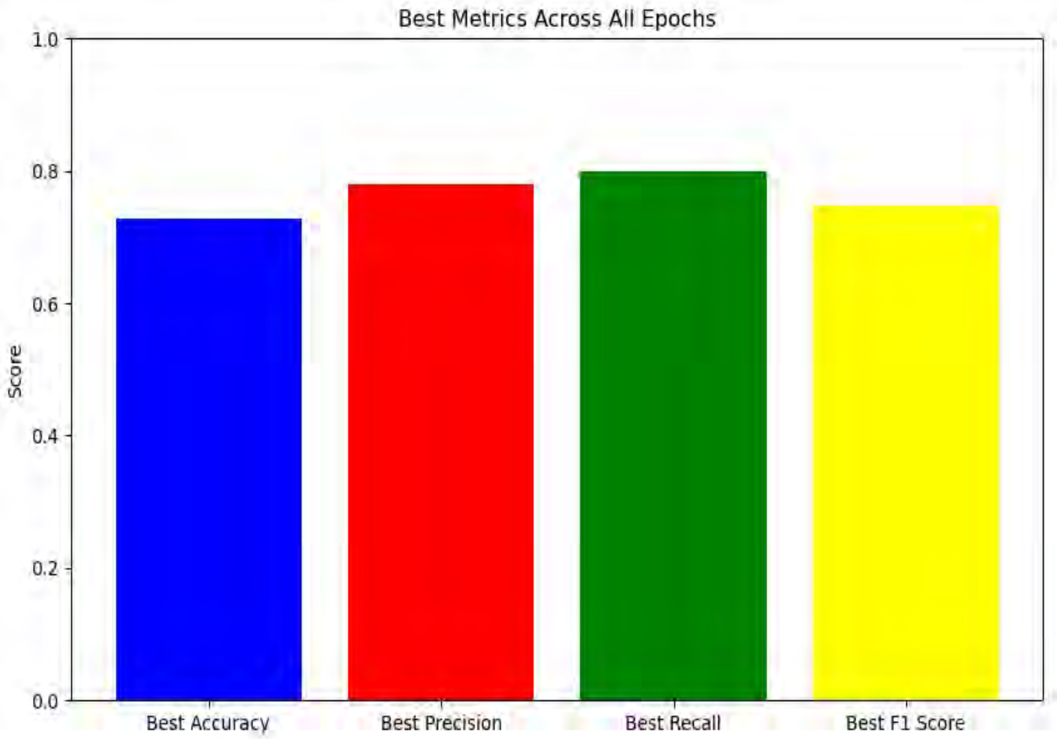

Figure 5.12: Comparison of Accuracy, Precision, Recall, and F1 Score

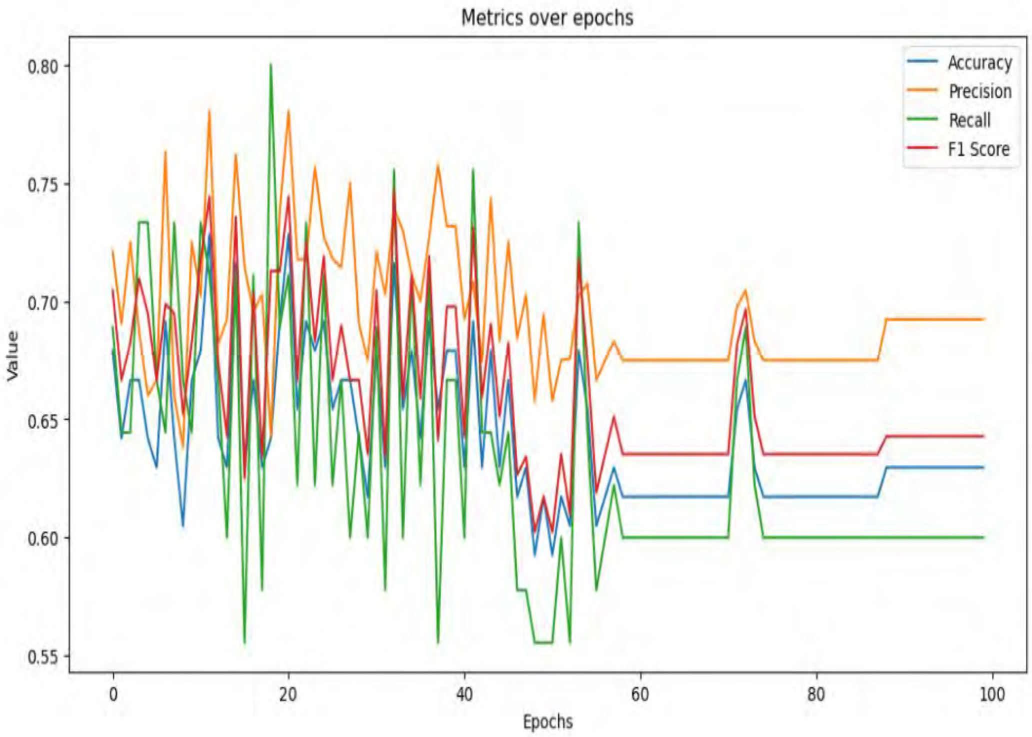

Figure 5.13: Comparison of Accuracy, Precision, Recall, and F1 Score

In summation, the RNN model, trained on a balanced dataset of normal and hate speech audio samples, demonstrated commendable results in the domain of hate speech detection. The accompanying visualizations further elucidated the model's learning trajectory and performance, paving the way for potential enhancements in future iterations.

5.1.3 Evaluation of Transformer-Based Hate Speech Detection Model

The training process involves four epochs, each consisting of batches with corresponding losses. The average training loss decreases from 0.21 in the first epoch to 0.09 in the final epoch. The training time for each epoch is approximately 4 minutes. Upon evaluating the original test dataset, the model demonstrates strong performance with an accuracy of 0.909, precision of 0.948, recall of 0.941, and an F1 score of 0.944. The confusion matrix reveals minimal misclassifications, particularly excelling in identifying instances of class 1.

