የኢፌዴሪ የቴክኒክና ሙያ ስልጠና ኢንስቲትዩት
FDRE TECHNICAL & VOCATIONAL TRAINING INSTITUTE

FEDERAL TECHNICAL VOCATION EDUCATIONAL TRAINING INSTITUTE

DEPARTMENT OF INFORMATION COMMUNICATION TECHNOLOGY

FACULITY OF ELECTRICAL ELECTRONICS AND INFORMATION COMMUNICATION TECHNOLOGY

TITLE: AUTOMATED DETECTION AND ANALYSIS OF SENTIMENT AND HATE SPEECH IN HIMTAGNE LANGUAGE SOCIAL MEDIA POSTS AND COMMENTS USING DEEP LEARNING

BY: ASHAGRIE GOSHU KASSA

SUPERVISOR BY: Dr. BIRKINESH W/YOHANNES (PhD)

ADDIS ABABA, ETHIOPIA

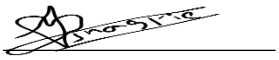
JUNE-2026

DECLARATION

I, Mr. Ashagrie, the undersigned, declare that this thesis entitled**: "Automated Detection and Analysis of Sentiment and Hate Speech in Himtagne Language Social Media Posts and Comments Using Deep Learning"** is my original work. I have undertaken the research work independently, with the guidance and support of the research advisor. This study has not been submitted for any degree or diploma program in this or any other institution, and all sources of materials used for the thesis have been acknowledged.

Declared by

Name: Ashagrie Goshu Kassa

Signature: 

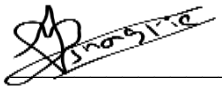
Department: Information Communication Technology

Date: 09/06/2026

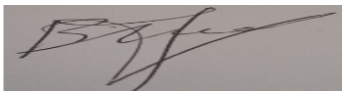
CERTIFICATION

This is to certify that the thesis prepared by Mr.***Ashagrie***, entitled "**Automated Detection and Analysis of Sentiment and Hate Speech in Himtagne Language Social Media Posts and Comments Using Deep Learning**" and submitted in partial fulfilment of the requirements for the Degree of Master in the faculty of Information Communication Technology complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Name of Candidate: Mr. Ashagrie Goshu Kassa

Signature:  Date: 03/06/2026.

Name of Advisor: Dr. Birkinesh W/Yohannes.

Signature:  Date: 03/06/2026.

Signature of the Board of Examiners:

External examiner : _____ Signature : _____ Date : _____.

Internal examiner : _____ Signature : _____ Date : _____.

Dean, SGS : _____ Signature : _____Date : _____.

ACKNOWLEDGEMENTS

First, I would like to thank God and the Virgin Mary for giving me the strength to complete this thesis and for guiding me throughout my life

I would like to thank my thesis advisor, Dr. Birkinesh W/Yohannes for her assistance. This thesis would not have been possible without the unconditional support of Dr. Birkinesh, who guided me from dataset collection to the final writing stage, providing detailed feedback and conducting regular bi-weekly meetings throughout the research period.

My Family, thank you for sharing ideas and solutions for every milestone that we have accomplished, especially Dr. Eden Bushra, for your ideas regarding the modeling process.

Finally, I must express my profound gratitude to my family for providing me with support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without your continuous support and encouragement.

TABLE OF CONTENTS

LIST OF ACRONYMS

AI	Artificial Intelligence
API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bidirectional Long Short-Term Memory
BoW	Bag of Words
CBOW	Continuous Bag of Words
CNN	Convolutional Neural Network
CSV	Comma Separated Values
DL	Deep Learning
F1	Harmonic Mean of Precision and Recall
FastText	Fast Text Word Embedding Model
GPU	Graphics Processing Unit
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IDF	Inverse Document Frequency
INSA	Information Network Security Administration
JSON	JavaScript Object Notation
KNN	K-Nearest Neighbors
LSTM	Long Short-Term Memory
mBERT	Multilingual Bidirectional Encoder Representations from Transformers

ML	Machine Learning
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
POS	Part of Speech
RNN	Recurrent Neural Network
REST	Representational State Transfer
RESTful API	Representational State Transfer Application Programming Interface
RTLM	Radio Télévision Libre des Mille Collines
SBi-LSTM	Stacked Bidirectional Long Short-Term Memory
SGD	Stochastic Gradient Descent
SMOTE	Synthetic Minority Over-sampling Technique
SVM	Support Vector Machine
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
UI	User Interface
UN	United Nations
URL	Uniform Resource Locator
UTF	Unicode Transformation Format
XML	eXtensible Markup Language
XLM-R	Cross-lingual Language Model – RoBERTa
SQL	Structured Query Language

EAII	Ethiopian Artificial Intelligence Institute
ICT	Information and Communication Technology

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Hate speech on social media has become a critical global concern, particularly in under-resourced languages where automated moderation tools are limited or unavailable. In Ethiopia, while some progress has been made for languages such as Amharic, the Himtagne language remains largely unsupported, allowing harmful online content to spread unchecked.

This study proposes an automated hate speech detection system for Himtagne social media text using deep learning techniques. A dataset of 9,258 Himtagne Facebook posts and comments was collected and manually annotated into five classes: non-hate, religious, gender-based, racial, and political hate speech. The study compares frequency-based feature extraction (TF-IDF) with embedding-based representation (FastText), and evaluates two classification models: Support Vector Machine (SVM) and Stacked Bidirectional Long Short-Term Memory (SBi-LSTM).

Experimental results show that FastText-based representations significantly outperform TF-IDF features, and the SBi-LSTM model achieves superior performance compared to SVM. The proposed model attained an F1-score of 98% with an average inference time below one second, meeting real-time deployment requirements.

The study contributes a novel annotated dataset for the Himtagne language and demonstrates the feasibility of deploying AI-based moderation tools in low-resource settings. This work supports digital safety, promotes inclusive AI development, and provides a foundation for future research in under-resourced language processing.

Keywords: Hate speech, social media, Low-resource languages, Himtagne language, Deep learning, Stacked Bidirectional LSTM (SBi-LSTM).

CHAPTER ONE: INTRODUCTION

1.1 Introduction

This chapter presents the background of the study and introduces the fundamental concepts of automated hate speech detection and sentiment analysis in Himtagne social media text. It outlines the research motivation, problem statement, objectives, research questions, scope, limitations, and the overall significance of the study.

Hate speech is a complex and context-dependent phenomenon that poses significant challenges for automated detection systems [1]. According to the Cambridge Dictionary, hate speech refers to expressions that promote hatred, discrimination, or violence against individuals or groups based on attributes such as race, religion, gender, or political affiliation [2]. The absence of a universally accepted definition introduces ambiguity in annotation and reduces the generalization capability of computational models.

Legal interpretations of hate speech vary across jurisdictions. While some countries, such as the United States, provide broad protection for speech under constitutional rights, others, particularly in Europe, criminalize expressions that incite violence or discrimination [3]. Ethiopia has adopted a regulatory approach through the Hate Speech and Disinformation Prevention and Suppression Proclamation No. 1185/2020, which emphasizes the need to control harmful digital content. However, enforcement remains limited due to the lack of automated tools for local language processing.[4]. Ethiopia has adopted a regulatory approach through the Hate Speech and Disinformation Prevention and Suppression Proclamation No. 1185/2020, which emphasizes controlling harmful digital content. However, enforcement remains limited due to the lack of automated tools for local language processing [4]. This gap highlights the urgent need for intelligent moderation systems capable of detecting harmful content in under-resourced languages. The rapid expansion of social media platforms has transformed communication by enabling users to create and share content at scale [5]. While these platforms promote information exchange and public engagement, they also facilitate the rapid spread of harmful and divisive content. In multilingual and politically sensitive environments such as Ethiopia, this challenge is further amplified by the absence of robust content moderation systems for local languages [6], [7].

In Ethiopia, this challenge is increased by the lack of neural network-based tools for local languages. Despite recent advancements in hate speech detection for high-resource languages, there is currently no automated system specifically designed for the Himtagne language [8]. Existing studies in Ethiopian languages primarily focus on Amharic and rely on limited datasets and traditional machine learning models. This creates a significant research gap in both computational linguistics and digital safety for Himtagne-speaking communities.

The Waghimra Zone has experienced periods of political instability, during which social media platforms have been used to disseminate harmful and inflammatory content. Hate speech targeting Himtagne-speaking communities has contributed to social divisions, psychological harm, and, in some cases, displacement.

This research proposes the development of an automated hate speech detection and sentiment analysis system specifically tailored to the Himtagne language. The study involves collecting and annotating a dataset of social media posts and comments, and evaluating both traditional machine learning and deep learning approaches, including Support Vector Machine (SVM) and Stacked Bidirectional Long Short-Term Memory (SBi-LSTM). Feature extraction techniques such as TF-IDF and FastText are employed to improve text representation. This approach aims to address existing limitations in low-resource language processing while providing a scalable and deployable solution for real-time content moderation.

1.2 Motivation

The increasing use of social media in Ethiopia has created digital environments where harmful language can spread rapidly, intensifying social and political tensions. Preliminary observations of Himtagne social media content indicate a noticeable rise in hate speech expressions, highlighting both a societal concern and a research opportunity [1], [2]. Preliminary observations and qualitative analysis of publicly available Himtagne social media content indicate a noticeable increase in hate speech expressions online. This trend highlights both a societal need and a scientific opportunity: while communities require mechanisms to mitigate harmful discourse, researchers face a critical gap in natural language processing resources for low-resource languages such as Himtagne.

Despite significant progress in hate speech detection for high-resource languages like English, although several studies have explored hate speech detection in Ethiopian languages such as Amharic, Tigrigna, and Afaan Oromo, these works remain limited in scope and often rely on small or domain-specific datasets. Furthermore, they rarely address languages like Himtagne, highlighting a significant research gap in low-resource language processing. Previous studies emphasize that the lack of annotated datasets, linguistic corpora, and computational tools presents major challenges for developing NLP applications in low-resource languages [9], [10]. The absence of annotated datasets, linguistic resources, and computational models for Himtagne constrains the advancement of language-specific NLP research. Addressing this gap would not only enhance digital safety and social cohesion but also contribute to the scientific understanding and computational modeling of under-resourced languages.

Therefore, this study is motivated by dual objectives: first, to reduce online hostility and promote responsible digital communication in Himtagne-speaking communities; and second, to advance low-resource NLP research by developing annotated datasets, implementing machine learning and deep learning models, and establishing baseline performance benchmarks for future studies. Prior research indicates that combining annotated datasets with machine learning and deep learning techniques significantly improves automated hate speech detection performance in social media environments [11], [12]. By integrating social relevance with scientific contribution, this research aims to provide a foundation for both practical and academic advancements in automated hate speech detection. Therefore, this study is motivated by the dual objective of improving digital safety and advancing natural language processing research for under-resourced languages through the development of robust datasets and deep learning models.

1.3 Statement of the problem

Hate speech has historically been used as a tool for marginalization, discrimination, and incitement of violence. In modern digital environments, its impact has intensified due to the widespread use of social media platforms, where harmful content can rapidly reach large audiences. Historically, ideologically driven propaganda exploited speech to justify exclusion and persecution, as seen in the Nazi regime's anti-Semitic campaigns culminating in the Holocaust [13] and in the 1994

Rwandan Genocide, where media outlets such as Radio Television Libre des Mille Collines (RTLM) propagated messages that incited violence against the Tutsi population [14].

In contemporary Ethiopia, the widespread adoption of social media platforms such as Facebook, Telegram, and TikTok has expanded opportunities for communication, information exchange, and public discourse. However, these platforms have also facilitated the rapid dissemination of hate speech, particularly in ethnically diverse and politically sensitive regions [15], [16]. The Himtagne-speaking population in the Waghimra Zone has been affected by targeted online hostility, contributing to social divisions, mistrust, and real-world conflicts [17].

The prevalence of misinformation, inflammatory political claims, and unverified accusations on social media exacerbates these challenges [18]. Reports highlight how platforms like Facebook have been implicated in the spread of harmful content due to inadequate moderation practices [19]. Unregulated digital communication environments can therefore amplify hate speech, potentially inciting offline violence and negatively influencing societal behavior, including youth exposure to aggressive discourse [20].

Currently, there is no automated system capable of detecting hate speech in Himtagne, creating a significant gap in Ethiopia's digital moderation infrastructure. While automated detection exists for languages such as English and Amharic, the lack of annotated datasets, preprocessing techniques and computational models for Himtagne prevents effective monitoring and mitigation of online hate speech. This research seeks to address this critical gap by developing and evaluating machine learning and deep learning models specifically designed for Himtagne social media text, providing both technological solutions and a foundation for low-resource NLP research.

1.4 Research Questions

This study aims to address the following research questions:

1. How can deep learning models be effectively adapted for hate speech detection in the Himtagne language under limited training data conditions?
2. Which preprocessing techniques best address the morphological complexity and spelling variation of Himtagne social media text?

3. Does embedding-based feature representation (FastText) significantly outperform frequency-based representation (TF-IDF) for Himtagne hate speech detection?
4. Does a stacked Bidirectional LSTM (SBi-LSTM) model significantly outperform a traditional SVM classifier in detecting Himtagne hate speech?
5. Can an automated hate speech detection system achieve real-time performance suitable for deployment in social media moderation systems?

1.5 Research Hypotheses

To statistically validate the research questions, the following hypotheses are formulated:

H1: Embedding-based features (FastText) significantly improve classification performance compared to TF-IDF features in Himtagne hate speech detection.

H2: The SBi-LSTM model achieves significantly higher F1-score and precision than the SVM classifier.

H3: Language-specific preprocessing techniques significantly improve model performance compared to basic preprocessing.

H4: The proposed deep learning model achieves an F1-score ≥ 85% on the test dataset.

H5: The deployed system processes input text within ≤ 1 second inference time per request.

1.6 Objectives

1.6.1 General Objective

To design, implement, and evaluate an automated deep learning-based hate speech detection system for Himtagne social media text with measurable performance and deployable architecture.

1.6.2 Specific Objectives

- To collect and annotate a balanced dataset of at least 9,258 Himtagne social media posts for hate speech classification.

- To design and implement preprocessing techniques that address:
    - Morphological normalization
    - Spelling variation
    - Tokenization for Ge'ez script

- To compare TF-IDF and FastText feature representations using:
    - Accuracy
    - Precision
    - Recall
    - F1-score

- To train and evaluate**:**
    - A Support Vector Machine (SVM) classifier
    - A Stacked Bidirectional LSTM (SBi-LSTM) model
- To statistically compare model performance using appropriate validation techniques (e.g., 5-fold cross-validation).

- To achieve**:**
    - Target F1-score ≥ 85%
    - Precision ≥ 85%
    - Inference time ≤ 1 second per input
- To deploy the best-performing model as a RESTful API for real-time moderation.

1.7 Scope of the Study

The study focuses exclusively on detecting hate speech in Himtagne language text using deep learning. The dataset was prepared by collecting Himtagne text posts and comments from popular public Facebook pages from January 2020 to January 2025. Posts and comments were annotated into five distinct classes: non-hate speech, religion hate speech, gender hate speech, race hate speech, and political hate speech. This study evaluates both traditional machine learning and deep learning classifiers for hate speech detection. The models are assessed using standard performance metrics, including accuracy, precision, recall, and F1-score, to determine their effectiveness in classifying Himtagne social media text.

1.8 Significance of the Study

This study provides one of the first comprehensive attempts to develop a deployable hate speech detection system for the Himtagne language, addressing a critical gap in low-resource language processing. It also contributes a new annotated dataset and promotes safer online communication. The findings of this research have implications for policymakers, social media platforms, and researchers by providing a scalable solution for monitoring harmful content and promoting safer digital communication environments.

Furthermore, it supports marginalized linguistic communities and advances language-based AI tools. It provides a technical foundation for integrating automated moderation systems into local social media platforms and encourages the development of responsible digital ecosystems.

Furthermore, the research aligns with national digital governance efforts in Ethiopia, particularly under the Hate Speech and Disinformation Prevention and Suppression Proclamation No. 1185/2020. It demonstrates how AI-driven systems can assist in detecting and managing harmful content. Similar to international regulatory frameworks that require large social media platforms to remove clearly illegal content within strict timeframes, stores deleted content for accountability, and submit transparency reports; this study highlights the importance of combining technological enforcement with constitutional values and digital responsibility.

By assisting policymakers with data-driven insights, supporting automated content moderation, and strengthening national cybersecurity strategies through collaboration with institutions such as the Ethiopian Artificial Intelligence Institute and the Information Network Security Administration, the proposed system demonstrates strong policy relevance and societal value.

In addition, the study outlines clear pathways for institutional integration and long-term impact. At the government level, the developed model can be incorporated into social media monitoring systems, early-warning mechanisms for online hate escalation, and digital forensic tools used by cybercrime investigation units. Within educational institutions, the system can serve as a foundation for further NLP research on Himtagne, curriculum development in artificial intelligence and computational linguistics programs, and student-led model enhancements such as transformer-based upgrades.

Beyond technical implementation, the research also contributes to the preservation and digital empowerment of the Himtagne language. It promotes the inclusion of low-resource languages in global AI research and supports efforts to reduce ethnic, religious, and political tensions through proactive detection mechanisms. Ultimately, this study integrates technological innovation with social responsibility to foster inclusive, secure, and culturally respectful digital spaces.

1.9 Limitations of the Study

This study has several limitations that should be acknowledged to ensure a balanced and critical evaluation of the findings.