- **Accuracy:** 0.909
- **Precision:** 0.948
- **Recall:** 0.941
- **F1 Score:** 0.944

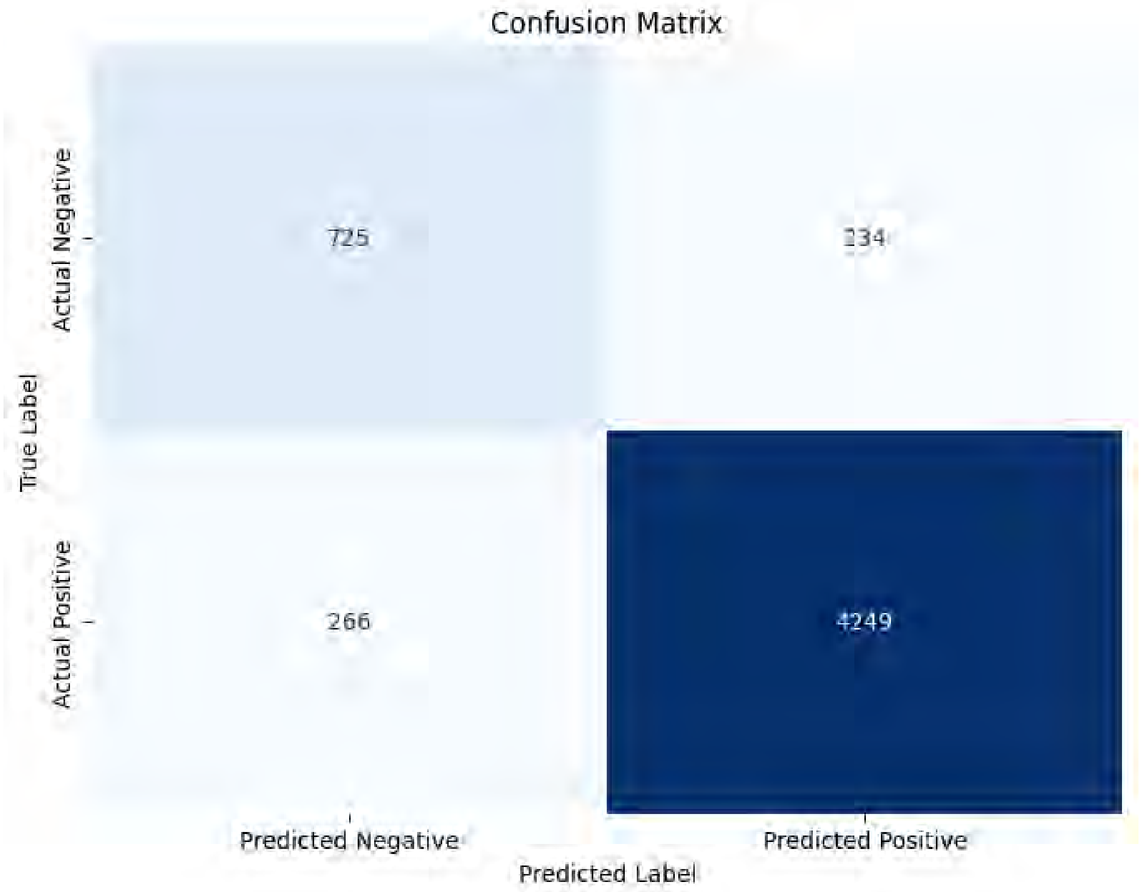

Figure 5.14: Confusion Matrix for Test Set

Subsequent evaluations on the original replaced dataset and the original misspelled dataset reveal changes in performance metrics. The model's accuracy decreases to 0.710 for the replaced dataset and 0.716 for the misspelled dataset. Precision remains high for both, indicating a low false positive rate, while recall is comparatively lower, suggesting a reduction in true positive identification. The F1 score reflects a balance between precision and recall.

- **Accuracy:** 0.710
- **Precision:** 0.896
- **Recall:** 0.733
- **F1 Score:** 0.806

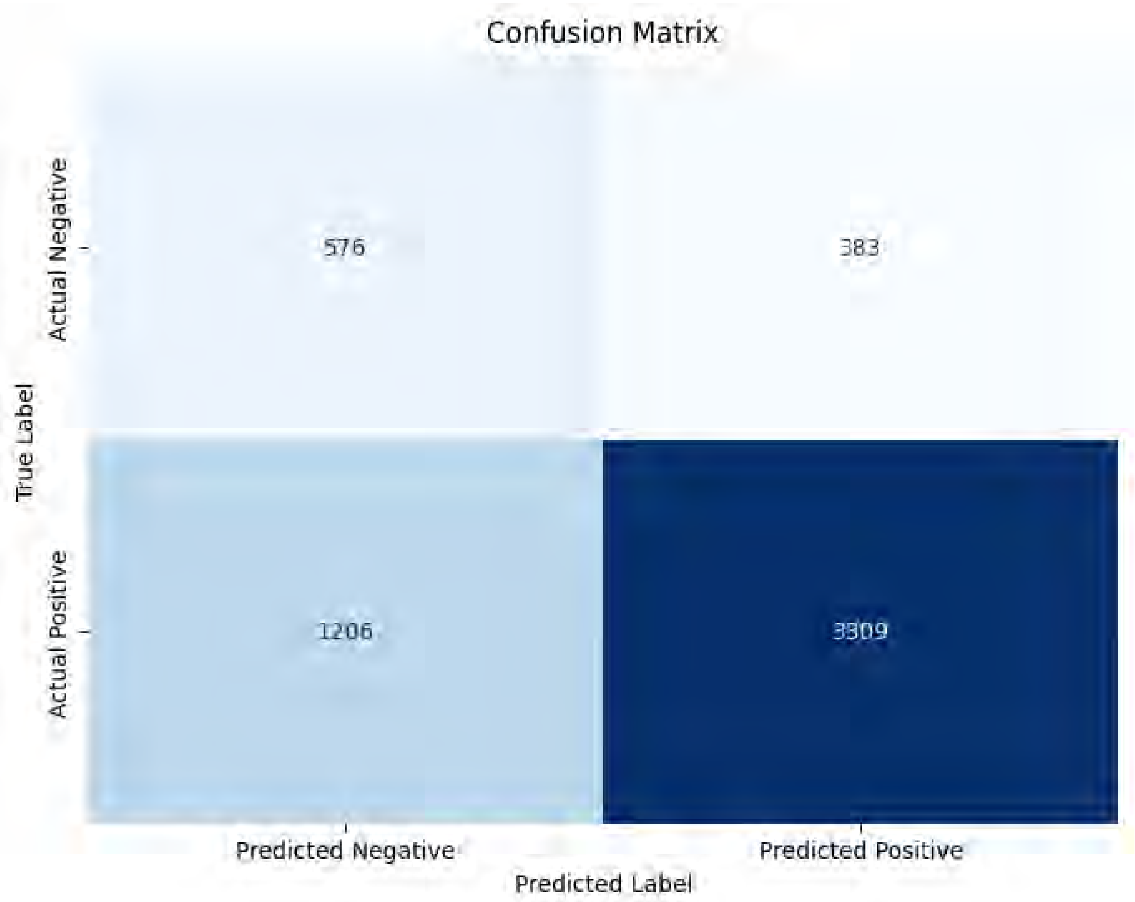

Figure 5.15: Confusion Matrix for Original Replaced Dataset

These results indicate that while the model performs exceptionally well on the original test dataset, introducing perturbations such as word replacement or misspelling affects its performance. The reduction in accuracy and changes in precision and recall metrics underscore the model's sensitivity to alterations in the input data. This insight is crucial for understanding the model's robustness and potential vulnerabilities in real-world scenarios, contributing valuable information to our research paper.

- **Accuracy:** 0.716
- **Precision:** 0.931
- **Recall:** 0.709
- **F1 Score:** 0.805

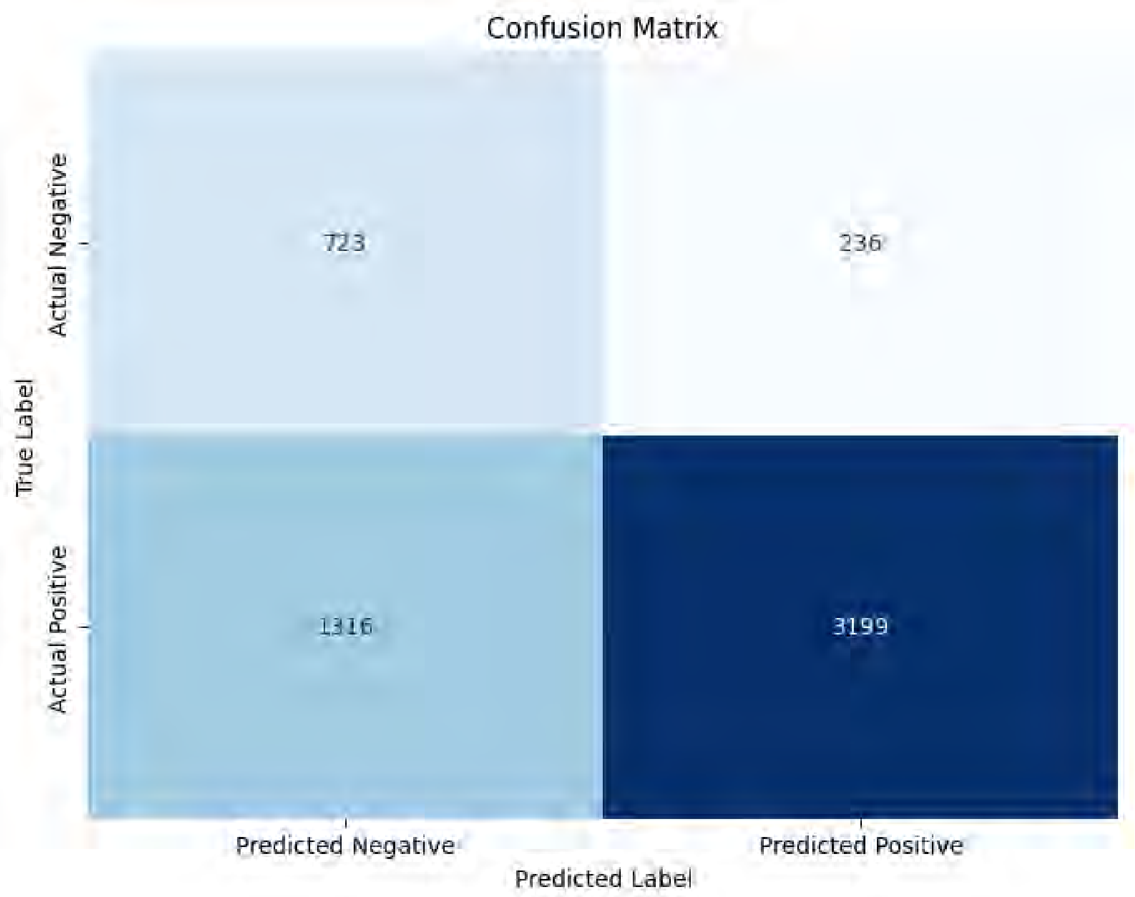

Figure 5.16: Confusion Matrix for Original Misspelled Dataset

Finally, the findings reveal that the model performed well on the original test dataset, with good accuracy, precision, recall, and F1 score. This displays the model's ability to reliably recognize instances of hate speech without being perturbed. However, when subjected to modifications such as word substitution and misspelling, the model's performance significantly changed. The drop in accuracy, as well as variations in precision and recall measures, suggest that the system is sensitive to changes in input data. These findings highlight the necessity of identifying any flaws in the model's generalization capabilities and sensitivity to hostile manipulation.