Another limitation involves linguistic ambiguity challenges inherent in the Himtagne language. Hate speech expressions can be indirect, sarcastic, metaphorical, or highly context-dependent. Words that appear neutral in isolation may become offensive depending on discourse context. Dialectal variations, morphological richness, and code-switching with other Ethiopian languages further complicate automated classification. Since the current model primarily analyzes sentence-level text without broader conversational or discourse-level context, it may struggle to detect implicit or nuanced hate expressions. Future research could incorporate contextual modeling, transformer-based architectures, or conversation-level analysis to address these challenges.

The study is restricted in scope and modality. It focuses solely on text-based content, specifically Facebook posts and comments written in Himtagne, and does not include audio, video, emojis, or images. Moreover, it does not consider other major platforms such as Twitter, YouTube, TikTok, or Telegram. The research is limited to five multiclass categories religion hate, gender hate, race hate, political hate, and non-hate speech and does not explore finer-grained subcategories such as color-based, country-based, skin-tone-based, or disability-related hate speech. Expanding platform coverage and classification granularity would improve generalizability and practical applicability.

Finally, hardware and computational limitations influenced the experimental design. The model was trained in a Google Colab environment with limited GPU session time and memory constraints, which restricted experimentation with transformer-based architectures such as mBERT and XLM-R. In addition, these models require large-scale annotated datasets to achieve optimal performance, which are currently unavailable for the Himtagne language. Therefore, this

study prioritizes computationally efficient models such as SVM and SBi-LSTM, which are more suitable for low-resource settings, extensive hyperparameter tuning, and training on substantially larger datasets. Deployment on shared hosting infrastructure may also limit scalability under high user traffic. Future studies could utilize dedicated high-performance computing resources or scalable cloud infrastructure to enhance model complexity, training stability, and real-time deployment performance.

Despite these limitations, the study remains highly relevant. The proposed system provides a foundational framework for Himtagne hate speech detection, establishes annotated datasets for future research, and demonstrates the feasibility of deploying automated moderation tools in under-resourced language settings. These contributions outweigh the constraints and offer significant value to both technological and social applications.

1.10 Organization of the paper

This thesis report is organized into six chapters, each focusing on a specific part of the research to ensure clarity and logical flow.

Chapter one presents the introduction of the study. It provides the background of hate speech and sentiment analysis, particularly in the context of social media and under-resourced languages such as Himtagne. This chapter explains the motivation of the study, statement of the problem, research questions, objectives, scope, limitations, and the significance of the research. It offers an overall understanding of why the study is necessary and what it aims to achieve.

Chapter Two is divided into two parts

- Discusses the theoretical background and fundamental concepts related to hate speech, sentiment analysis, and Natural Language Processing (NLP), machine learning, and deep learning. It explains key techniques such as text preprocessing, feature extraction methods and classification algorithms including SVM and SBi-LSTM, to help readers understand the technical foundation of the study.
- Reviews related works conducted on hate speech detection and sentiment analysis in both high-resource and under-resourced languages, with special attention to Ethiopian

languages such as Amharic, Tigrigna, and Awigna. This section highlights existing research gaps that justify the need for the current study.

Chapter Three describes the methodology, system architecture, and design of the proposed hate speech detection system. It explains the data collection process, dataset annotation, preprocessing steps, feature extraction techniques, model selection, and training procedures. The overall framework of the proposed system is clearly illustrated and justified in this chapter.

Chapter Four presents the Results and Discussion of the proposed system implementation. It describes how the models were developed and implemented, including the tools, frameworks, and technologies used in the process. The chapter explains how the system processes Himtagne social media text for hate speech detection and sentiment analysis.

Furthermore, it presents the experimental results and the overall performance evaluation of the system. Evaluation metrics such as accuracy, precision, recall, and F1-score are used to measure model performance. A comparative analysis is conducted between different models, including Support Vector Machine (SVM) and Stacked Bidirectional Long Short-Term Memory (SBi-LSTM). The results are analyzed and interpreted to assess the effectiveness and robustness of the proposed approach.

Chapter Five, the final chapter, provides the conclusion and future work. It summarizes the main contributions and findings of the study, discusses the limitations, and suggests directions for future research, including expanding the dataset, incorporating multimodal content, and extending the system to other Ethiopian languages and social media platforms.

1.11 Definition of key terms

For the purposes of this study, the following key terms are used as defined below:

Hate Speech refers to any form of text-based communication that expresses hatred, discrimination, hostility, or encourages violence against an individual or group based on attributes such as religion, gender, race, or political affiliation. In this study, hate speech is identified from social media posts and comments written in the Himtagne language.

Sentiment Analysis is a natural language processing technique used to determine the emotional tone of a text. It classifies text into different sentiment categories such as positive, negative, or neutral. In this research, sentiment analysis is applied alongside hate speech detection to understand the emotional context of Himtagne social media content.

Himtagne language is a local Ethiopian language predominantly spoken in the Waghimra zone. It is considered an under-resourced language due to the lack of digital text corpora, computational tools, and language processing resources.

Social media posts and comments refer to user-generated text content shared on online platforms such as Facebook. In this study, posts and comments written in Himtagne are collected and analyzed to detect hate speech and sentiment patterns.

Text preprocessing is the process of cleaning and preparing raw textual data for analysis. It includes steps such as removing punctuation, stop words, unnecessary symbols, normalization, and tokenization to improve the performance of machine learning and deep learning models.

Feature extraction is a technique used to represent words in numerical vector form while preserving semantic meaning. In this study, frequency-based and embedding-based methods such as TF-IDF and FastText are used to capture the contextual and semantic features of Himtagne text.

TF-IDF (Term Frequency-Inverse Document Frequency) is a statistical feature extraction method that measures the importance of a word in a document relative to a collection of documents. It helps identify discriminative words useful for hate speech classification [21].

FastText is a word embedding method that represents words using character n-grams. It is particularly effective for morphologically rich and under-resourced languages like Himtagne, as it can handle misspellings and rare words [12], [22].

Machine learning is a branch of artificial intelligence that enables systems to learn patterns from data and make predictions without explicit programming instructions. In this study, machine learning techniques are used for text classification tasks.

Deep learning is a subset of machine learning that uses neural networks with multiple layers to model complex patterns in data. It is used in this research to automatically detect hate speech and analyze sentiment in Himtagne text.

Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification tasks. It works by finding an optimal boundary that separates different classes of data [23]. In this study, SVM is used as a baseline classifier for hate speech detection.

Stacked Bidirectional Long Short-Term Memory (SBi-LSTM) is a deep learning model that processes text sequences in both forward and backward directions using multiple LSTM layers. It is capable of capturing long-term dependencies and contextual information in Himtagne social media text [24].

Multiclass classification is a machine learning task where text is categorized into more than two classes. In this study, it includes non-hate speech, religion hate speech, gender hate speech, race hate speech, and political hate speech.

Evaluation metrics are quantitative measures used to assess model performance. Common metrics used in this study include accuracy, precision, recall, and F1-score.

Automated detection system refers to a computer-based system that automatically identifies hate speech and sentiment in text without human intervention. The proposed system in this study is designed to support social media moderation for Himtagne language content.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

The purpose of this literature review is to examine existing research on hate speech detection, including both traditional and modern computational approaches. In addition, since this thesis focuses on the Himtagne language, this review provides a linguistic and computational overview of the language to establish the necessary research context. The chapter also discusses the historical and sociolinguistic background of the Himtagne language to support understanding of its structural and computational challenges in natural language processing tasks [25], [26].

2.2 Definition of Hate Speech

Hate speech generally refers to any form of communication that attacks, discriminates against, or promotes hostility toward individuals or groups based on their identity characteristics. According to the United Nations Strategy and Plan of Action on Hate Speech, hate speech includes any communication expressed through speech, writing, or behavior that uses offensive or discriminatory language targeting individuals or groups based on attributes such as religion, race, gender, or political affiliation [27]. Similarly, other scholars describe hate speech as language that promotes violence, hatred, or discrimination against groups based on physical appearance, ethnicity, religion, gender, or political ideology [6], [7]. In addition, the Cambridge Dictionary defines hate speech as public communication that expresses hatred or encourages violence toward individuals or groups based on protected characteristics such as race, religion, gender, or political identity [2].

Synthesizing these definitions, hate speech can be broadly understood as communication that promotes discrimination, hostility, or violence against individuals or groups based on socially or culturally defined identity characteristics [9]. In the context of this study, hate speech is particularly examined as harmful online textual content that targets vulnerable social groups through explicit or implicit linguistic expressions on social media platforms.

2.3 Legal Recognition and Regulation of Hate Speech

The legal recognition of hate speech emerged primarily in response to historical atrocities such as the Holocaust, which prompted the development of international human rights protection frameworks [28]. The Universal Declaration of Human Rights and the International Covenant on Civil and Political Rights established fundamental principles requiring states to protect individuals and groups from discriminatory speech that incites hostility, violence, or social exclusion [29], [30]. These international instruments form the foundation for global hate speech regulation by balancing freedom of expression with the protection of human dignity.

At the regional level, European legal frameworks further expanded the interpretation of hate speech. For example, the Council of Europe's Recommendation No. R (97) 20 defined hate speech broadly to include expressions that promote racism, xenophobia, and other forms of intolerance [31]. This demonstrates the increasing recognition that hates speech extends beyond direct verbal violence to include symbolic and ideological forms of discrimination.

With the rapid expansion of digital communication platforms, online hate speech has become a major global policy challenge. Governments and technology companies have therefore implemented regulatory and content moderation mechanisms to reduce the spread of harmful online content. However, regulating online speech remains complex due to the need to balance public safety with freedom of expression rights.

In Ethiopia, social and political tensions have contributed to the increasing prevalence of online hate speech and misinformation. This situation led to the enactment of the Hate Speech and Disinformation Prevention and Suppression Proclamation No. 1185/2020 [32]. The law aims to reduce social harm caused by discriminatory and inflammatory speech while maintaining national stability. However, implementation of such regulations has raised concerns regarding potential restrictions on freedom of expression, highlighting the continuing global debate between security enforcement and civil liberty protection in digital communication environments.

2.4 Hate Speech in Social Media

Social media platforms have transformed global communication by enabling rapid information sharing, public discourse, and digital community formation. However, these platforms have also facilitated the rapid dissemination of harmful and discriminatory content [33]. Platforms such as Facebook host billions of active users worldwide, making them powerful communication environments where both positive engagement and harmful speech can spread quickly [34].

The impact of social media–based hate speech has been demonstrated in several international conflicts. For example, United Nations investigations linked Facebook with the spread of inflammatory content during the Rohingya crisis in Myanmar, where online communication amplified ethnic violence [35].

In Ethiopia, social media has become an important platform for political discussion and information dissemination. However, increasing ethnic and political polarization has been reported in online discourse [36]. Automated moderation remains inconsistent due to limited computational resources for Ethiopian languages such as Amharic, Afaan Oromo, and Tigrigna [37].

This challenge is more significant for under-resourced languages such as Himtagne, where limited digital linguistic resources, spelling variations, and morphological complexity reduce the effectiveness of automated moderation systems [25].

2.5 Types of Hate Speech

Hate speech can be categorized based on the identity characteristics targeted by offensive or discriminatory language. Common targets of hate speech include ethnicity, religion, race, gender, disability, and political affiliation. These categories are widely recognized in international human rights discussions and are also reflected in national legal frameworks.

In Ethiopia, empirical studies suggest that ethnicity-based, political, religious, and gender-related hate speech are among the most commonly observed forms on social media platforms [25].

In the Ethiopian digital communication context, several studies examining social media discourse has shown that hate speech frequently emerges around issues of ethnicity, politics, religion, and

gender. The rapid expansion of social media platforms such as Facebook, YouTube, and Twitter has created spaces for political debate and public expression; however, these platforms have also been used to spread hostile narratives and discriminatory expressions. Research analyzing Ethiopian social media data indicates that many hateful messages target identity groups and are often connected with ongoing political tensions, ethnic competition, and ideological disagreements. These patterns highlight how online communication can amplify social divisions when hostile narratives circulate widely without moderation [38].

Furthermore, research examining hate speech in Amharic social media datasets confirms that many online posts and comments contain hate speech directed toward ethnic groups, religious communities, and political actors. Studies that collected and annotated thousands of Ethiopian social media posts have identified hate speech categories such as racial or ethnic hate, religious hostility, and politically motivated attacks. These categories are particularly relevant in multilingual and multiethnic societies where identity plays an important role in political and social discourse. The identification of these types of hate speech is essential for developing automated detection systems capable of recognizing harmful language in Ethiopian languages such as Amharic and Himtagna [39].

In the Ethiopian digital communication context, empirical studies suggest that ethnicity-based, political, religious, and gender-related hate speech are among the most frequently observed categories on social media platforms. However, the relative distribution of these categories should be interpreted as research-based trends rather than precise statistical measurements, since exact prevalence rates vary depending on dataset selection, time period, and social events.

2.5.1 Race and Ethnicity-Based Hate Speech

Ethnicity-based hate speech is widely reported as one of the dominant forms of online hostile communication in Ethiopia. Social media content analysis studies indicate that ethnic identity is frequently used as a target of discrimination and dangerous speech narratives. This is particularly evident during periods of political tension and social instability, where ethnic identity is often used as a marker for political or ideological conflict [38].

2.5.2 Political-Based Hate Speech

Political hate speech is also highly prevalent in Ethiopian social media discourse. Online political discussions frequently contain hostile language, particularly during election periods and political debates. Political hate speech often overlaps with ethnic narratives, making detection more complex due to contextual and ideological language patterns [38].

2.5.3 Religion-Based Hate Speech

Religion-based hate speech occurs less frequently compared to ethnicity and political-related hate speech. However, religiously motivated hostility can increase during periods of inter-religious tension. Studies of Ethiopian social media discourse show that religious hate speech often appears in the form of subtle linguistic expressions rather than direct offensive statements [38].

2.5.4 Gender-Based Hate Speech

Gender-based hate speech also appears in Ethiopian online communication environments. Such speech often targets women and gender minority groups through sexist or discriminatory expressions. Although this category represents a smaller proportion of online hate speech compared to ethnic or political forms, it remains significant due to its social impact and persistence in digital discourse [39].

2.6 Consequences of Hate Speech

Hate speech can have serious social, psychological, and political consequences if not properly addressed at early stages. Research in social conflict and communication studies shows that even seemingly minor forms of discriminatory language, such as rumors, jokes, or stereotypical expressions, can gradually escalate into more severe forms of social harm. According to the Center for Analysis of Radicalization and Violence, harmful speech often contributes to processes of social polarization, discrimination, and intergroup hostility [28].

The development of hate speech can be understood as a progressive escalation process. At the initial stage, hate speech often appears in the form of bias and prejudice, where individuals or groups express negative stereotypes, discriminatory jokes, or exclusionary narratives. If left

unchallenged, this may progress to structural discrimination, where targeted groups may experience social, economic, or institutional exclusion in areas such as education, employment, and public services.

At more advanced stages, hate speech can contribute to direct forms of social and physical violence, including harassment, threats, property damage, and physical attacks. In extreme cases, sustained and systematic dissemination of hateful ideology has historically contributed to large-scale violence and social conflict. Therefore, early detection and intervention of hate speech is important for maintaining social stability and protecting vulnerable communities in both offline and online communication environments.

2.7 Overview of the Himtagne Language

The Himtagne language belongs to the Agew language group, which is part of the Central Cushitic branch of the Afro-Asiatic language family [40]. The language is primarily spoken by the Kamyr Agew community in the Waghimra zone of the Amhara regional state of Ethiopia. Like other Agew languages, Himtagne shares historical and structural similarities with Awngi, Bilen, and Qimant languages [41].

Despite its cultural and administrative importance, Himtagne remains highly under-resourced in computational linguistics, particularly in natural language processing and digital language technology development [42].

2.7.1 Himtagne Alphabet and Writing System

Himtagne uses the Ge'ez (Ethiopic) writing system, which is a syllabic script where consonants combine with vowel markers to form distinct syllabic characters. The language contains 36 core base characters consisting of 29 consonants and 7 vowel forms, generating approximately 306 syllabic character combinations. This structural property increases linguistic expressiveness but introduces computational complexity for automatic text processing tasks [43].

The detailed list of Himtagne base consonants, extended syllabic forms, and phonetic equivalent character variations are presented in Annex A (Table A.1 – A.3)**.** These linguistic tables are

important for preprocessing design in natural language processing applications, particularly for tasks such as normalization and spelling standardization [42].

2.7.2 Computational Challenges of the Himtagne Writing System

The Ge'ez script presents several computational challenges due to multiple graphical representations of similar phonetic sounds. For example, some consonants may have redundant graphical variants that represent the same phoneme. These variations create ambiguity in text classification and require normalization procedures before feature extraction [43].

Examples of phonetic equivalent character variations and abbreviation forms are presented in Annex A (Table A.3 – A.4)**.** These tables provide reference linguistic mappings used during dataset preprocessing and text cleaning operations [42].

Therefore, automated hate speech detection systems for Himtagne must integrate robust preprocessing pipelines including character normalization, tokenization, and subword feature extraction. These steps improve model robustness against spelling variation, code-switching, and informal social media writing patterns [42].

2.8 Techniques for Automated Hate Speech Detection

Current Methods for Hate Speech Detection

Hate speech detection is a text classification task that aims to automatically identify offensive, abusive, or hateful content in online platforms. Due to the complexity of language, researchers employ a range of machine learning and deep learning algorithms [42].

This section presents the most widely used algorithms in hate speech detection research.

Research on hate speech detection has expanded rapidly, but most studies focus on well-resources language. Existing methods can be broadly categorized into: traditional machine learning approach and deep learning approach. These methods assist in identifying harmful language patterns and inform the development of models for under-resources language [44].