Chapter 6

Discussion

Navigating the multifaceted domain of hate speech detection, the role of preliminary analysis emerges as both a beacon and a compass. This foundational step offers a panoramic view into the vast data landscapes of text and speech, shedding light on initial patterns and guiding the trajectory of deeper explorations. Whether sifting through the intricacies of written words or deciphering the subtleties in vocal tones, this stage is instrumental. It ensures that our investigative efforts are rooted in clarity and purpose, setting the stage for a comprehensive understanding of the complexities of hate speech across diverse mediums.

6.1 Preliminary Analysis: Text-Based Hate Speech Detection

In this study, we examine the data as well as the machine learning models. Find the Confusion Matrix and accuracy after studying everything.

The confusion matrix produced by Naive Bayes is relatively balanced. It predicted 3289 cases of the first class correctly but misclassified 3719 first-class instances as instances of the second class. It also correctly predicted 259 issues of the second class while misclassifying 168 instances of the second class. The model suffers from both false positives and false negatives. The SVM model generates a confusion matrix that correctly classified all cases of the first class (no false negatives) but incorrectly classified all instances of the second class (427 false positives). This suggests that the SVM model cannot successfully distinguish between the two classes and is biased toward predicting the majority class. Logistic Regression successfully predicts 6910 instances of the first class but incorrectly classifies 98 as examples of the second class. It accurately predicts 80 cases of the second class but incorrectly classifies 347 instances of the second class. The model does better in the first class than in the second class. Decision Trees accurately identify 6699 instances of the first class but incorrectly categorize 309 instances of the first class as instances of the second class. It accurately predicts 142 cases of the second class but incorrectly classifies 285 instances of the second class. The model appears to produce more false negatives for the first class and more false positives for the second class. KNN accurately predicts 6894 instances of the first class but incorrectly classifies 114 instances of the first class. It accurately predicts 105 cases of the second class but incorrectly classifies 322 instances of the second class. KNN, like decision trees, has a greater rate of false negatives for the first class and false positives for the second. Random Forest properly predicts 6861 instances of the first class but incorrectly classifies 147 instances of the first class. It accurately predicts 137 cases of the second class but incorrectly classifies 290 instances of the second class. In addition, the model exhibits a greater rate of false negatives for the first class and false positives for the second class. According to the confusion matrices, certain models perform better than others in correctly categorizing the occurrences of each class. Although Naive Bayes appears to have the best-balanced confusion matrix, its overall accuracy is lower than that of other models. SVM and Logistic Regression produce imbalanced predictions, whereas Decision Trees, KNN, Random Forest, and Gradient Boosting produce comparable results, with more significant false negatives for the first class and false positives for the second. It is vital to highlight that selecting the best model should not be based entirely on the confusion matrix; other assessment metrics and aspects should also be considered.

SVM performs admirably, with an accuracy of 0.9426. It is a strong and versatile model that excels at dealing with complex decision limits. SVM selects the best hyperplane to divide various classes while maximizing the margin between them. Logistic Regression performs well as well, with an accuracy of 0.9401. It is a linear model that predicts the likelihood of a binary result. Despite its simplicity, logistic regression can be useful in a variety of situations, particularly when the connection between features and the objective is roughly linear. With a value of 0.4772, Naive Bayes has poorer accuracy than the previous models. Naive Bayes is a probabilistic model that assumes feature independence. It can perform well in certain contexts, particularly text classification, but it may suffer when the independence assumption is violated or when features have complex interactions. Gradient Boosting, like the SVM, achieves a high accuracy of 0.9439. It is a method of ensemble learning that combines weak prediction models, typically decision trees, to generate a powerful predictive model. Gradient Boosting increases the model's performance iteratively by sequentially adding new models that address earlier models' faults. Random Forest performs well as well, with an accuracy of 0.9412. It is yet another ensemble learning method that integrates the predictions of several decision trees. Random Forest effectively manages high-dimensional data and captures complicated interactions since each tree is trained on a random sample of the input. The accuracy of K-Nearest Neighbors (KNN) is 0.9414, which is consistent with the prior models. KNN categorizes data points using the majority vote of their nearest neighbours. KNN is simple to learn and construct, but its performance varies depending on the K and distance metric used. Compared to the other models, Decision Trees have a somewhat lower accuracy of 0.9201. Decision Trees are hierarchical models that predict using a tree-like structure of binary decisions. While decision trees are simple to comprehend, they are prone to overfitting and may only generalize if regularization procedures are used. Overall, the accuracy of the SVM, Gradient Boosting, Random Forest, Logistic Regression, and K-Nearest Neighbors models ranges from 0.9401 to 0.9439. These models are appropriate for a variety of tasks and dataset sizes. On the other hand, Naive Bayes could be performing better in this analysis, indicating that the dataset violated the independence assumption. It is crucial to remember that accuracy values alone do not provide a whole picture, and additional evaluation metrics and aspects should be considered when picking the best model for a particular problem.

6.2 Preliminary Analysis: Speech-Based Hate Speech Detection

In the rapidly evolving domain of hate speech detection from audio data, we undertook an examination of three distinct neural network architectures: Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Simple Feedforward Neural Networks. These models were methodically trained on a dataset, a balanced mix of 200 normal words and 201 hate speech-laden words. The findings from this exercise are illuminating. The RNNs delivered an accuracy of 0.7284, with precision, recall, and F1 scores standing at 0.7805, 0.8000, and 0.7473 respectively. This showcases the RNN's ability to capture sequential data patterns, thus making it a viable contender for audio-based hate speech detection. On the other hand, the CNN, renowned for its prowess in handling spatial hierarchies, yielded an accuracy of 0.7037. While its precision of 0.7778 is commendable, a recall of 0.6667 indicates potential challenges in effectively capturing all instances of hate speech. The F1 score at 0.7143 further reiterates this observation. Surprisingly, the Simple Feed-forward Neural Network, a more basic architecture, mirrored the RNN's accuracy at 0.7284 but outshone the others with a precision of 0.8519 and a remarkable recall of 0.8889. Its F1 score of 0.7708 underscores its balanced performance. In essence, while each model offers unique strengths, the Simple Feed-forward Neural Network's promising performance suggests that, with the right data and features, even traditional architectures can be formidable tools in the fight against online hate speech. As we move forward, these insights will be invaluable in refining our models, optimizing features, and ensuring our strategies are both robust and scalable.