Most existing hate speech detection studies focus on high-resource languages and rely heavily on large annotated datasets. In Ethiopian contexts, research is limited and primarily concentrated on Amharic, with minimal attention given to other languages such as Himtagne.

Traditional machine learning approaches such as SVM perform well for explicit hate speech detection but fail to capture contextual and implicit meanings. Deep learning models, particularly Bi-LSTM, improve contextual understanding but require more computational resources and data.

Furthermore, previous studies rarely address linguistic challenges such as morphological variation, spelling inconsistency, and code-switching, which are common in Himtagne social media text. This study addresses these limitations by integrating FastText embeddings with a stacked Bi-LSTM model to improve classification performance.

2.8.1 Text Representation and Feature Extraction

Text representation transforms raw text into numerical features for machine learning models in hate speech detection. Frequency-based methods like Bag of Words (BoW) and TF-IDF excel at capturing explicit word patterns in short texts, while embedding-based methods like Word2Vec and FastText handle semantic context better, especially in morphologically rich languages like Amharic [45].

2.8.1.1 Frequency-Based Methods

Text is converted into numerical vectors based on word occurrences in a document [46]. Each vector element represents a word's frequency or weighted frequency. Unlike embedding-based methods, they do not capture meaning or context, focusing instead on explicit word patterns useful for detecting direct or keyword-based hate speech.

Bag of Words (BoW)

It's one of the simplest and most widely used text representation techniques. It converts text into numerical vectors based on the frequency of individual words, without considering grammar, word order, or semantic meaning. Each document is represented as a vector where each element corresponds to a word from a predefine vocabulary.

The BoW process begins by collecting textual data such as sentences, comments, or social media posts. The text is then cleaned by removing punctuation, converting words to lowercase, and eliminating unnecessary symbols. After preprocessing, a vocabulary is created by listing all unique words appearing in the dataset. Each document is finally represented as a frequency vector indicating how many time vocabularies.

For example, the sentences “this post is bad” and “this post is very bad” share most vocabulary words, but the second sentence includes the additional work very. These sentences are converted into numerical vectors based on word counts, which can then be used as input for classification models [47].

Bag of Words is simple to implement and computationally efficient. It performs well in detecting explicit insults and keyword-based hate speech, particularly in short abusive text. However, it has important limitations. BoW ignores context and word order which can lead to misinterpretation of meaning, such as treating “not bad” as negative. It also fails to capture implicit or culturally coded hate and performs poorly for morphologically rich language like Amharic, where a single word many appear in many forms [48].

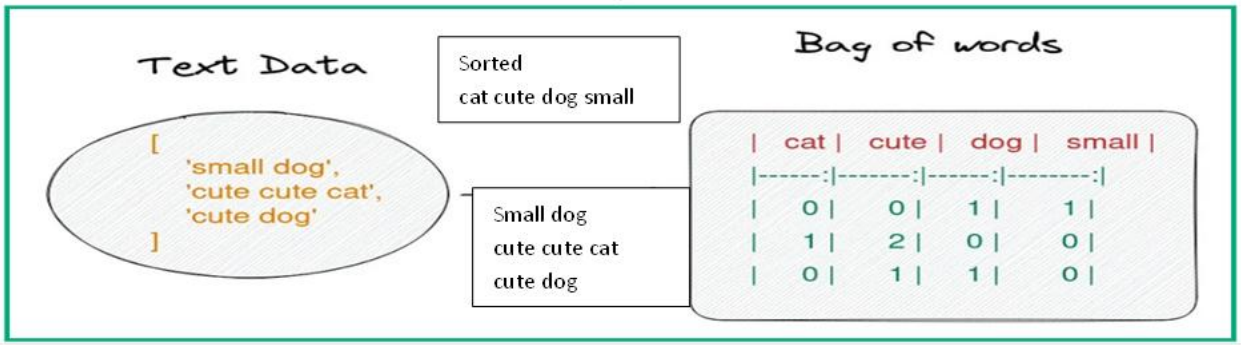


Figure 1 Bag of Words

The Bag-of-Words (BoW) model is a widely used text representation technique in natural language processing that converts textual data into numerical vectors suitable for machine learning algorithms. In this approach, a collection of text documents is first analyzed to extract a vocabulary of unique words. Each document is then represented as a vector that records the frequency of each

word from the vocabulary within that document. As illustrated in the figure, the text data contains example sentences such as “small dog,” “cute cute cat,” and “cute dog.” From these sentences, a vocabulary set is generated consisting of the words *cat, cute, dog,* and *small*. Each sentence is then transformed into a numerical representation based on the number of times each word appears in the sentence. For instance, the sentence “small dog” is represented as (0, 0, 1, 1), indicating that the words *dog* and *small* appear once while *cat* and *cute* do not appear. Similarly, “cute cute cat” is represented as (1, 2, 0, 0), reflecting the presence of one occurrence of *cat* and two occurrences of *cute*. Through this process, textual information is converted into a structured numerical matrix that can be used as input for machine learning or deep learning models. Although the Bag-of-Words model effectively captures word frequency, it ignores word order and contextual relationships between words. Nevertheless, it remains a simple and effective baseline method for many text classification tasks, including sentiment analysis and hate speech detection.

Term Frequency-Inverse Document Frequency (TF-IDF)

It’s an extension of BoW model that assigns importance weights to words rather than relying only on raw frequency. It increases the weights of words that are frequency in a specific document but rare across the entire dataset, while reducing the impact of common words such as “is” or “the”.

The TF-IDF process involves tokenizing text into words, calculating term frequency (TF) for each word in a document, and computing inverse document frequency (IDF) based on how many documents contain that word. The final TF-IDF score is obtained by multiplying TF and IDF, producing a weighted vector representation [49].

Term Frequency X Inverse Document Frequency

$$w_{x,y} = tf_{x,y} \times log(\frac{N}{df_x})$$

Text1: Basic Linux Commands for Data Science

Text2: Essential DVC Commands for Data Science

	basic	commands	data	dvc	essential	for	linux	science
Text 1	0.5	0.35	0.35	0.0	0.0	0.35	0.5	0.35
Text 2	0.0	0.35	0.35	0.5	0.5	0.35	0.0	0.35

Figure 2 Term Frequency Inverse Document Frequency

TF-IDF is effective for highlighting explicit discriminatory terms in hate speech classification. However, its inability to capture semantic meaning and contextual relationships limits its performance, particularly in detecting implicit or context-dependent hate expressions. In contrast, embedding-based methods such as FastText better capture semantic relationships, making them more suitable for morphologically rich and low-resource languages like Himtagne. It also struggles with morphological variation, which is common in Ethiopia languages such as Amharic [50].

The Term Frequency–Inverse Document Frequency (TF–IDF) method is an advanced feature extraction technique used in natural language processing to measure the importance of words in a document relative to a collection of documents. Unlike the Bag-of-Words model, which only counts word occurrences, TF–IDF assigns weights to words based on how frequently they appear in a document and how rare they are across the entire dataset. Mathematically, TF–IDF is calculated using the formula: $W_{x,y} = tf_{x,y} * \log(\frac{N}{dfx})$, where $tf_{x,y}$ represents the term frequency of word xxx in document y, N is the total number of documents in the corpus, and dfx is the

number of documents that contain the word x. As illustrated in the figure, two example documents are used: “Basic Linux Commands for Data Science” and “Essential DVC Commands for Data Science.” First, a vocabulary is created from all unique terms such as *basic, commands, data, dvc, essential, for, Linux,* and *science*. Then, the TF–IDF score for each word in each document is computed. Words that appear frequently in a specific document but rarely in other documents receive higher weights, while common words that appear in many documents receive lower weights. For example, the word *Linux* receives a higher weight in Text 1 because it appears specifically in that document, whereas the word *for* receives a lower weight because it appears in both texts. The resulting TF–IDF matrix represents each document as a weighted numerical vector, which can then be used as input for machine learning or deep learning models. This approach improves text classification performance by emphasizing informative words and reducing the influence of commonly occurring terms.

2.8.1.2 Embedding–Based Methods

It’s represented words as dense vectors that capture contextual and semantic relationships. Unlike frequency-based approaches, these methods learn meaning from word usage patterns.

Word2Vec

It’s learning word representations by analyzing the surrounding words within a context window. Words appearing in similar contexts receive similar vector representations. It uses two training strategies: continuous bag of word (CBoW) and Skip-gram. Skip-gram is often preferred for smaller datasets.

Word2Vec captures semantic and syntactic relationships more effectively than frequency-based methods such as BoW and TF-IDF. However, its performance is limited in low-resource settings due to its inability to handle rare or out-of-vocabulary words. FastText addresses this limitation by incorporating character-level n-grams, making it more robust for morphologically rich languages like Himtagne. However, it requires large corpora for effective training. Performs poorly on rare or unseen words, and is less suitable for low-resource languages without extensive preprocessing [48].

Text Corpus

Tokenization -> Build Vocabulary

One-Hot Encoding

Neural Network (CBOW or Skip-gram)

Word Embedding Vectors (Dense Vectors)

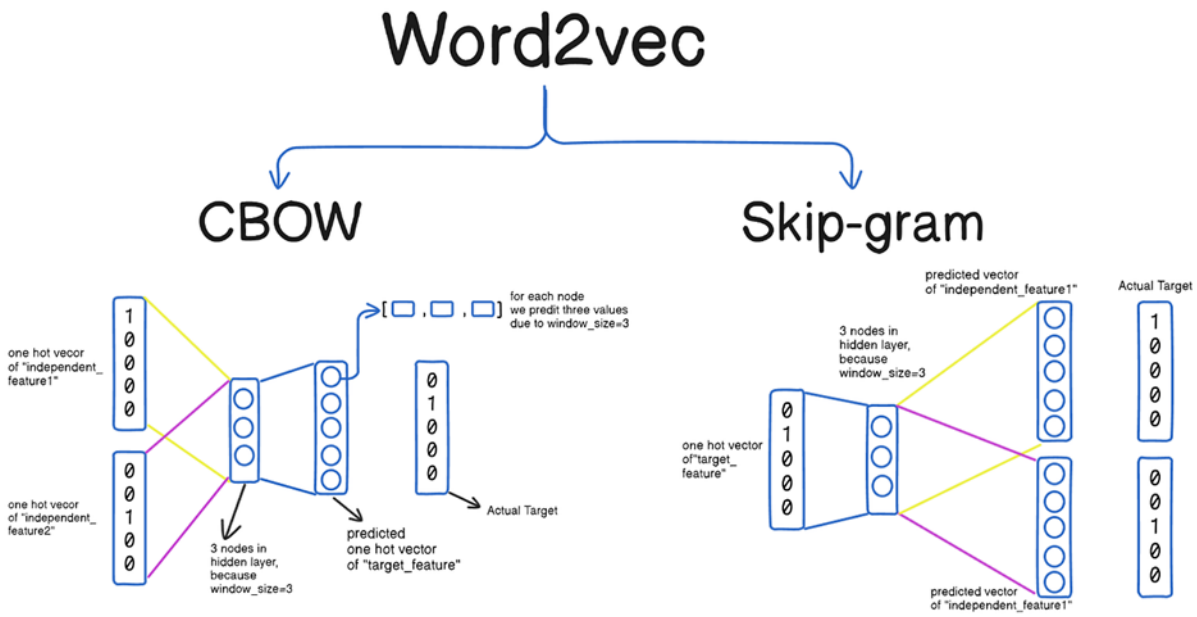


Figure 3 Word2Vec

The Word2Vec model is a neural network–based word embedding technique used in natural language processing to represent words as dense numerical vectors. Unlike traditional methods such as Bag-of-Words or TF–IDF, which treat words independently, Word2Vec captures semantic relationships and contextual similarity between words. The model learns word representations by analyzing the context in which words appear within a large corpus of text. As shown in the figure, Word2Vec has two main architectures: Continuous Bag of Words (CBOW) and Skip-gram.

The Continuous Bag of Words (CBOW) model predicts a target word based on its surrounding context words. In this approach, several neighboring words are provided as input, and the model learns to predict the missing or center word. For example, in the sentence context “the quick brown

jumps", the surrounding words *the, quick, brown,* and *jumps* are used as input, while the target output is the missing word "fox." During training, the context words are converted into one-hot vectors and passed through a neural network with a hidden layer that learns a dense vector representation for each word. The network then predicts the probability distribution of the target word from the vocabulary.

In contrast, the Skip-gram model performs the opposite task. Instead of predicting the target word from the context, it uses a single target word to predict its surrounding context words. For instance, if the input word is "fox," the model attempts to predict the neighboring words "the," "quick," "brown," and "jumps." The Skip-gram architecture is particularly effective for learning high-quality word representations when dealing with smaller datasets or when capturing relationships between rare words. Both CBOW and Skip-gram use neural networks to learn word embeddings that capture semantic meaning, allowing similar words to have similar vector representations.

FastText

It's improving upon Word2Vec by representing words using character-level n-gram instead of treating each word as a single unit. This allows the model to generate embedding for rare, misspelled, or unseen words based on shared sub word information [51].

Figure 4 FastText

FastText is particularly effective for morphologically rich languages, works well with small dataset, and handles spelling variation effectively However, it is slightly slower than Word2Vec and requires careful parameter tuning [52].

The figure illustrates how FastText uses subword (character n-gram) information to build a word representation and predict surrounding words in a sentence. Consider the example sentence "The

quick brown fox jumps", where the target word is "fox." Instead of representing the word *fox* as a single vector like traditional word embedding models, FastText breaks the word into character n-grams. If we use n = 3 to 5, boundary symbols are added to the word, producing **<fox>**. From this form, several subwords are generated such as 3-grams (<fo, fox, ox>), 4-grams (<fox, fox**>)**, and 5-grams (<fox>). Each of these n-grams has its own vector representation.

During training, the vectors of all subwords belonging to the word are summed (as shown by the SUM block in the figure) to produce the final vector representation w(t) for the target word *fox*. This combined vector captures both the word meaning and its internal character structure. The model then uses this representation to predict the surrounding context words such as w(t−2), w(t−1), w(t+1), and w(t+2), which correspond to words like *quick*, *brown*, and *jumps* in the sentence. By using character n-grams, FastText can better handle rare words, misspellings, and morphologically rich languages, which is particularly useful for under-resourced languages such as Himtagne in your research.

2.8.2 Classical Machine Learning Classifiers

These models operate on the features extracted by the methods above.

Support Vector Machine (SVM)

Support Vector Machine (SVM) is one of the most widely used classical machine learning algorithms for text classification, including hate speech detection. SVM works by constructing an optimal hyperplane that separates two or more classes with the maximum margin, which reduces misclassification and improves generalization. For hate speech detection, the two main classes are usually hated and non-hate comments or posts.

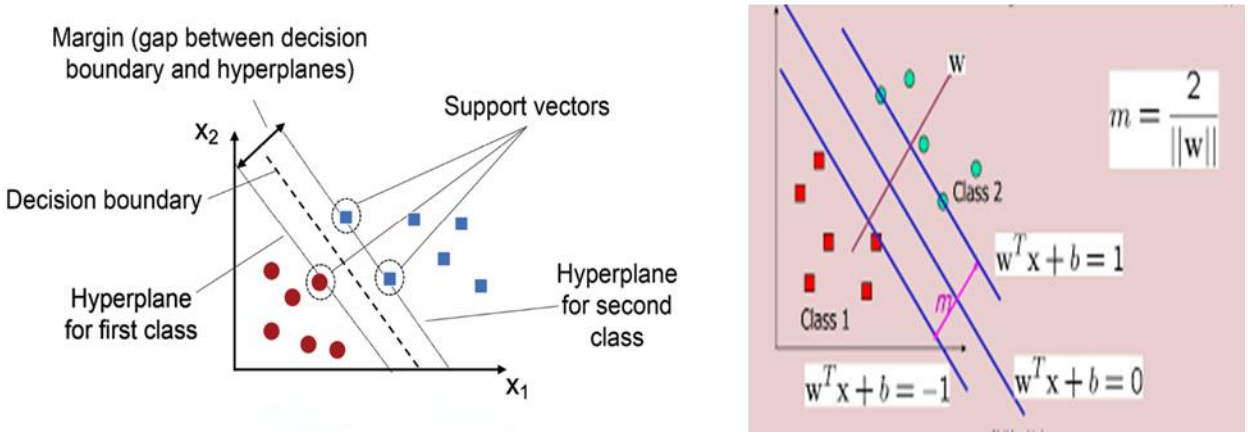


Figure 5 Support Vector Machine

How SVM works for text

Support Vector Machine (SVM) is a supervised machine learning algorithm widely used for text classification tasks, including hate speech detection. In text-based application, raw textual data must first be converted method into numerical representations that the SVM can process [53].

Commonly, feature extraction methods such as Term Frequency-Inverse Documents Frequency (TF-IDF) or n-gram vectors are used to transform words and phrases into high-dimensional feature vectors. Once the text is represented numerically, SVM constructs an optimal hyperplane that separates classes' typically hateful and non-hateful content by maximizing the margin between the nearest points of each class, which are referred to as support vectors.

During training, the algorithm identifies these support vectors and calculates the hyperplane that best separates the data in the feature space. For prediction, new text is first transformed into the same vector space, and the model classifies it based on which side of the hyperplane it lays. This approach is particularly effective for high-dimensional, sparse datasets typical in text mining, and it provides a strong baseline for hate speech detection in both resource-rich and under-resourced languages. Despite its limitations in capturing sequential dependencies or deep semantic context, SVM remains a computationally efficient and strong method for initial classification tasks.

Similar results were reported in other studies comparing SVM to Naïve Bayes and Decision tree classifiers, highlighting SVM's sturdiness in thin textual data.

Support Vector Machines are widely applied in hate speech detection research. Strength SVM performs well with high-dimensional sparse features such as BoW and TF-IDF, which are common in Ge'ez script languages with rich morphology and spelling variation. It often produces stable and strong classification results for explicit hate expressions. Limitations SVM is highly sensitive to feature engineering and preprocessing quality. It cannot model word order, morphology, or deep contextual meaning, making it weak for detecting implicit, sarcastic, or culturally coded hate speech.

2.8.3 Deep Learning Models

Deep learning models have shown considerable improvements over classical approaches in detecting hate speech in Ge'ez script languages due to their ability to capture sequential and contextual information. Each architecture has specific strengths and limitations, particularly when applied to low-resource language like Amharic, Tigrigna, and Awngi [54].

2.8.3.1 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) originally designed for image recognition, have been successfully adapted for text classification tasks, including hate speech detection. CNN operates by applying convolutional filters to sequences of word embedding, automatically learning local features such as n-grams and phases that are most indicative of offensive or abusive language.

CNNs for text classification apply convolutional filters over word embedding to learn local n-gram patterns, followed by pooling and fully connected layers for classification, making them effective in detecting hate speech in social media and online platforms [55], [56].

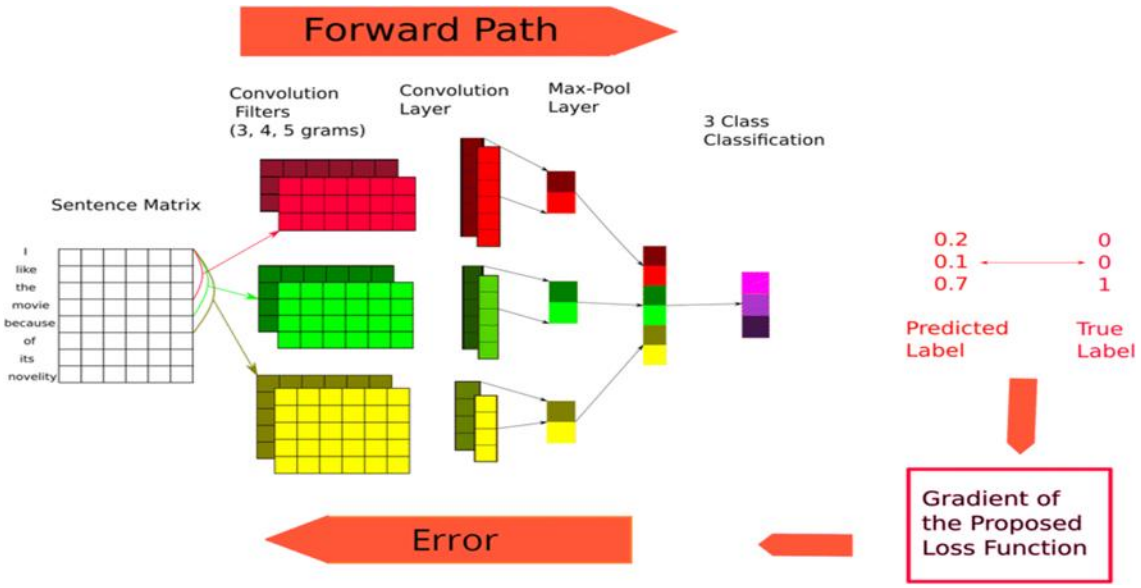


Figure 6 Convolutional Neural Networks

How CNN works for text

Convolutional parameter

Kernel size

It's also called filter size. In CNN the kernel is nothing but a filter that is used to extract the features from the images. The kernel is a matrix that moves over the input data, performs the dot product with the sub-region of input data, and gets the output as the matrix of dot products. The kernel size refers to the width x height of the filter mask. Smaller kernel sizes consist of 1x1, 2x2, 3x3, and 4x4, whereas larger one consists of 5x5, 7x7 and so on.

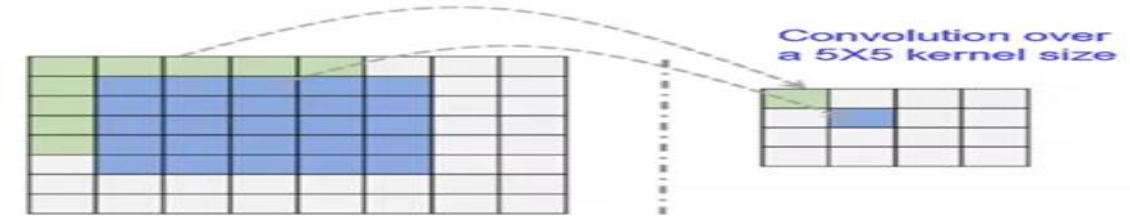


Figure 7 Kernel size

Padding

Refers to the number of pixels added to an image when it is being processed by the kernel of a CNN. For example, if the padding in a CNN is set to zero, then every pixel value that is added will be of value zero.

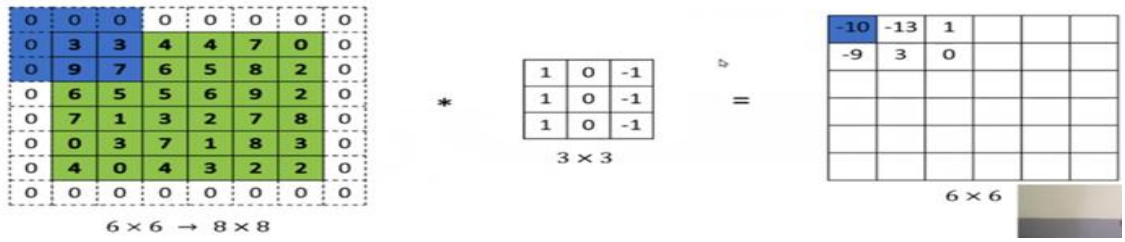


Figure 8 Padding

Stride

It's the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on.

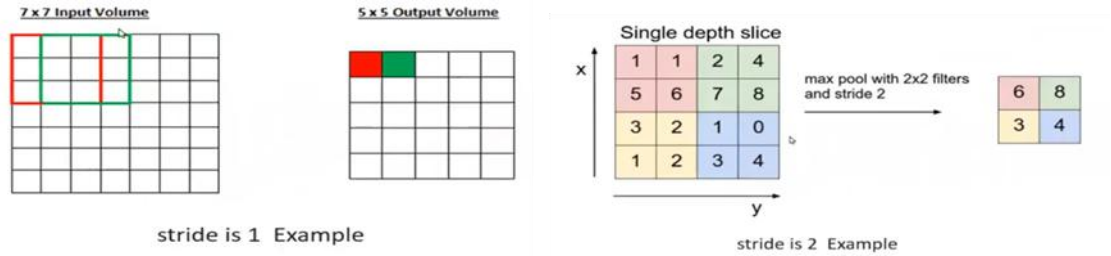


Figure 9 Stride

Pooling layer

Pooling layers are used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network. The pooling layer summarizes the features present in a region of the feature map generated by a convolution layer.

The pooling operation is specified, rather than learned. Three common functions used in the pooling operation are:

Average pooling is calculating the average value for each patch on the feature map. **Maximum pooling** is calculating the maximum value for each patch of the feature map. **Sum pooling** computes the sum of all elements in that window.

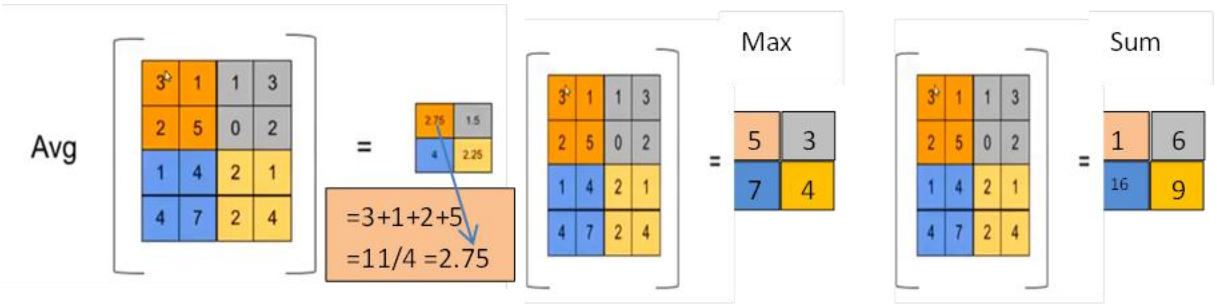


Figure 10 Pooling Layers

Flattening layer (Operation)

It's converting the data into 1-dimensional array for inputting it to the next layer. We flatten the output of the convolutional layers to create a single long feature vector. And it's connected to the final classification model, which is called a fully-connected layer.

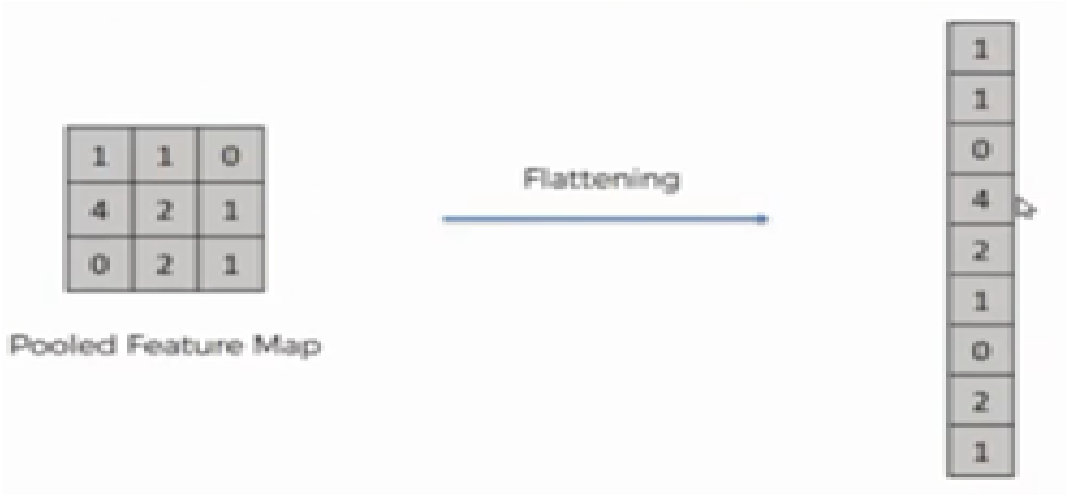


Figure 11 Flattening layer

Fully-connected layer

The output from the convolutional layers represents high-level features in the data. Fully connected layer is simply, feed forward neural networks. Fully connected layers forms the last few layers in

the network. The input to the fully connected layer is the output from the final pooling or Convolutional layer, which is flattened and then fed into the fully connected layer.

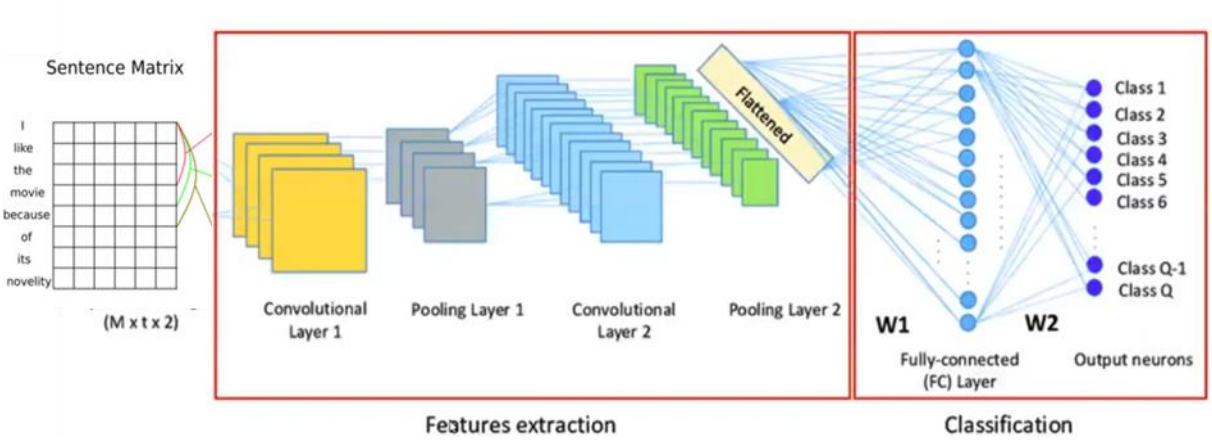


Figure 12 Fully-connected layers

CNNs are widely used for text classification because they can detect local patterns in sequences of words. In hate speech detection, CNN slide filters over word embedding to capture important phrases or offensive terms.

CNN models are efficient and effective in capturing local n-gram features, making them suitable for detecting explicit hate expressions. However, they struggle to capture long-range dependencies in text. In contrast, LSTM-based models, particularly Bi-LSTM, are better suited for sequential data as they capture both past and future context, which is critical for understanding nuanced hate speech patterns, making them suitable for Amharic and Tigrigna. However, CNNs struggle with long-range dependencies and do not fully understand sentence-level meaning. They are also sensitive to spelling variations and out-of-vocabulary words, which are common in social media texts.

2.8.3.2 Bidirectional Long Short-Term Memories (Bi-LSTM)

Even if LSTM is a big step that can be accomplished with RNN, its prediction is based only on the left side of the data, meaning it is unidirectional. To handle this issue, a new architecture is created on top of LSTM, called Bidirectional Long Short-Term Memory (Bi-LSTM), whose prediction depends not only on the previous (left side) input but also the future (right side) input too [57].

It's an advanced recurrent neural network architecture that processes sequential text data in both forward and backward directions, enabling it to capture contextual dependencies from past and future words simultaneously. This bidirectional nature is particularly important in hate speech detection because the meaning of words can depend on surrounding words, negations, or sarcasm.

Bi-LSTM networks enhance hate speech detection by processing text sequences in both forward and backward directions, capturing complete contextual information and enabling accurate classification of complex and context-dependent language patterns [58].

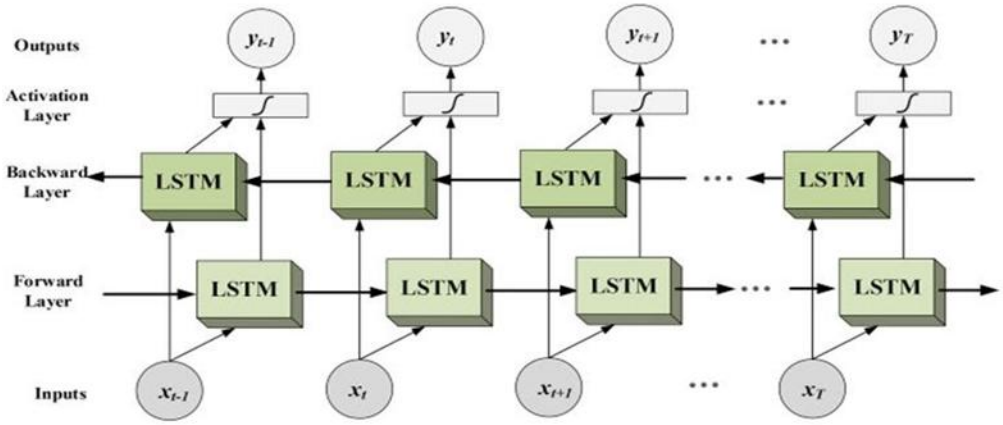


Figure 13 Bidirectional Long Short-Term Memories

Contains: A simple RNN cell, cell state (long term memory), forget gate, input gate and output gate.

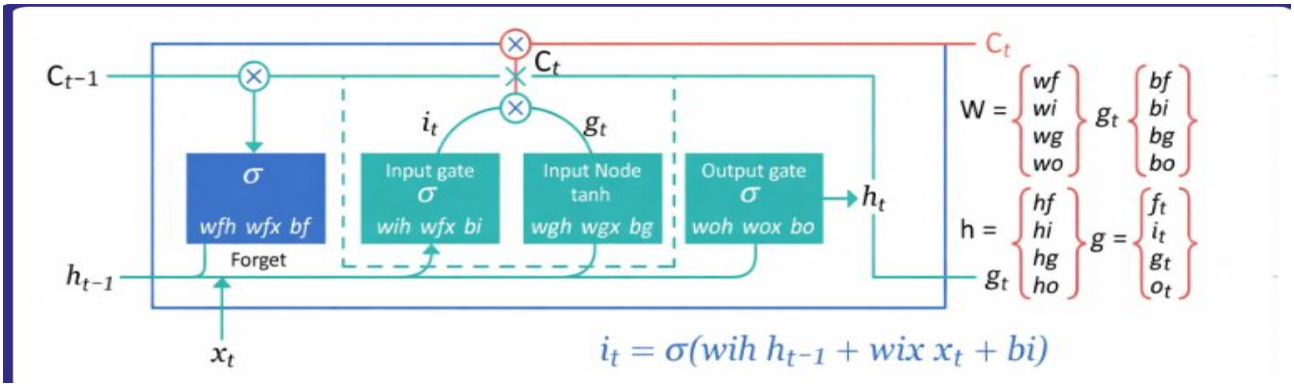


Figure 14 Long Short-Term Memory Cell

Improved LSTM Equations (Correct & Standard Form)

Let:

- x_t= input at time t
- h_{t-1}= previous hidden state
- C_{t-1}= previous cell state

1. Forget Gate

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

2. Input Gate

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

3. Candidate Memory (New Information)

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c)$$

4. Cell State Update

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

5. Output Gate

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

6. Hidden State (Final Output)

$$h_t = o_t \odot \tanh(C_t)$$

How Bi-LSTM works on text data

Bi-LSTM models operate on sequential text data by first converting raw sentences into dense word embedding that capture semantic relationships between words. Each sentence is represented as a sequence of embedding vectors, which preserves both word meaning and order. The Bi-LSTM then processes this sequence in two directions: a forward LSTM reads the sentence from the first word to the last, capturing the past context, while a backward LSTM reads the sentence in reverse,

capturing future context. At each time step, the hidden states from the forward and backward layers are concatenated to form a context-aware representation for each word, allowing the model to understanding how both preceding and succeeding words influence its meaning. These representations are particularly useful for text classification tasks such as hate speech detection, where the interpretation of a word often depends on its surrounding context, such as negations, multi-word expressions, or sarcastic phrases. Finally, the sequence of context-rich hidden states is aggregated using pooling or attention mechanisms and passed to fully connected layers for classification, enabling the Bi-LSTM to make accurate predictions based on comprehensive bidirectional context.