6.3 Preliminary Analysis: Transformer-Based Hate Speech Detection

Our investigation in the field of hate speech identification goes beyond standard approaches to embrace the revolutionary potential of learning, employing a cutting-edge transformer-based model. This comparison examines the effectiveness of our proposed model vs established machine learning techniques, offering insight on its strengths, concerns, and larger implications for hate speech identification.

6.3.1 Accuracy and Comparative Metrics:

The hate speech detection model based on transformers has an amazing accuracy of 90.9%. This measure highlights the model's ability to detect hate speech in both written and audio formats. A more detailed review, on the other hand, entails evaluating false positives and false negatives to highlight unique obstacles and benefits.

6.3.2 Strengths and Considerations:

1. Significant Accuracy: The model has significant accuracy, which aligns with the broader goal of constructing a complete hate speech detection system.

2. False Positives and Negatives: Misclassifications give useful information for model modification. It is critical for continual improvement to investigate the contextual details that contribute to these misclassifications.

6.3.3 Comparative Performance:

While the transformer model's accuracy is noteworthy, it is critical to grasp its relative strengths and weaknesses in comparison to existing models.

- Collecting subtle Correlations: Because the transformer excels at collecting subtle correlations in data, it is a potential model for hate speech detection.

- Early study supports the transformer's potential, but further research into fine-tuning, interpretability, and generalization across varied datasets is required.

The early research lays the groundwork for more in-depth investigations into model fine-tuning, interpretability, and generalization across various datasets. Future research should focus on resolving particular issues indicated in the confusion matrix, enabling a continuous development toward more robust and morally sound hate speech detection algorithms. We pave the path for a thorough understanding of the transformer-based hate speech detection model's function in fighting hate speech across varied mediums by including its analysis in our larger study.

Chapter 7

Conclusion

7.1 Conclusion

The identification of hate speech in online social media for a variety of major languages is a complicated issue because of the wide variety of languages and usage patterns employed by users. First, we will analyze hate speech and decide what to focus on. We found three main ways hate speech is spread. Then, mixed-language, offensive, and hate tweets and posts from Facebook and Twitter were gathered. Combining them creates a dataset. We created a range of data from the dataset. We retrieved critical features from each dataset. After collecting characteristics, we employed several machine learning and deep learning algorithms to assess if the content was hostile. We merged the findings from each algorithm and approach uniquely to reach the final result.

Accuracy ranged from 0.9201 to 0.9439 for SVM, Logistic Regression, Decision Trees, KNN, Random Forest, and Gradient Boosting models. These models show encouraging results when classifying instances of hate speech. In contrast, Naive Bayes achieved a lower accuracy of 0.4772, demonstrating that it struggled to classify instances of hate speech correctly. As hate speech identification frequently involves complicated interactions between features, this may imply that the independence assumption given by Naive Bayes does not hold well for this problem. Examining the confusion matrices reveals that the models have difficulty reliably recognizing instances of hate speech. The confusion matrices' false positives and false negatives reflect misclassifications of hate speech incidents as non-hate speech and vice versa. It is vital to highlight that selecting the best model for hate speech detection should be based on more than just accuracy or the confusion matrix. Other considerations include the problem's specific needs, computing efficiency, interpretability, and the availability of labeled data. Further study and model enhancement could improve hate speech detection performance. Techniques such as feature engineering, enhanced text preprocessing, and the use of deep learning models specifically intended for natural language processing tasks could be investigated. Furthermore, testing the models using additional metrics such as precision, recall, and F1-score would provide a more comprehensive assessment of their performance, particularly given the unbalanced nature of hate speech detection, where incidences of hate speech are often a minority class. In conclusion, while various models indicate reasonably high accuracies, further improvements and considerations are required to classify instances of hate speech and reduce misclassifications successfully.

This study uses machine learning methods to address the problem of detecting hate speech on Twitter. Initially, using the extracted characteristics, machine learning-based classifiers like LR, RF, NB, SVM, DT, GB, and KNN were used to detect HS-related tweets on Twitter. Hate speech and offensive language on Facebook and Twitter are identified in the initial data collection we are attempting to compile. Furthermore, everyday language is contaminated with hate speech and abusive vocabulary. We detect them using machine learning methods and test their accuracy. This research examines the accuracy of specific fundamental machine learning techniques utilizing deep learning and neural language processing.

Our exploration into the detection of hate speech from audio using various neural network architectures underscores the multifaceted nature of this challenge. Each model, from the Recurrent Neural Networks and Convolutional Neural Networks to the Simple Feed-forward Neural Network, brought its strengths to the fore. Particularly notable was the performance of the Simple Feed-forward Neural Network, which, despite its basic architecture, rivaled and even surpassed its more complex counterparts in certain metrics. This suggests that with appropriate data preprocessing and feature engineering, even traditional models can be harnessed effectively for such contemporary challenges.