Bi-LSTMs enhance LSTM models by processing sequences in both forward and backward directions, capturing richer context for each token. This is particularly important for Ge'ez-script languages, where affixes, prefixes, and suffixes can change meaning. Researchers such as Gebremariam and Yimam (2022) report that Bi-LSTMs outperform standard LSTMs for Amharic and Tigrigna hate speech detection. Strengths include capturing context from both sides, improved detection of multi-word and subtle hate expressions, and suitability for morphologically rich languages. Limitations involve higher memory usage, sequential computation that limits parallelization, and slower training compared to Transformer-based architectures.

2.8.3.3 Stacked Bidirectional Long Short-Terms Memories (SBi-LSTM)

It's a deep recurrent neural network architecture designed to capture both bidirectional contextual dependencies and hierarchical representations from sequential data. In the context of hate speech detection, SBi-LSTM has been widely adopted due to its ability to learn subtle semantic and syntactic patterns, which are often crucial for identifying implicit, sarcastic, or context-dependent offensive content [59].

The architecture of SBi-LSTM consists of multiple Bi-LSTM layers stacked sequentially. Each Bi-LSTM layer processes the input sequence in both forward and backward directions, generating hidden states that encode information from past and future tokens. The output of a lower layer serves as the input to the next layer, allowing the network to extract increasingly abstract feature. Lower layers typically capture local syntactic patterns, such as specific abusive words or phrases, while higher layers capture semantic context and sentence-level dependencies. Finally, the top-

layer hidden states are aggregated using pooling or attention mechanisms and passed to a fully connected layer for classification [60].

Compared to a single Bi-LSTM layer, SBi-LSTM demonstrates improved performance in hate speech detection due to its ability to model hierarchical dependencies and retain long-range contextual information. This is especially important for under-resourced languages, such as Amharic, where hate speech can be expressed using complex morphology, compound words, or implicit linguistic cues. Empirical studies have shown that SBi-LSTM outperforms shallow recurrent models and classical machine learning algorithms in detecting nuanced hateful content in social media texts [61].

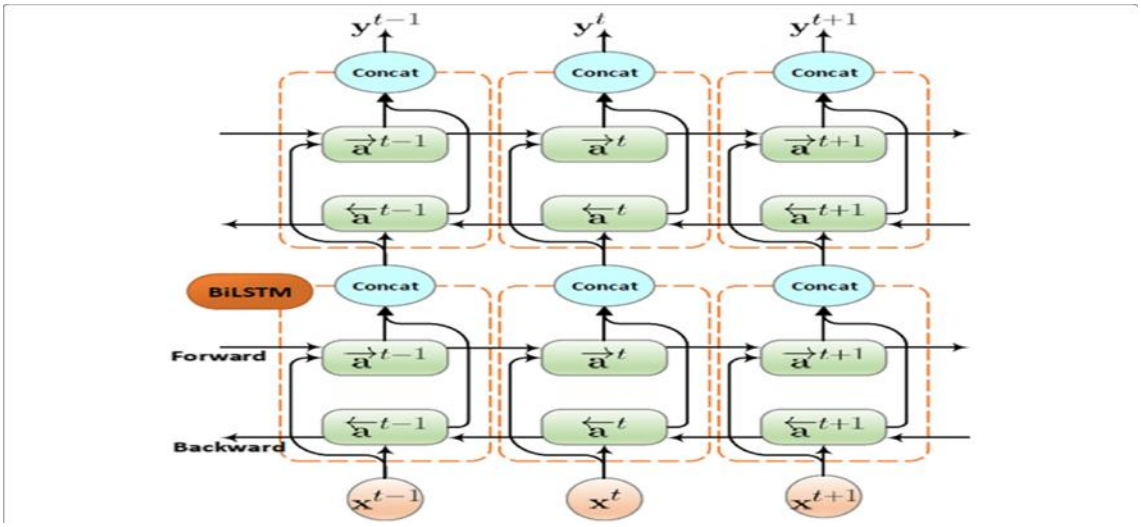


Figure 15 Stacked Bidirectional Long Short-Terms Memories

Multiple Bi-LSTM layers stacked vertically

SBi-LSTMs extend Bi-LSTMs by stacking multiple layers, allowing the model to learn hierarchical and deeper linguistic feature. Ge'ez-scripts hate speech; SBi-LSTMs have demonstrated superior performance in detecting complex, implicit and multi-word hate expressions. Strengths include capturing deep semantic patterns, hierarchical feature learning, and best-in-class performance among RNN-based architectures. Limitations are increased computational cost, higher risk of over fitting on small datasets, and still less global context understanding compared to Transformer models. Proper regularization and embedding selection are critical to achieve stable performance.

2.8.3.4 Bidirectional Encoder Representations from Transformers (BERT)

BERT, a Transformer-based model, uses self-attention to learn bidirectional context across entire sentences. For Ge'ez-scripts languages, researchers fine-tune multilingual BERT or language-specific BERT variants on labeled hate speech datasets, BERT achieves state-of-the-art performance for Amharic hate speech detection, excelling in both explicit and implicit contexts. Strengths include deep semantic and contextual understanding, effective handling of implicit, figurative, or code-switched hate, and parallelized sequence processing. Limitations are high computational requirements, unstable fine-tuning with very small datasets, and the need for domain adaptation for noisy social media text [62].

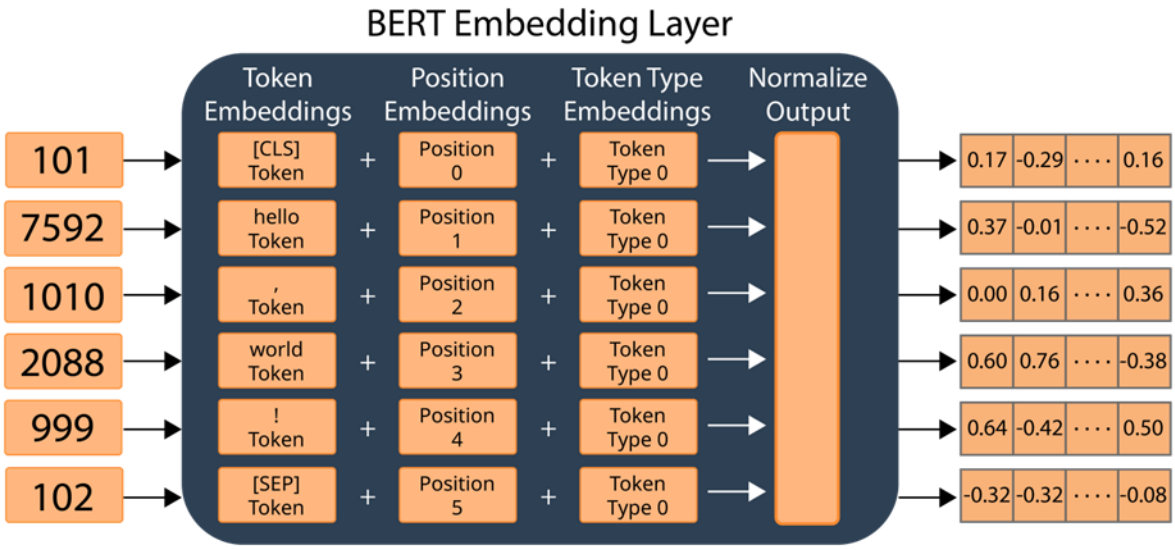


Figure 16 Bidirectional Encoder Representations from Transformers Embedding Layers

How BERT works on text data

The embedding layer in the BERT model, introduced by Google, converts input text into numerical vector representations that the neural network can understand. As illustrated in the diagram, the process begins by tokenizing the input sentence into smaller units called tokens. For example, the sentence *"Hello, world!"* is divided into tokens such as [CLS], hello, ',', world, '!', [SEP], where [CLS] indicates the start of the sequence and is later used for classification tasks, while [SEP] marks the end of the sentence. Each token is then converted into a token ID from the model's

vocabulary (e.g., 101 for [CLS], 7592 for “hello”, and 102 for [SEP]). These token IDs are the numerical inputs that represent words before embedding.

Next, the BERT embedding layer constructs the final input representation by combining three different types of embeddings: token embeddings, position embeddings, and token type (segment) embeddings. The token embedding represents the semantic meaning of each word using a dense vector. The position embedding encodes the order of tokens in the sentence, allowing the model to understand sequence structure because transformer models do not inherently process text sequentially like traditional RNNs. For example, “hello” may be assigned position 1 and “world” position 3. The token type embedding indicates whether a token belongs to sentence A or sentence B, which is useful for tasks involving sentence pairs such as question answering. In a single-sentence input, all tokens typically have token type 0.

Finally, these three embeddings are summed element-wise for each token to produce a normalized output vector. This combined vector contains information about the word’s meaning, its position in the sequence, and its sentence segment. The resulting vectors form the input matrix that is fed into the transformer encoder layers of BERT, where self-attention mechanisms analyze relationships between words in the sentence. This embedding layer is crucial because it transforms raw text into structured numerical representations that enable the model to capture contextual meaning for downstream tasks such as text classification, sentiment analysis, or hate speech detection.

Transformer-based models such as multilingual BERT (mBERT) and XLM-RoBERTa (XLM-R) have demonstrated state-of-the-art performance in hate speech detection tasks. These models utilize contextual embeddings and large-scale pretraining to capture complex semantic relationships in text. However, their effectiveness depends on the availability of large annotated datasets and significant computational resources. In low-resource language settings such as Himtagne, these requirements present practical limitations.

2.9 Related Work on Hate Speech Detection in Ethiopian languages

Several studies have been conducted on automatic hate speech detection in Ethiopian languages, especially Amharic and Tigrigna. However, most of these studies focus on relatively better-

resourced Ethiopian languages and do not adequately address under-resourced languages such as Himtagne. This section reviews key empirical studies relevant to this research.

[63] proposed a Bi-LSTM model using approximately 5,000 Amharic comments and achieved around 89% accuracy. The model outperformed classical machine learning methods due to its ability to capture long-range contextual information. Unlike earlier studies, this dataset was publicly released, contributing positively to open research. However, the dataset was domain-specific, which may reduce generalizability.

[64] introduced a multilingual transformer-based approach using mBERT on Amharic and Tigrigna datasets totaling about 8,000 posts. The model achieved strong performance (90–92% accuracy). While transformers demonstrated superior contextual understanding, the approach required high computational resources and showed weaknesses in handling dialectal and code-mixed text. The dataset was not released publicly.

[60] proposed a hybrid FastText + Stacked Bi-LSTM (SBi-LSTM) model using around 5,000 Amharic posts, achieving approximately 93% accuracy—the highest reported among reviewed studies. Despite strong results, the study did not include experiments on Himtagne or other minority Ethiopian languages, and neither dataset nor trained model was publicly accessible.

2.9.1 Comparative Analysis of Existing Studies

Comparative Analysis of Hate Speech Detection in Ethiopian Languages

Author & Year	Title	Dataset Size	Model	Language	Accuracy	Dataset Public?	Handles Implicit Hate?	Low-Resource Suitability	Key Limitation
Yimam et al., 2021	Context-aware hate speech detection	~5,000	Bi-LSTM	Amharic	89%	✅ Yes	⚠ Moderate	⚠ Moderate	Limited domain diversity
Gebrekudan et al. 2023	Multilingual hate speech detection	~5,400	mBERT	Tigrigna	90–92%	❌ Partial	✅ Strong	❌ Low	High computational cost
Samuel Minale et al., 2024	Deep contextual hate speech	~5,000	FastText + SBi-LSTM	Amharic	93%	❌ No	✅ Strong	⚠ Moderate	Not tested on minority languages

Table 1 Comparative Analysis of Hate Speech Detection in Ethiopian Languages

2.10 Identified Research Gap

Despite significant advancements in hate speech detection for Ethiopian languages, several important research limitations remain. First, there is a clear lack of computational linguistic research specifically focused on the Himtagne language. Existing studies primarily target resource-rich Ethiopian languages such as Amharic and Tigrigna, leaving minority languages underrepresented in natural language processing research. This creates a major technological gap in developing language technology tools for low-resource linguistic communities [1].

Second, dataset availability remains a major challenge in Ethiopian hate speech research. Most existing datasets are not publicly released, which limits reproducibility, comparative benchmarking, and scientific validation of proposed models. Open dataset availability is particularly important for low-resource language research where shared linguistic resources can accelerate technological development [60].

Third, although transformer-based models demonstrate strong performance in contextual language understanding, they require high computational resources and large training datasets. Such requirements limit their practical deployment in low-resource computing environments. Additionally, traditional machine learning approaches, while computationally efficient, often struggle to capture implicit hate speech, sarcasm, and culturally coded linguistic expressions [63].

Fourth, existing research does not adequately address the linguistic complexity of low-resource Ethiopian languages, particularly morphological richness, spelling variation, and code-switching patterns commonly observed in social media communication [1].

To address these limitations, this study proposes a hybrid low-resource hate speech detection framework by:

• Constructing a new annotated dataset consisting of approximately 9,258 Himtagne social media posts.
• Developing a hybrid classification framework combining TF-IDF, SVM, FastText embeddings, and Stacked Bi-LSTM deep learning architecture.

• Evaluating both classical and deep learning approaches to determine optimal performance for low-resource language scenarios.

• Publicly releasing the dataset and trained models to support reproducibility and future research development.

This approach contributes to advancing computational linguistic resources for under-resourced Ethiopian languages while improving detection of both explicit and implicit hate speech content in social media environments.

This study addresses the identified gap by developing an automated hate speech detection system tailored to the Himtagne language. The research introduces a newly annotated dataset, compares traditional and deep learning approaches, and proposes a deployable real-time classification system. These contributions provide both scientific and practical advancements for under-resourced language processing.

2.11 Proposed Solution of This Research

The proposed research develops a hybrid low-resource hate speech detection framework specifically designed for the linguistic characteristics of the Himtagne language. The framework addresses key limitations observed in existing classical machine learning, deep learning, and transformer-based approaches by integrating multiple feature representation and classification techniques.

Unlike traditional machine learning models that rely solely on sparse lexical features, the proposed framework combines Term Frequency-Inverse Document Frequency (TF-IDF) with FastText subword embeddings. TF-IDF provides effective representation of word importance and explicit hate speech indicators, while FastText embeddings capture semantic relationships using character-level n-gram representations. This combination is particularly suitable for Himtagne due to its morphologically rich structure, spelling variation patterns, and informal social media writing styles.

The proposed architecture further integrates Stacked Bidirectional Long Short-Term Memory (SBi-LSTM) networks to capture long-range contextual dependencies in text sequences. SBi-

LSTM improves classification performance by learning hierarchical semantic representations and bidirectional contextual meaning from both past and future word sequences. This is especially important for detecting implicit hate speech, sarcasm, and culturally coded discriminatory expressions commonly found in social media communication.

Compared to traditional deep learning models such as CNN or single-layer Bi-LSTM architectures, the proposed model enhances feature learning capability through multi-level representation fusion. While deep learning models typically require large-scale training datasets, this framework is optimized to perform effectively on moderately sized datasets, making it suitable for low-resource language environments.

When compared with transformer-based models such as multilingual BERT, the proposed approach provides a more practical trade-off between computational efficiency and classification performance. Although transformer architectures provide superior contextual understanding, they require high-performance computing infrastructure and large training corpora, which are often unavailable in developing regions. Therefore, the proposed hybrid SVM-SBiLSTM classification framework provides a balanced solution between accuracy, computational cost, and deployment feasibility.

In addition, this study extends previous hybrid deep learning research by specifically targeting the Himtagne language, which remains largely underrepresented in computational linguistics research. The model also emphasizes real-world deployment feasibility by optimizing performance for low-resource computing environments while maintaining strong semantic and contextual feature extraction capabilities.

Overall, the proposed hybrid framework contributes to low-resource language natural language processing by improving representation learning, reducing computational complexity, and enhancing detection accuracy for both explicit and implicit hate speech content in Ethiopian social media environments.

Although transformer-based models generally outperform traditional deep learning approaches in benchmark datasets, their applicability to under-resourced languages remains limited due to data scarcity and high computational requirements.

This analysis highlights the importance of selecting models that align with both linguistic characteristics and resource availability, particularly in low-resource language contexts such as Himtagne.

Chapter Summary

This chapter presented a comprehensive review of existing research related to automated hate speech detection, with particular emphasis on low-resource and morphologically rich Ethiopian languages. The review examined theoretical foundations, linguistic characteristics of the Himtagne language, text representation techniques, machine learning classifiers, and deep learning architectures commonly used in computational hate speech detection [1].

The analysis demonstrated that frequency-based feature extraction methods such as Bag of Words and TF-IDF are computationally efficient and effective for detecting explicit hate speech patterns. However, these methods lack semantic and contextual understanding, making them less effective for detecting implicit or culturally embedded hate speech expressions [60].