7.2 Future Work

For future endeavors in the realm of hate speech detection from audio, several avenues beckon exploration. The augmentation of our dataset can enhance the generalizability and robustness of our models, introducing variability and potentially revealing more intricate hate speech patterns. Ensemble methods, which leverage the strengths of each individual model, offer a promising approach to boost overall performance. There's also potential in transfer learning, especially given our dataset's limited size. By fine-tuning pre-trained models from larger audio datasets, we might achieve more nuanced detection capabilities. Advanced deep learning architectures, such as attention mechanisms or transformers, which have shown substantial promise in other domains, warrant investigation. Context, an often-underestimated factor in hate speech, can be further integrated into our models. By assimilating metadata or additional information from audio sources, we can refine the accuracy of our detections. Ethical considerations remain paramount; as our models evolve, we must ensure they neither perpetuate nor amplify existing biases. Rigorous evaluation, possibly incorporating adversarial testing, can help ascertain both the effectiveness and ethical soundness of our models. Collaboration with experts from diverse fields, including sociologists, linguists, and ethicists, will undoubtedly lead to more holistic and impactful solutions in our fight against online hate speech.

1. **Data Augmentation:** To improve the generalizability and robustness of our models, we can explore techniques to augment our dataset, introducing variability and potentially uncovering more intricate patterns of hate speech.

2. **Ensemble Methods:** Leveraging the strengths of each individual model, en-

semble methods could be employed to combine predictions, potentially boosting overall performance.

3. **Transfer Learning:** Given the limited size of our dataset, pre-trained models on larger audio datasets could be fine-tuned for our specific hate speech detection task.

4. **Deep Learning Architectures:** Advanced architectures like attention mechanisms or transformers, which have shown promise in various domains, could be explored for this task.

5. **Ethical and Bias Considerations:** As we refine our models, it's crucial to ensure they don't perpetuate or amplify existing biases. Rigorous evaluation and perhaps even adversarial testing can ensure our models are both effective and ethically sound.

6. **Refinement of the Model:** Addressing particular concerns indicated in the confusion matrix will be critical for improving the transformer-based model. To ensure accessibility and ethical concerns, future research should focus on improving the interpretability of the model's judgments.

7. **Robust Generalization:** Ongoing development intends to encourage a more in-depth knowledge of the model's function, opening the way for robust and ethically sound hate speech identification across several mediums.

In the broader scheme, the fight against online hate speech is not just a technical challenge but also a societal one. While our models form a crucial line of defense, collaboration with sociologists, linguists, and ethicists will be pivotal in creating holistic and effective solutions. Understanding the model's sensitivity to perturbations is critical for real-world applications as we dig into the intricacies of hate speech identification. More study and exploration into strategies for improving robustness and minimizing adversarial effects would help to enhance hate speech detection models in the long run. This work provides a solid platform for future developments in the field of hate speech identification.

7.3 Limitations

7.3.1 Various Experimental Findings

Detecting hate speech, whether sourced from audio or text, is an intricate task laden with challenges. The nuances of human communication, such as tone, pitch, and contextual ambiguity, can often transform the meaning of words, making sarcasm in audio or humor in text tricky to decipher. Variability in language, be it accents in spoken words or slang and abbreviations in written form, further complicates detection. Additionally, the dynamic nature of hate speech, which evolves with societal changes, mandates regular model updates to maintain relevance. Data imbalance, where genuine hate speech instances are dwarfed by non-hate instances, can skew detection algorithms. There are also pressing ethical dilemmas, especially when probing personal communications. Factors like background noises in audio or the lack of context in text can lead to over-generalization, while cultural and regional differences can introduce variability in what's considered hate speech. The resource-intensive nature of processing vast audio or textual datasets, combined with the potential for false positives and negatives, underlines the complexity of the task. In essence, while the endeavor to detect hate speech is vital in our digital age, it remains a multifaceted challenge requiring continuous refinement and ethical oversight.

7.3.2 Transformer Based Approach

Several restrictions were discovered during the construction of the hate speech detection model, the majority of which stemmed from resource limits and language difficulties. As restrictions in the scope of this work, the following constraints and concerns should be acknowledged:

- Resource Constraints: The detection model was constrained by available resources, primarily RAM and GPU capacity. These constraints impacted the model's scalability and caused issues when dealing with bigger datasets or more complicated models, lowering the overall efficiency of the hate speech identification process.

- Language Proficiency: Because English is not the researcher's first language, there were inherent limits in linguistic subtleties and contextual knowledge. The potential influence of cultural or language-specific expressions on hate speech identification may be underestimated, injecting linguistic bias into the model's performance.

- Word Shuffling for resilience: To improve the model's resilience, word shuffling strategies were considered. However, the existing technical landscape made it difficult to adopt such techniques. The lack of support for word shuffling in existing technologies limited the investigation of this strategy for enhancing the model's robustness to hostile attacks.

- Incorporation of Native Language Components: The study's limitations extend to the incorporation of components peculiar to the researcher's native language. This may affect the model's generalization to diverse language and cultural situations.

- Word Restrictions: We encountered difficulties due to vocabulary and expression limits in English. The number of words and phrases that may be properly included in the hate speech detection model may be limited, thereby reducing the model's sensitivity to certain linguistic structures.

These limitations must be acknowledged since they affect the generalizability, flexibility, and comprehensiveness of the hate speech detection model. Future research might overcome these constraints by examining resource optimization strategies, including language-specific concerns, and researching new techniques for model robustness.

References

[1] Paz, M.A., Montero-Díaz, J. and Moreno-Delgado, A., 2020. Hate speech: A systematized review. Sage Open, 10(4), p.2158244020973022.

[2] Warner, W. and Hirschberg, J., 2012, June. Detecting hate speech on the world wide web. In Proceedings of the second workshop on language in social media (pp. 19-26).

[3] Nockleby, J.T., Levy, L.W., Karst, K.L. and Mahoney, D.J., 2000. Encyclopedia of the American constitution. Detroit, MI: Macmillan Reference, 3(2).

[4] Del Vigna12, F., Cimino23, A., Dell'Orletta, F., Petrocchi, M. and Tesconi, M., 2017. Hate me, hate me not: Hate speech detection on facebook. In Proceedings of the First Italian Conference on Cybersecurity (ITASEC17) (pp. 86-95).