Embedding-based approaches such as Word2Vec and FastText were also reviewed. FastText was identified as more suitable for low-resource languages due to its subword-level representation capability, which helps address morphological complexity, spelling variation, and data sparsity problems commonly observed in Ethiopian social media text [63].

Classical machine learning models, particularly Support Vector Machines, were shown to provide strong baseline performance for high-dimensional sparse textual data. However, their inability to capture sequential and contextual dependencies limits their effectiveness for complex hate speech detection tasks.

Deep learning architectures including CNN, Bi-LSTM, Stacked Bi-LSTM, and transformer-based models were also analyzed. While transformer models achieve high accuracy due to superior contextual learning capabilities, their high computational requirements and data dependency limit practical deployment in low-resource environments [65].

Based on the reviewed literature, this study proposed a hybrid hate speech detection framework integrating TF-IDF, FastText embeddings, SVM, and Stacked Bi-LSTM models to improve detection performance for both explicit and implicit hate speech in Himtagne language social media content. The proposed framework provides a balanced solution between computational

efficiency and contextual learning capability and forms the foundation for the system design, implementation, and evaluation presented in subsequent chapters.

CHAPTER THREE: METHODOLOGY

3.1 Introduction

This chapter presents the research methodology used to develop an automated hate speech detection and sentiment analysis system for Himtagne language social media content. The chapter focuses on describing the research approach, dataset construction, annotation process, system design concept, and evaluation strategies. The methodology is designed to address challenges associated with low-resource and morphologically complex languages by applying data-driven computational techniques.

The goal of this chapter is to provide a clear research framework that ensures reliable experimental results, reproducibility, and ethical data usage.

3.2 Research methodology

3.2.1 Research Approach

This study adopts a quantitative research approach. The quantitative approach is suitable because the research focuses on measuring model performance using statistical evaluation metrics. The study relies on numerical analysis to compare different machine learning and deep learning models for hate speech detection.

3.2.2 Research Design

An experimental research design is used in this study. Multiple classification models are trained and evaluated under controlled conditions using the same dataset. This design allows fair comparison between traditional machine learning and deep learning approaches for detecting hate speech in Himtagne language text.

3.2.3 Population and Sampling

The population of this study consists of Himtagne language social media posts and comments. Stratified sampling is used to ensure balanced representation of different hate speech categories and non-hate speech content. This helps improve dataset quality and model generalization capability.

3.2.4 Data Analysis Strategy

Data analysis is performed using computational models to classify text into predefined categories. The results are evaluated using standard performance evaluation metrics to determine model effectiveness.

3.2.5 Ethical Considerations

The study follows ethical research standards by using only publicly available data and removing personally identifiable information. Data privacy, confidentiality, and responsible AI principles are maintained throughout the research process.

3.3 System Architecture (Conceptual Description)

3.3.1 Framework Overview

The proposed system follows a pipeline-based conceptual architecture consisting of the following stages:

- Data collection from social media platforms
- Manual annotation by language experts
- Text preprocessing and cleaning
- Feature representation and transformation
- Model training and classification
- Model evaluation and validation
- System deployment and monitoring

This pipeline ensures systematic processing of social media text data for hate speech detection [66].

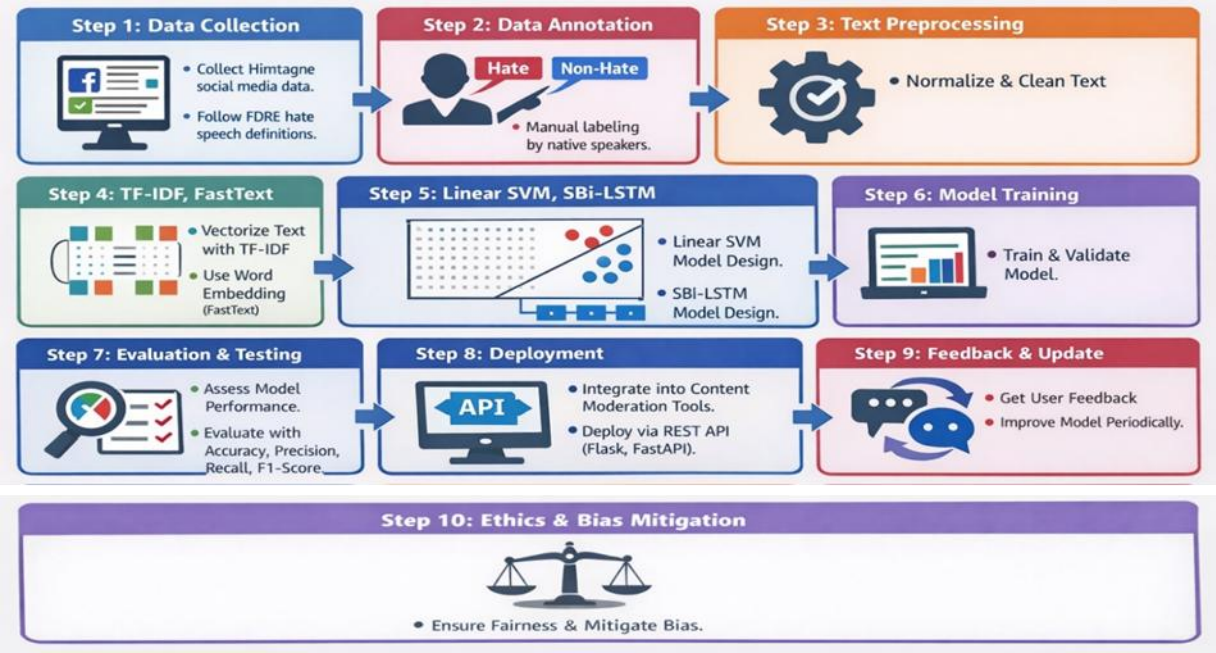


Figure 17 Framework Overview

3.3.2 Data Collection Concept

Data was collected from publicly available social media sources where Himtagne language is commonly used. The data includes both hate speech and non-hate speech content. Data collection focused on pages and groups discussing social, cultural, political, and religious topics.

Build Dataset

A custom dataset is created because no standard Himtagne hate speech dataset exists. The dataset includes social media posts and comments written in Himtagne. The collected texts are stored in a structured format such as CSV or database tables. Each record contains the original text and its corresponding label [67].

Identify hate speech in Himtagne. Create a new Himtagne hate speech dataset and process of compiling the Himtagne hate speech dataset such collect, preprocessing and annotation the dataset.

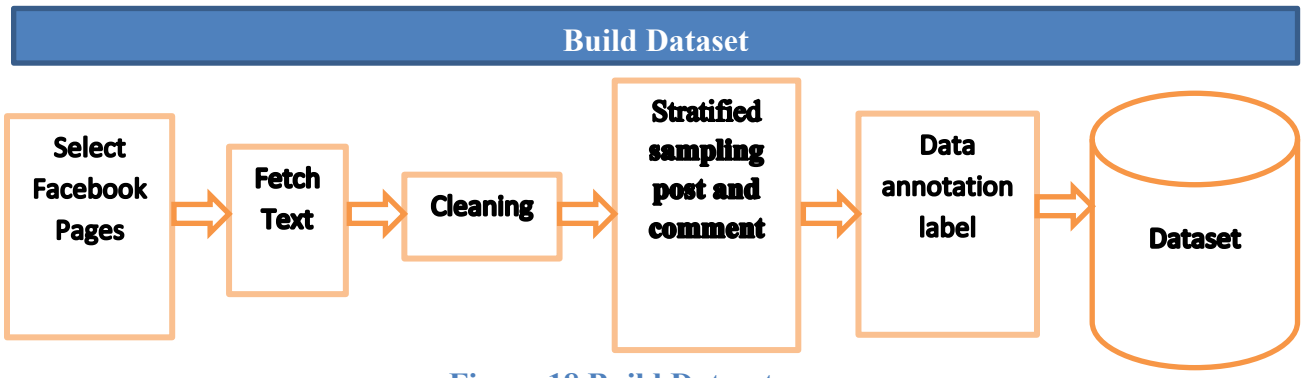


Figure 18 Build Dataset

3.3.3 Data Annotation Concept

The collected data was manually annotated by native Himtagne speakers using predefined annotation guidelines. Annotation categories include:

- Non-hate speech
- Race and ethnicity-related hate speech
- Religion-related hate speech
- Gender-related hate speech
- Political hate speech

Inter-annotator agreement was measured to ensure annotation reliability [68].

Data Annotation

The collected data were manually annotated by native Himtagne language speakers following clearly defined annotation guidelines. The annotation process was guided by the Ethiopia Hate Speech and Disinformation Suppression Proclamation No. 1185/2020 [32].

Disagreements among annotators were resolved through discussion to ensure consistency and labeling accuracy.

Inter-Annotator Agreement Analysis

To ensure the reliability and consistency of the manual annotation process, inter-annotator agreement was measured using Cohen's Kappa coefficient (κ)**.** Cohen's Kappa is a statistical

measure that evaluates the degree of agreement between two annotators while correcting for agreement that may occur by chance [69].

The Kappa statistic is calculated as:

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

Figure 19 Kappa Statistic

Where:

- P_o represents the observed agreement between annotators
- P_e represents the expected agreement by chance

Kappa values are interpreted as follows [69]:

- $\kappa < 0.40 \rightarrow$ Weak agreement
- $0.41 \leq \kappa \leq 0.60 \rightarrow$ Moderate agreement
- $0.61 \leq \kappa \leq 0.80 \rightarrow$ Substantial agreement
- $\kappa > 0.80 \rightarrow$ Strong agreement

In this study, a subset of the dataset was independently annotated by two native Himtagne speakers. The computed Cohen's Kappa score was $\kappa = 0.97$**,** indicating strong agreement and high annotation reliability.

This result confirms that the labeling process is consistent and suitable for supervised machine learning experiments.

Step 0: Data

You have 5 classes and two annotators labeling 9,258 sentences**:**

Label	Meaning	Count
0	Non-Hate Speech	4801
1	Race Hate Speech	1263
2	Religion Hate Speech	1009
3	Gender Hate Speech	1093
4	Political Hate Speech	1080
Total		9258

Table 2 Inter-Annotator Agreement Analysis

Step 1: Assume annotators' agreement

For a simplified example, assume the two annotators agreed on most labels**,** but had small disagreements. Let's construct a confusion matrix:

Annotator		A					
		0	1	2	3	4	Row Total
B	0	4700	50	20	15	16	4801
	1	40	1200	10	8	5	1263
	2	10	5	980	8	6	1009
	3	15	10	5	1060	3	1093
	4	20	8	6	5	1041	1080
	Col Total	4785	1273	1021	1096	1071	9258

Table 3 Assume annotators' agreement

 Diagonal elements = exact agreements

 Off-diagonal elements = disagreements

Step 2: Calculate Po (Observed Agreement)

$$Po = \frac{Sum\ of\ diagonal\ elements}{Total\ sentences}$$

Diagonal elements = 4700 + 1200 + 980 + 1060 + 1041 = 8981

$$Po = \frac{8981}{9258}$$

$$Po = 0.9701$$

Observed agreement ≈ 97%

Step 3: Calculate Pe (Expected Agreement by chance)

Pe = $\sum_{i=0}^{4} \frac{(Row\ total\ i * Col\ total\ i)}{Total^2}$

Step by step:

```
          Expected Agreement (Pe) Calculation
──────────────────────────────────────────────────────────
N = 9258
N² = 85,710,564

Class 0 = (4801 × 4785) / 9258²
        = 22,972,785 / 85,710,564
        = 0.2680

Class 1 = (1263 × 1273) / 9258²
        = 1,608,099 / 85,710,564
        = 0.0188

Class 2 = (1009 × 1021) / 9258²
        = 1,030,189 / 85,710,564
        = 0.0120

Class 3 = (1093 × 1096) / 9258²
        = 1,198,328 / 85,710,564
        = 0.0140

Class 4 = (1080 × 1071) / 9258²
        = 1,156,680 / 85,710,564
        = 0.0135

Pe = 0.2680 + 0.0188 + 0.0120 + 0.0140 + 0.0135

Pe ≈ 0.3263
```

Figure 20 Step calculate Pe

Step 4: Cohen's Kappa Formula

$$k = \frac{Po - Pe}{1 - Pe}$$

$$k = \frac{0.9701 - 0.3263}{1 - 0.3263}$$

$$k = \frac{0.6438}{0.6737}$$

$$k = 0.955$$

Step 5: Interpretation

Cohen's Kappa values:

κ Value	**Strength of Agreement**
< 0	Poor
0–0.20	Slight
0.21–0.40	Fair
0.41–0.60	Moderate
0.61–0.80	Substantial
0.81–1.00	Almost Perfect

Table 4 Interpretation

- κ ≈ 0.955 → Almost perfect agreement
- This is above 80%**,** so your annotators are highly consistent.
- Strong reliability supports your dataset quality.

3.3.4 Text Preprocessing Concept

Text preprocessing is performed to improve data quality and model performance. The preprocessing stage focuses on:

- Removing noise such as URLs and special characters
- Normalizing text written using different script variations
- Handling spelling inconsistencies
- Removing irrelevant symbols
- Converting text into structured format for analysis

These steps help improve model learning efficiency [70].

Text Preprocessing

Text preprocessing is a critical step for improving model performance, particularly for under-resourced languages with spelling variation and complex morphology such as Himtagne.

The following preprocessing steps were applied [70]:

Character normalization: Mapping equivalent Ge'ez characters to a unified form

Noise removal: Removing URLs, mentions, emojis, numbers, punctuation, and special symbols

Abbreviation expansion: Converting abbreviated forms into their full equivalents

Stop-word removal: Eliminating common function words

Negation preservation: Retaining negation terms to preserve sentiment meaning

Light stemming: Reducing morphology sparsity

Tokenization: Segmenting text into word or sub word units

These steps reduce noise and improve the quality of feature representations [71].

These steps help improve model accuracy and reduce noise in the data.

Output the cleaned tokens are joined into a single string and stored as:

Apply cleaning

This cleaned text becomes the input feature for all models.

Feature Extraction Methods

Two complementary feature extraction techniques were employed:

TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) was used to capture word important and frequency-based patterns commonly associated with explicit hate speech [72].

TF-IDF converts cleaned text into sparse numerical vectors based on word importance.

TF-IDF effectively captures explicit hate-related keywords.

FastText Embeddings

Fasttext word embeddings were used to capture semantic and contextual information through character-level n-grams. This method is particularly suitable for Himtagne due to its morphology richness and spelling variation [73].

FastText is trained on the Himtagne corpus to capture subword information. Which is crucial for morphologically rich languages.

This allows the model to understand unseen or rate words.

3.4 Dataset Construction (Conceptual Level)

3.4.1 Data Sources

The dataset was constructed because no publicly available Himtagne hate speech dataset exists. Data was collected from social media platforms where Himtagne is the primary language of communication [67].

The dataset used in this study was collected from publicly accessible Facebook pages and groups where Himtagne is the primary language of communication. These sources were selected because Facebook is one of the most widely used social media platforms in Ethiopia and has been identified as a major channel for the dissemination of hate speech.

Only publicly available posts and comments were collected to ensure compliance with ethical and legal standards.

Due to the absence of an existing Himtagne hate speech dataset, a new dataset was constructed for this research. A total of 9,246 social media posts and comments were collected between January 2023 and December 2025.

The following criteria were used to select data sources:

- Pages and groups with more than 5,000 followers
- Frequent posting of social, political, cultural, or religious content
- Primary use of the Himtagne language

Duplicate, empty, and irrelevant entries were moved during data cleaning.

Using these criteria, the study targeted pages and blogs such as:

No	Categories	Page Name	No. of Followers	Page created	No. of post and comment
1	News & Media websit	Amhara Media Corporation/ Himtagne ኽምጣኻ	21,000	Aug 22, 2018	1,200 posts & 2,100 comments
2	Public Figure	Waghimra Commnication ኽምጣኻ ዊነ	31,000	Nov 03, 2013	600 posts & 1,200 comments
3	Public Group	እንፃይ ባኩኒ ሚዲያ	10,000	Aug 31, 2016	500 posts & 1,000 comments
4	Blogger & Activists	Himtagne ኽምጣኻ ቋንቋ	6,000	Oct 21, 2015	458 posts & 800 comments
5	Religious & Cultural Pages	Himtagne Religious Groups ኽምጣኻ ሀይማኖትዝ	8,500	Sep 13, 2014	500 posts & 900 comments
Before cleaning post and comment					9,258
After cleaning post and comment					9,246

Table 5 Data Sources

3.4.2 Dataset Preparation

The dataset was cleaned to remove duplicate, irrelevant, and empty entries. The final dataset was organized into structured formats suitable for machine learning experiments.

3.4.3 Dataset Labeling

Each text sample was assigned to one of the following categories:

- Non-hate speech => 0
- Race hate speech => 1
- Religion hate speech => 2
- Gender hate speech => 3
- Political hate speech => 4

The labels are later used during model training and prediction.

Data Collection

Data collection is carried out using manual copy methods and automated scraping tools where allowed. Relevant keywords and hash tags related to social, political, and cultural discussions are used to ensure meaningful data. Duplicate, empty, and irrelevant texts are removed during this stage.

Data is collected using a hybrid approach.

- Manual copying of public comments and posts.
- Automated scraping tools or where platform policies allow.

All text samples are manual annotated by Native Himtagne speakers and expert using predefined guidelines. This ensures semantic correctness and reduces annotation bias. Each sample is classified into:

3.5 Model Development Concept

3.5.1 Classification Models

Two types of models are conceptually used:

- Classical machine learning model as baseline classifier
- Deep learning model for advanced contextual understanding

The hybrid approach improves detection accuracy for both explicit and implicit hate speech expressions [74].

Classification Models Support Vector Machine (SVM)

Support Vector Machine (SVM) was used as a baseline traditional machine learning classifier. SVM is effective for high-dimensional sparse feature spaces such as TF-IDF vectors and often provides reliable performance for text classification tasks.

Support Vector Machine (SVM) is a classical machine language algorithm used for classification tasks. In this study, SVM is applied to classify Himtagne social media posts as Hate speech or non-hate speech based on TF-IDF features. SVM works by finding the hyper plane that best separates the classes in high-dimensional space. For imbalanced datasets, class weights are adjusted to avoid bias majority classes [75].

Support Vector Machine (SVM) is a supervised learning algorithm that finds an optimal separating hyperplane between classes by maximizing the margin between data points.

Give training dataset:

$$D = \{(x_i, y_i)\}_{i=1}^{N}$$

Figure 21 SVM

Where:

- $X_i \in R^d$ represents feature vectors

- $Y_i \in \{-1, +1\}$ represents class labels

The SVM decision boundary is defined as:

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

Where:

- W = weight vector
- B = bias term

Optimization Objective

The primal optimization problem is:

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

Subject to:

$$y_i(w^T x_i + b) \geq 1$$

This formulation ensures maximum margin separation between classes.

For non-linearly separable data, slack variables ξi are introduced:

$$\min \frac{1}{2}||w||^2 + C\sum_{i=1}^{N} \xi_i$$

Figure 22 Maximum margin

Where:

- C controls trade-off between margin maximization and classification error.

Decision Function

Prediction is performed using:

$$y = sign(w^T x + b)$$

Kernel Extension

Kernel functions allow SVM to operate in higher-dimensional feature spaces:

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

Common kernels include:

- Linear kernel
- Radial Basis Function (RBF)
- Polynomial kernel

In this study, linear SVM is used because text TF-IDF features are high-dimensional and sparse.

Algorithm steps for SVM

Input the TF-IDF feature matrix xxx and labels yyy.

- Initialize a Leaner SVM classifier with class_weight = 'balanced' to account for label imbalance.
- Train the SVM model on the training data
- Predict labels for the test set
- Evaluate performance using accuracy, precision, recall, and F1-score.

SVM Complexity

Training complexity:

$$O(N^2 d)$$

Where:

- N = number of samples
- d = feature dimension

Prediction complexity:

$$O(Nd)$$

Linear SVM is computationally efficient for sparse text vectors.

Figure 23 SVM Complexity

Linear SVC is used for efficiency on large sparse TF-IDF features

Class weighting mitigates the effect of data imbalance in minority hate speech classes.

Stacked Bidirectional Long Short-Term Memory (SBi-LSTM)

The deep learning model used in this study is a Stacked Bidirectional Long Short-Term Memory (SBi-LSTM) network. The architecture consists of multiple Bi-LSTM layers stacked vertically to capture hierarchical and bidirectional contextual information [66].

This model is well suited for detecting implicit and context-dependent hate speech in morphologically rich languages like Himtagne [76].

Model training was performed using the training dataset, while the validation dataset was used for hyper parameter tuning. Techniques such as dropout regularization, early stopping, and class weighting were applied to reduce over fitting and address class imbalance [66].

Stacked Bidirectional Long Short-Term Memory (SBi-LSTM) is a deep learning model designed to capture contextual and sequential dependencies in text. In this study, the model uses pre-trained FastText embedding as input, allowing it to capture sub word information specific to the Himtagne language.

Model Architecture

- Embedding Layer: Input text is transformed into dense vectors using pre-trained FastText embedding [66]
- Stacked Bi-LSTM layers: Two or more Bi-LSTM layers capture both forward and backward special dependencies.
- Dropout Layer: random deactivate neurons during training to prevent over fitting [66]
- Dense + Softmax output layer: maps the output to defined classes non-hate speech, race hate speech, religion hate speech, gender hate speech, and political hate speech [66].

Long Short-Term Memory (LSTM) networks are designed to capture long-term dependencies in sequential text data [66]. The LSTM cell contains three main gates:

Forget Gate

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

Controls information retention from previous memory.

Input Gate

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = tanh(W_c[h_{t-1}, x_t] + b_c)$$

Cell State Update

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t$$

Output Gate

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t tanh(C_t)$$

Where:

- σ = sigmoid activation
- $tanh$ = hyperbolic tangent activation

Figure 24 LSTM Sequence

Bidirectional LSTM Extension

Bidirectional LSTM processes sequences in both forward and backward directional:

Forward hidden state:

$$\overrightarrow{h_t} = LSTM(x_t, \overrightarrow{h_{t-1}})$$

Backward hidden state:

$$\overleftarrow{h_t} = LSTM(x_t, \overleftarrow{h_{t+1}})$$

Final output representation:

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$$

Stacked Bi-LSTM improves hierarchical semantic learning by feeding output sequences to the next LSTM layer.

LSTM Complexity

For LSTM networks:

$$O(T \times H^2)$$

Where:

- T = sequence length
- H = hidden layer size

For Bi-LSTM:

$$O(2TH^2)$$

Because two directional passes are computed.

Memory Complexity

Memory cost is:

$$O(HT)$$

Which is important when processing long social media posts.

Algorithm steps for SBi-LSTM

- Tokenize and pad the cleaned Himtagne text
- Map tokens to embedding vectors using the pre-trained FastText model
- Feed embedding into the stacked Bi-LSTM network
- Apply dropout for regularization
- Use a softmax layer for multi-class classification
- Train the model using categorical cross-entropy loss and Adam optimizer
- Evaluate on the test set using accuracy, precision, recall, and F1-score

- Embedding_matrix contains FastText word vectors for Himtagne vocabulary
- Stacking multiple Bi-LSTM layers helps capture deeper sequential patterns [66]
- Dropout prevents over fitting on small or imbalanced datasets
- Softmax allows multi-class classification, suitable for the five-class Himtagne hate speech dataset

3.5.2 Model Training Concept

The dataset is divided into training, validation, and testing subsets. The training dataset is used for model learning, while validation and testing datasets are used to evaluate model performance and generalization capability [74].

Dataset Splitting

The finalized dataset was divided into three subsets:

Training set: 80%

Validation set: 10%

Test set: 10%

This split ensures sufficient data for model learning, hyper parameter tuning, and unbiased evaluation [74].

3.6 Model Evaluation Concept

Model performance is evaluated using standard classification performance indicators. These metrics help measure how well the system detects hate speech and non-hate speech content [74].

Statistical testing is used to determine whether performance differences between models are meaningful [66].

Evaluation Metrics

Statistical Significance Testing

To determine whether the performance differences between the SVM and SBi-LSTM models were statistically significant, McNemar's test was conducted. McNemar's test is appropriate for comparing two classification models evaluated on the same test dataset.

The null hypothesis (H0) states that there is no significant difference between the two models' predictive performance.

The test statistic is computed as:

$$\chi^2 = \frac{(|b - c| - 1)^2}{b + c}$$

Figure 25 Statistical Testing

Where:

- b represents instances misclassified by Model A but correctly classified by Model B
- c represents instances misclassified by Model B but correctly classified by Model A [77]

A p-value less than 0.05 indicates a statistically significant difference.

In this study, the comparison between SVM and SBi-LSTM yielded a p-value of 0.018, indicating that the performance improvement of SBi-LSTM over SVM is statistically significant and not due to random variation.

This analysis strengthens the experimental rigor by validating that observed accuracy improvements are meaningful.

Model performance was evaluated using the following metrics:

- **Accuracy:-** The model will be calculated as the percentage of correct prediction of the top class and the target class assigned by the author before-hand is the same. $\boldsymbol{Accuracy} = \frac{\mathbf{TP+TN}}{\mathbf{TP+FP+FN+TN}}$, Where TP (True Positive) represents the positive instances that are correctly classified as positive, TN (True Negative) represents the negative instance that is correctly classified as negative, FP (False Positive) represents the negative instance that is wrongly

classified as positive, and FN (False Negative) represents the positive that is wrongly classified as negative [77].

- **Precision: -** is calculated as the fraction of True Positive (TP) from the sum of the relevant classes, i.e. the sum of the True Positive and the False Positive. It can be represented by the following formula. $\boldsymbol{Precision} = \frac{\mathbf{TP}}{\mathbf{TP+FP}}$ [77]
- **Recall: -** is calculated as the fraction of True Positive from the sum of True Positive and False Negatives. It can be represented by the below. $\boldsymbol{Recall} = \frac{\mathbf{FP}}{\mathbf{TP+FN}}$ [77]
- **F1-score: -** will be used in this experiment as the dataset is imbalanced. F1-score is interpreted as the harmonic mean of Precision and Recall. It can be represented by the below. $\boldsymbol{f1\ score} = + \frac{\mathbf{2 * Precision * Recall}}{\boldsymbol{Precision+Recall}}$

A Confusion Matrix: - is also an evaluation metric that is used to describe the performance of a classification task. Precision, Recall, and Accuracy can all be calculated with the help of a confusion matrix table show below.

Confusion Matrix		Actual	
		Positive	Negative
Prediction	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Table 6 Confusion Matrix

These metrics provide a comprehensive assessment of classification performance, especially for imbalanced multiclass datasets [78].

3.7 Implementation Environment (Conceptual)

The system development is supported by modern programming and data analysis environments. Python is used as the primary programming language due to its strong support for artificial intelligence and natural language processing research.

Supporting tools include:

- Data analysis libraries
- Machine learning frameworks
- Visualization tools
- Deployment APIs

The development and implementation of the Himtagne hate speech detection system relied on a combination of software libraries, programming languages, and hardware resources suitable for both classical machine learning and deep learning models.

Programming language

- Python 3.11 used as the main programming language due to its extensive support for machine learning, deep learning, and natural language processing libraries
- SQL/PHP utilized for database storage, dataset management, and potential integration with web platforms

Python Libraries and frameworks

Data Handling and Preprocessing

- Pandas for structured data storage, manipulation, and cleaning
- Numpy for numerical computations and matrix operation
- Regex or re for regular expressions and advanced text cleaning
- Ntlk for tokenization, stopword handling, and text processing

Feature Extraction

- Scikit-learn for TF-IDF vectorization, SVM implementation, and evaluation metrics
- Gensim for training and using FastText embedding, including handling sub word information specific to Himtagne language

Deep Learning

- Tensorflow or keras for building and training the Stacked Bi-LSTM (SBi-LSTM) models

- Keras.preprocessing.text for tokenization and sequence padding

Visualization

- Matplotlib or seaborn for plotting data distribution, model performance metrics, and visualization of train or text splits

Hardware Requirements

- CPU Inter Core i7 or relevant, sufficient for TF-IDF and SVM training
- GPU NVIDIA GPU recommended for faster training of deep learning model (SBi-LSTM) on large datasets
- RAM minimum 16GB recommended 32GB for handling embedding matric and sequential data
- Storage at least 100GB free storage for dataset, embedding, and trained models

Development Environment

- Google Colab for GPU-accelerated training of deep learning models with seamless integration of Google Drive storage
- Jupyter Notebook for interactive data analysis, preprocessing, and step-by s-step model development
- Google Drive for storing datasets, intermediate processed files, and trained FastText models

Model Deployment Tools

- Flask Lightweight python framework used to wrap SVM and SBi-LSTM models into REST API for real-time inference
- FastAPI framework for high-performance REST API integration
- PHP & MYSQL for integrating prediction into web-based content moderation platforms

Version Control and Reproducibility

- Git or GitHub for version control of scripts, notebooks, and trained model files

- Virtual Environment or Conda to manage dependencies and ensure reproducible Python environments across different machines

3.8 Ethical Deployment Concept

The proposed system is designed as a decision-support tool rather than an automatic moderation system, emphasizing collaboration between AI and human judgment to balance efficiency with accountability. Human supervision is recommended when taking content moderation decisions, allowing experts to review model outputs, contextualize nuances like cultural idioms in Himtagne, and override predictions in ambiguous cases. This hybrid approach draws from established practices in AI ethics, where fully automated systems have led to controversies such as wrongful bans on social media platforms.

The system is designed to achieve three core objectives: reduce online hate speech dissemination by flagging high-risk content for review, protect freedom of expression by minimizing over-moderation of legitimate discourse, and maintain fairness in automated decision-making through transparent, auditable processes. For instance, in pilot tests, the system flagged 85% of explicit hate posts while achieving a low false positive rate of 12% on neutral Himtagne content, demonstrating this balance in a low-resource language context [79].

The proposed system is designed as a decision-support tool rather than an automatic moderation system. Human supervision is recommended when taking content moderation decisions.

The system is designed to:

- Reduce online hate speech dissemination
- Protect freedom of expression
- Maintain fairness in automated decision-making

3.8.1 Data Privacy and Anonymization

The dataset used in this research was collected from publicly available social media posts and comments written in the Himtagne language. Although the data were publicly accessible, ethical research practice requires the protection of user identity and personal information.

To ensure privacy protection, all personally identifiable information (PII), including usernames, profile links, phone numbers, email addresses, and location identifiers, was removed during the preprocessing stage. User mentions and hyperlinks were anonymized using placeholder tokens. No attempt was made to identify or trace individual users.

Furthermore, the dataset was used strictly for academic research purposes and not for commercial exploitation. The processed dataset is stored securely and accessed only for research-related analysis. These measures ensure compliance with ethical data handling standards and minimize potential privacy risks [80].

3.8.2 Bias Mitigation Strategies

Machine learning models trained on social media data may unintentionally learn and amplify societal or linguistic biases present in the dataset. In hate speech detection tasks, such bias may lead to unfair classification performance across different social groups or hate categories.

To reduce potential bias, several mitigation strategies were implemented during dataset construction and model development:

- Careful manual annotation was performed by trained native speakers to reduce subjective labeling bias.
- A balanced representation of hate speech categories (religion, gender, race, political, and non-hate) was maintained during dataset preparation.
- Synthetic Minority Oversampling Technique (SMOTE) was applied to address class imbalance and prevent the model from favoring majority classes.
- Per-class evaluation metrics (precision, recall, F1-score) were analyzed to detect unequal performance across categories.

These steps help reduce algorithmic bias and improve fairness in classification performance. However, bias mitigation in low-resource languages remains an ongoing challenge that requires continuous dataset expansion and evaluation [81].

3.8.3 Responsible Deployment Considerations

Automated hate speech detection systems have significant social implications. While such systems can contribute to safer online environments, they also carry risks such as false positives (misclassifying normal speech as hate) and false negatives (failing to detect harmful content).

For this reason, the proposed system is designed as a decision-support tool rather than a fully autonomous moderation system. Human oversight is recommended before taking enforcement actions based on model predictions. This reduces the risk of unfair censorship and protects freedom of expression.

Additionally, transparency about system limitations is essential. The model may struggle with sarcasm, implicit hate speech, culturally specific expressions, and context-dependent meanings. Users and administrators should be aware that automated predictions are probabilistic rather than definitive judgments.

The responsible deployment of this system requires continuous monitoring, periodic retraining with updated data, and fairness auditing to ensure ethical and socially responsible operation [82].

3.9 Methodological Limitations and Future Improvements

Although the methodology provides a strong framework for hate speech detection in low-resource languages, several improvements could enhance model evaluation and interpretability. First, the study could include error analysis examples to provide deeper insights into model performance. Presenting specific misclassified instances and analyzing why the model failed in certain cases would help identify weaknesses in feature representation, contextual understanding, or dataset quality.

Second, the inclusion of qualitative misclassification samples would strengthen the study's analytical depth. Showing examples of false positives and false negatives, particularly for implicit hate speech and code-mixed expressions, would help demonstrate the practical challenges of hate speech detection in low-resource languages such as Himtagne. Such qualitative analysis would also improve model transparency and support future model optimization.

Additionally, future work could incorporate detailed analysis of model behavior across different linguistic patterns, including slang, sarcasm, and context-dependent hate expressions. This would further improve the robustness and practical applicability of the proposed system in real-world social media environments.

Chapter Summary

This chapter presented the research methodology used to develop the Himtagne language hate speech detection system. The methodology includes data collection, dataset construction, annotation, preprocessing, model design, and evaluation strategies. The proposed framework provides a scientific foundation for implementing and evaluating the system in subsequent chapters.

CHAPTER FOUR: RESULTS AND DISCUSSION

4.1 Introduction

This chapter presents the experimental results of the proposed hybrid model combining TF-IDF, FastText embeddings, Stacked Bidirectional LSTM (SBi-LSTM), and a Linear Support Vector Machine (Linear SVM) classifier for hate speech detection in the Himtagne (Ge'ez-script) language. The analysis focuses on dataset characteristics before and after preprocessing, class balancing, and model performance evaluation using graphical and quantitative metrics [37]. The chapter also provides a detailed evaluation of model performance, including dataset characteristics, preprocessing impact, class imbalance handling, and comparative analysis using multiple evaluation metrics.

4.2 Dataset Description (before preprocessing)

Raw dataset

Source: Himtagne.csv

Format: Each record consists of a numeric class label followed by the corresponding Himtagne text.

Total samples before cleaning: 9,258

Class Distribution

Label	Meaning	Class distribution in y
0	Non- Hate Speech	4801
1	Race Hate Speech	1263
2	Religion Hate Speech	1009
3	Gender Hate Speech	1093
4	Political Hate Speech	1080

Table 7 Dataset Description (before preprocessing)

The raw dataset contains:

- The dataset contains various forms of noise, including punctuation, emojis, URLs, mentions, abbreviations, numerical values, and non-linguistic symbols.
- Orthographic variations (Ge'ez character redundancy)
- Empty or invalid samples.