[5] Burnap, P. and Williams, M.L., 2015. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. Policy internet, 7(2), pp.223-242.

[6] Wendling, M., 2015. The year that angry won the internet. BBC Trending.

[7] Fortuna, P. and Nunes, S., 2018. A survey on automatic detection of hate speech in text. ACM Computing Surveys (CSUR), 51(4), pp.1-30.

[8] Mullah, N.S. and Zainon, W.M.N.W., 2021. Advances in machine learning algorithms for hate speech detection in social media: a review. IEEE Access.

[9] Al-Garadi, M.A., Hussain, M.R., Khan, N., Murtaza, G., Nweke, H.F., Ali, I., Mujtaba, G., Chiroma, H., Khattak, H.A. and Gani, A., 2019. Predicting cyberbullying on social media in the big data era using machine learning algorithms: review of literature and open challenges. IEEE Access, 7, pp.70701-70718.

[10] Rodriguez, A., Argueta, C. and Chen, Y.L., 2019, February. Automatic detection of hate speech on facebook using sentiment and emotion analysis. In 2019 international conference on artificial intelligence in information and communication (ICAIIC) (pp. 169-174). IEEE.

[11] Weir, G., Owoeye, K., Oberacker, A. and Alshahrani, H., 2018, July. Cloud-based textual analysis as a basis for document classification. In 2018 International Conference on High Performance Computing Simulation (HPCS) (pp. 672-676). IEEE.

[12] Cheng, J., Danescu-Niculescu-Mizil, C. and Leskovec, J., 2015. Antisocial behavior in online discussion communities. In Proceedings of the international aaai conference on web and social media (Vol. 9, No. 1, pp. 61-70).

[13] Mullah, N.S. and Zainon, W.M.N.W., 2021. Advances in machine learning algorithms for hate speech detection in social media: a review. IEEE Access.

[14] Mehraj, H.K., Bhat, A.N. and Mehraj, H.R., 2014. Impacts of media on society: A sociological perspective. International Journal of Humanities and Social Science Invention, 3(6), pp.56-64.

[15] Muhid, A., Hadi, M., Fanani, A., Arifin, A. and Hanif, A., 2019, November. The Effect of Hate Speech Exposure on Religious Intolerance Among Indonesian Muslim Teenagers. In 2019 Ahmad Dahlan International Conference Series on Education Learning, Social Science Humanities (ADICS-ELSSH 2019) (pp. 39-44). Atlantis Press.

[16] Tontodimamma, A., Nissi, E., Sarra, A. and Fontanella, L., 2021. Thirty years of research into hate speech: topics of interest and their evolution. Scientometrics, 126(1), pp.157-179.

[17] Ring, C.E., 2013. Hate speech in social media: An exploration of the problem and its proposed solutions (Doctoral dissertation, University of Colorado at Boulder).

[18] Tsesis, A., 2002. Destructive messages: How hate speech paves the way for harmful social movements (Vol. 27). NYU Press.

[19] Karim, M., Dey, S.K., Islam, T., Shajalal, M. and Chakravarthi, B.R., 2022. Multimodal hate speech detection from bengali memes and texts. arXiv preprint arXiv:2204.10196.

[20] Rana, A. and Jha, S., 2022. Emotion Based Hate Speech Detection using Multimodal Learning. arXiv preprint arXiv:2202.06218.

[21] Badjatiya, P., Gupta, S., Gupta, M. and Varma, V., 2017, April. Deep learning for hate speech detection in tweets. In Proceedings of the 26th international conference on World Wide Web companion (pp. 759-760).

[22] Abro, S., Shaikh, S., Khand, Z.H., Zafar, A., Khan, S. and Mujtaba, G., 2020. Automatic hate speech detection using machine learning: A comparative study. International Journal of Advanced Computer Science and Applications, 11(8).

[23] Zimmerman, S., Kruschwitz, U. and Fox, C., 2018, May. Improving hate speech detection with deep learning ensembles. In Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018).

[24] Zhou, Y., Yang, Y., Liu, H., Liu, X. and Savage, N., 2020. Deep learning based fusion approach for hate speech detection. IEEE Access, 8, pp.128923-128929.

[25] Pratiwi, S.H., 2016. Detection of Hate Speech against Religion on Tweet in the Indonesian Language Using Naïve Bayes Algorithm and Support Vector Machine. B. Sc. Tesis, Universitas Indonesia, Indonesia.

[26] Alfina, I., Mulia, R., Fanany, M.I. and Ekanata, Y., 2017, October. Hate speech detection in the Indonesian language: A dataset and preliminary study. In 2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS) (pp. 233-238). IEEE.

[27] Schmidt, A. and Wiegand, M., 2017, April. A survey on hate speech detection using natural language processing. In Proceedings of the fifth international workshop on natural language processing for social media (pp. 1-10).

[28] Omar, A., Mahmoud, T.M. and Abd-El-Hafeez, T., 2020, April. Comparative performance of machine learning and deep learning algorithms for Arabic hate speech detection in osns. In The International Conference on Artificial Intelligence and Computer Vision (pp. 247-257). Springer, Cham.

[29] Velankar, A., Patil, H. and Joshi, R., 2022. A review of challenges in machine learning based automated hate speech detection. arXiv preprint arXiv:2209.05294.

[30] Roy, P.K., Tripathy, A.K., Das, T.K. and Gao, X.Z., 2020. A framework for hate speech detection using deep convolutional neural network. IEEE Access, 8, pp.204951-204962.

[31] Mousavi, S. S., Schukat, M., Howley, E. (2018). Deep reinforcement learning: an overview. In Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016: Volume 2 (pp. 426-440). Springer International Publishing.

[32] Liang, X., Du, X., Wang, G., Han, Z. (2019). A deep reinforcement learning network for traffic light cycle control. IEEE Transactions on Vehicular Technology, 68(2), 1243-1253

[33] Luong, N. C., Hoang, D. T., Gong, S., Niyato, D., Wang, P., Liang, Y. C., Kim, D. I. (2019). Applications of deep reinforcement learning in communications and networking: A survey. IEEE Communications Surveys Tutorials, 21(4), 3133-3174.

[34] Li, Y. (2017). Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274.

[35] Arulkumaran, K., Deisenroth, M. P., Brundage, M., Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. IEEE Signal Processing Magazine, 34(6), 26-38.

[36] Omar, A., Mahmoud, T. M., Abd-El-Hafeez, T. (2020). Comparative performance of machine learning and deep learning algorithms for Arabic hate speech detection in osns. In Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020) (pp. 247-257). Springer International Publishing.

[37] Watanabe, H., Bouazizi, M., & Ohtsuki, T. (2018). Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. IEEE access, 6, 13825-13835.

[38] Fernandez-Fernandez, R., Victores, J.G. and Balaguer, C., 2023. Deep robot sketching: an application of deep Q-learning networks for human-like sketching. Cognitive Systems Research, 81, pp.57-63.

[39] Cao, R. and Lee, R.K.W., 2020, December. Hategan: Adversarial generative-based data augmentation for hate speech detection. In Proceedings of the 28th International Conference on Computational Linguistics (pp. 6327-6338).

[40] Papadimitriou, C.H. and Tsitsiklis, J.N., 1987. The complexity of Markov decision processes. Mathematics of operations research, 12(3), pp.441-450.

[41] Polvara, R., Patacchiola, M., Hanheide, M. and Neumann, G., 2020. Sim-to-Real quadrotor landing via sequential deep Q-Networks and domain randomization. Robotics, 9(1), p.8.

[42] Zhou, S.K., Le, H.N., Luu, K., Nguyen, H.V. and Ayache, N., 2021. Deep reinforcement learning in medical imaging: A literature review. Medical image analysis, 73, p.102193.

[43] Farazi, N.P., Zou, B., Ahamed, T. and Barua, L., 2021. Deep reinforcement learning in transportation research: A review. Transportation research interdisciplinary perspectives, 11, p.100425.

[44] Dong, P., Chen, Z.M., Liao, X.W. and Yu, W., 2022. A deep reinforcement learning (DRL) based approach for well-testing interpretation to evaluate reservoir parameters. Petroleum Science, 19(1), pp.264-278.

[45] Al-Nima, R.R.O., Han, T., Al-Sumaidaee, S.A.M., Chen, T. and Woo, W.L., 2021. Robustness and performance of deep reinforcement learning. Applied Soft Computing, 105, p.107295.

[46] Sahidullah, M. and Saha, G., 2012. Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition. Speech communication, 54(4), pp.543-565.

[47] Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A. and Ng, A.Y., 2014. Deep speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567.

[48] Davis, S. and Mermelstein, P., 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE transactions on acoustics, speech, and signal processing, 28(4), pp.357-366.

[49] Jiang, D.N., Lu, L., Zhang, H.J., Tao, J.H. and Cai, L.H., 2002, August. Music type classification by spectral contrast feature. In Proceedings. IEEE international conference on multimedia and expo (Vol. 1, pp. 113-116). IEEE.

[50] Tzanetakis, G. and Cook, P., 2002. Musical genre classification of audio signals. IEEE Transactions on speech and audio processing, 10(5), pp.293-302.

[51] Kreuk, F., Synnaeve, G., Polyak, A., Singer, U., Défossez, A., Copet, J., Parikh, D., Taigman, Y. and Adi, Y., 2022. Audiogen: Textually guided audio generation. arXiv preprint arXiv:2209.15352.

[52] Wang, Y., Skerry-Ryan, R.J., Stanton, D., Wu, Y., Weiss, R.J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S. and Le, Q., 2017. Tacotron: Towards end-to-end speech synthesis. arXiv preprint arXiv:1703.10135.

[53] Muhammad, Bilal., Atif, Khan., Salman, Jan., Shah, M., Musa., Shaukat, Ali. (2023). Roman Urdu Hate Speech Detection Using Transformer-Based Model for Cyber Security Applications. Sensors, doi: 10.3390/s23083909

[54] Chuanpeng, Yang., Fuqing, Zhu., Guihua, Liu., Jizhong, Han., Songiln, Hu. (2022). Multimodal Hate Speech Detection via Cross-Domain Knowledge Transfer. doi: 10.1145/3503161.3548255

[55] (2022). Multimodal Amharic Hate Speech Detection Using Deep Learning. doi: 10.1109/ict4da56482.2022.9971436

[56] Abreham, Gebremedin, Debele., Michael, Melese, Woldeyohannis. (2022). Multimodal Amharic Hate Speech Detection Using Deep Learning. doi: 10.1109/ICT4DA56482.2022.9971436

[57] (2023). Hate Speech Detection using Multimodal Meme Analysis. doi: 10.1109/icaaic56838.2023.10140393

[58] Aruna, Bhat., Vaibhav, Vashisht., Sumit, Kumar, Meena. (2023). Hate Speech Detection using Multimodal Meme Analysis. doi: 10.1109/ICAAIC56838.2023.10140393

[59] Pengfei, Du., Yali, Gao., Xiaoyong, Li. (2022). Towards an Intrinsic Interpretability Approach for Multimodal Hate Speech Detection. International Journal of Pattern Recognition and Artificial Intelligence, doi: 10.1142/s0218001422500409

[60] Ona De Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. arXiv preprint arXiv:1809.04444.

[61] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the Type and Target of Offensive Posts in Social Media. In Proceedings of NAACL.

[62] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In Proceedings of the international AAAI conference on web and social media, volume 11, pages 512–515.

[63] Judea Pearl. 2009. Causal inference in statistics: An overview. Statistics surveys, 3:96–146.

[64] Dominik Janzing, David Balduzzi, Moritz GrosseWentrup, and Bernhard Schölkopf. 2013. Quantifying causal influences. The Annals of Statistics, 41(5):2324–2358