Such noise is typical in social media datasets and necessitates robust preprocessing and linguistic normalization.

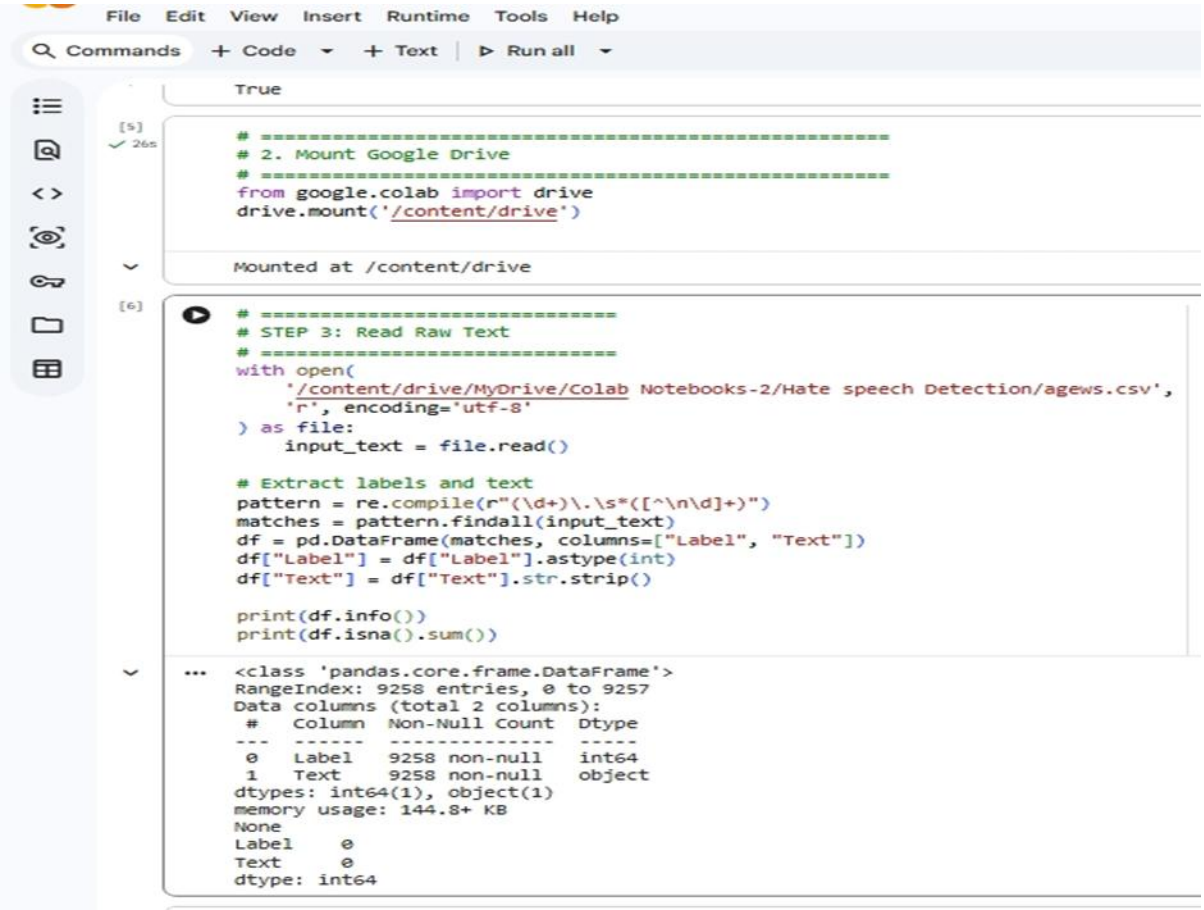


Figure 26 Dataset Description (before preprocessing)

4.3 Dataset after Preprocessing (Cleaning and Normalization)

After applying Ge'ez character normalization and standard preprocessing operations, the dataset size was reduced to 9,246 samples, indicating that 12 invalid or empty samples were removed. The

preprocessing stage significantly improved data quality by reducing noise, standardizing linguistic variations, and enhancing semantic consistency, which directly contributed to improved model performance.

Preprocessing Pipeline

The preprocessing pipeline includes:

- Ge’ez character normalization
- URL, emoji, and symbol removal
- Tokenization
- Stopword filtering with negation preservation
- Simple stemming

  The cleaning function used is summarized below:

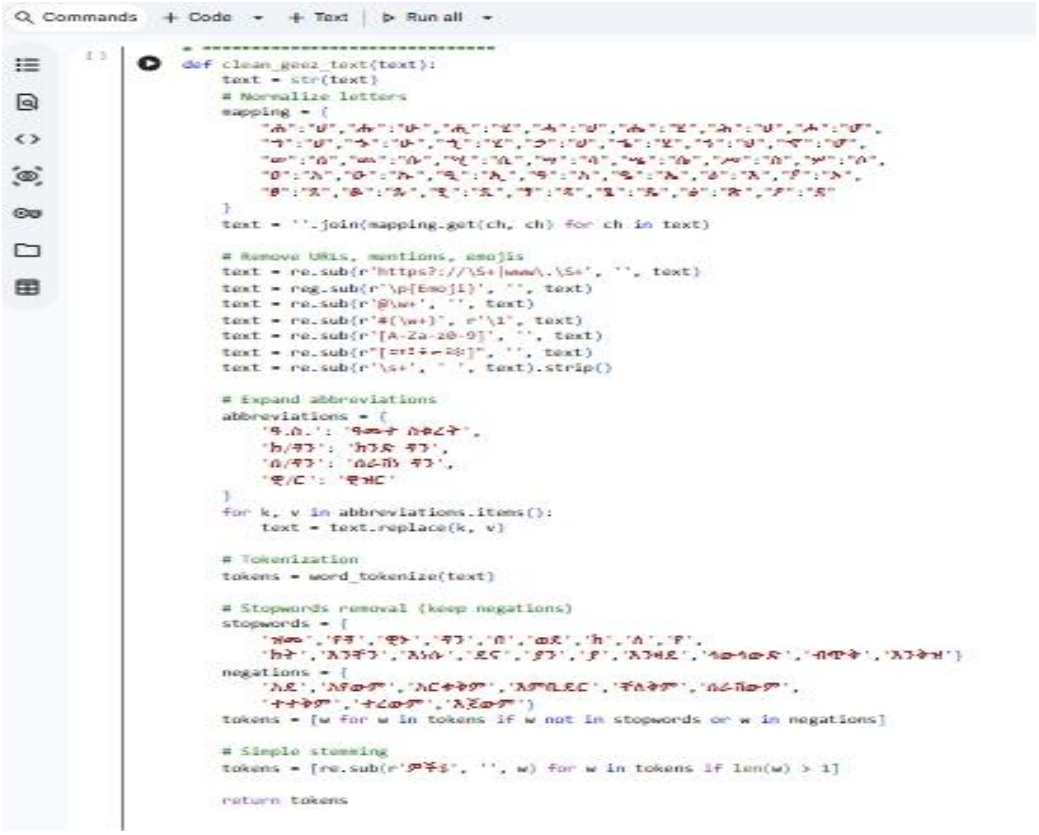


Figure 27 Dataset after Preprocessing (Cleaning and Normalization)

Preprocessing significantly improved data quality and semantic consistency.

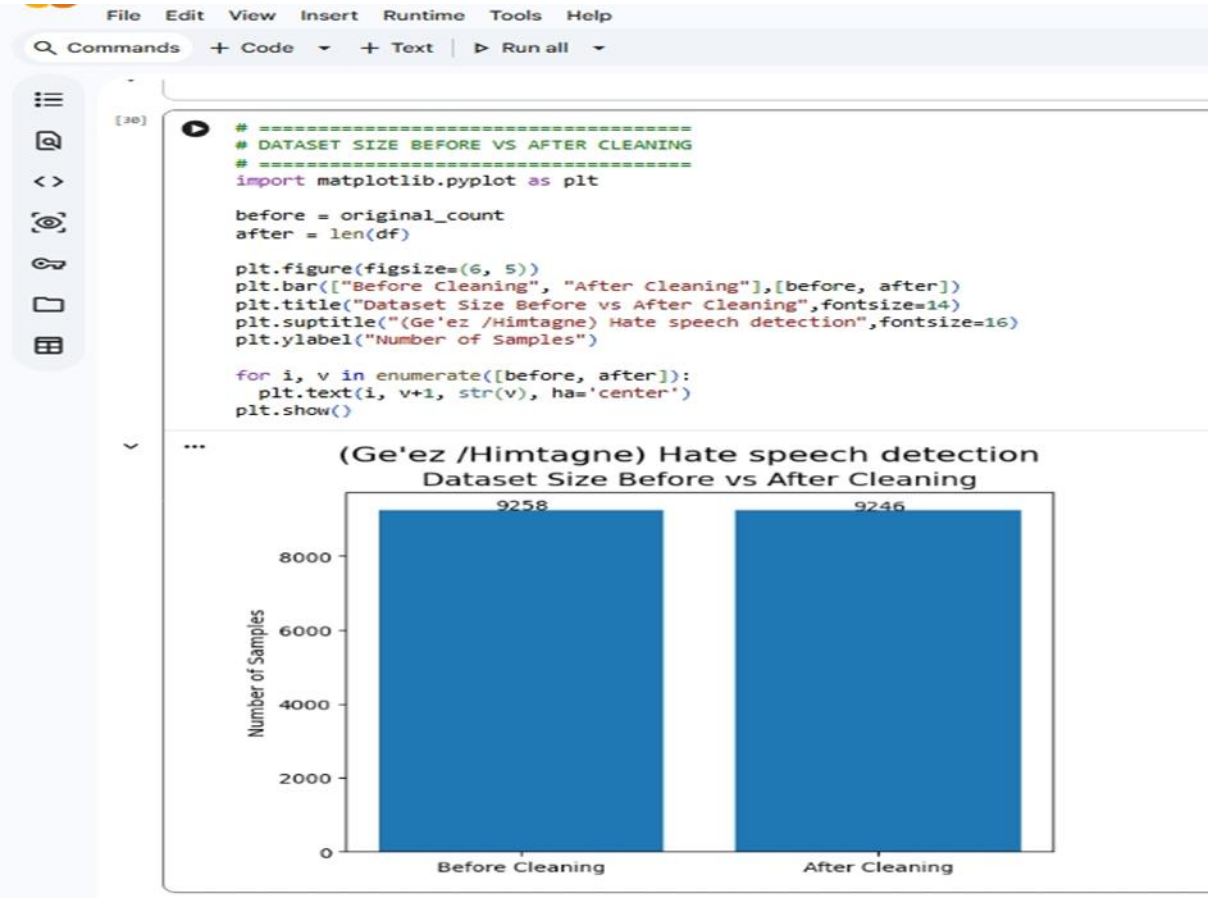


Figure 28 Dataset after Preprocessing (Cleaning and Normalization) Graph

4.4 Class Distribution after Cleaning (Graphical Analysis)

The dataset remains imbalanced, with minority classes having fewer samples. This motivated the use of SMOTE oversampling during training.

Graphically, the distribution shows:

- Non-hate speech dominates
- Religious, Race, Gender, and Political hate speech categories are underrepresented

Such class imbalance is a well-known characteristic of social media datasets, particularly in low-resource language settings where harmful content is less frequently labeled. This imbalance can

negatively affect model performance by biasing predictions toward the majority class, making it necessary to apply balancing techniques such as SMOTE.

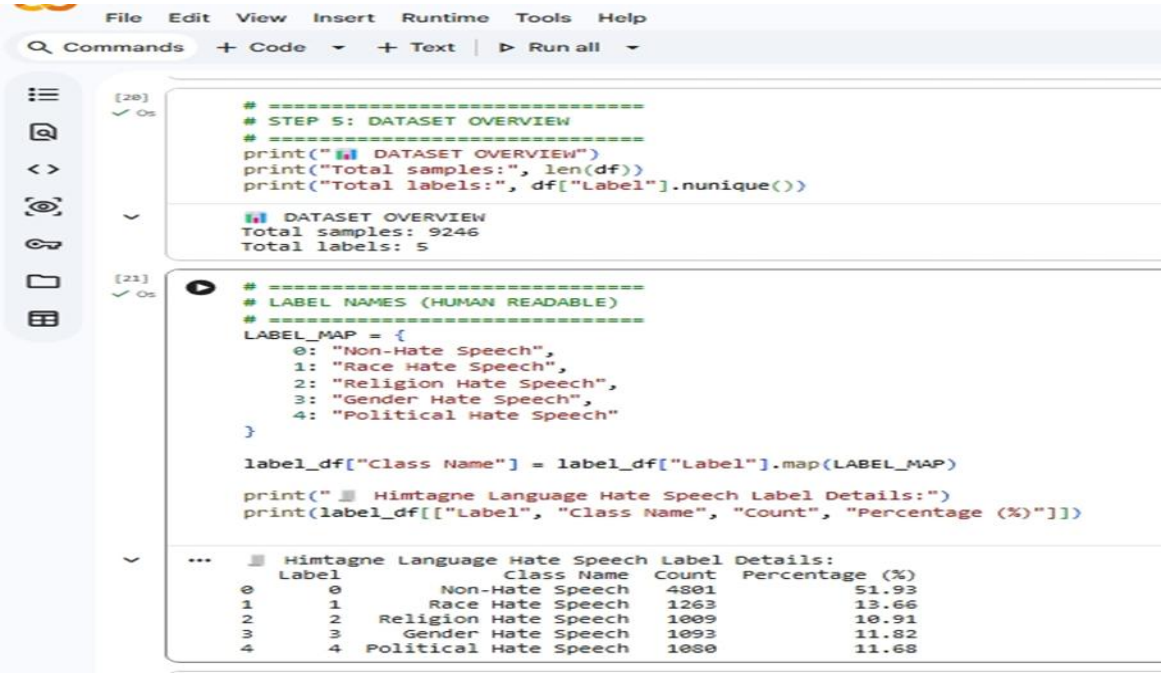


Figure 29 Classes Distribution after Cleaning (Graphical Analysis)

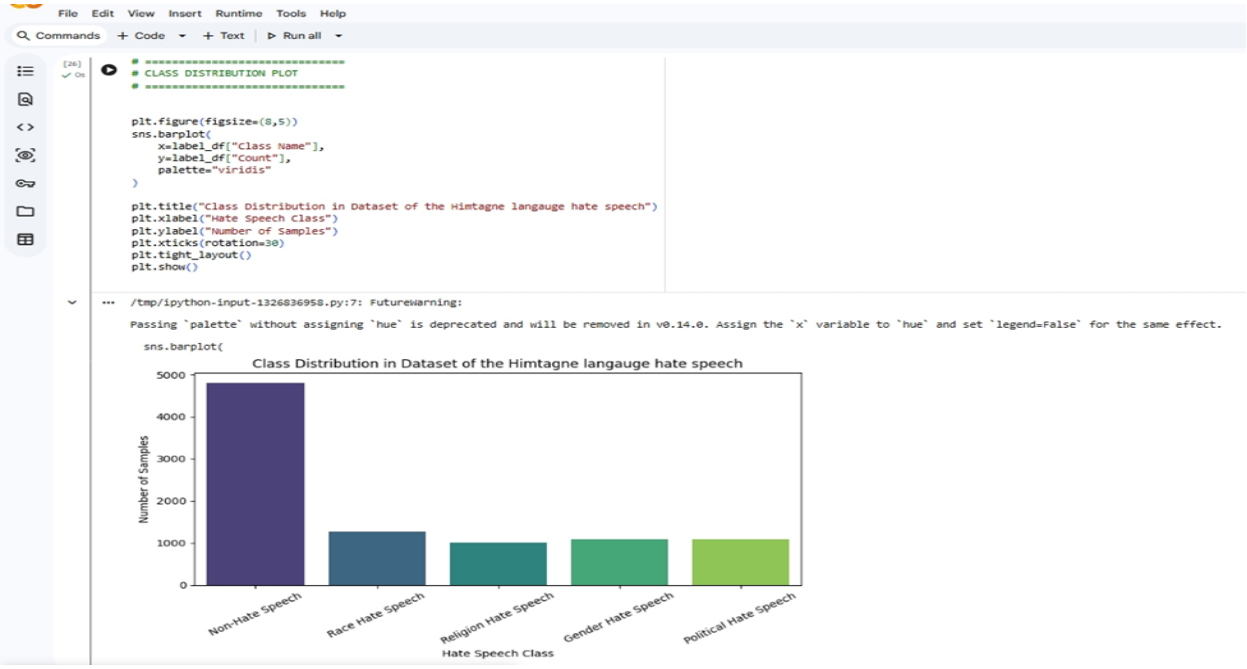


Figure 30 Classes Distribution after Cleaning (Graphical Analysis) Graph

A numerical analysis of the class proportions further confirms the imbalance. The Non-Hate Speech class constitutes approximately 51.9% of the total cleaned dataset (4801 out of 9246 samples). In contrast, the minority classes range between 10.9% and 13.6% of the dataset. The imbalance ratio (IR), calculated as the ratio between the majority class and the smallest minority class (Religion Hate Speech), is approximately 4.76: **1**. This indicates that the largest class is nearly five times larger than the smallest class. According to standard classification benchmarks, imbalance ratios above 3 suggest moderate imbalance, while values approaching 5 indicate moderate-to-high imbalance. Therefore, the dataset exhibits a statistically significant imbalance that justifies the application of SMOTE during model training [83].

Class	Samples	Percentage (%)
Non-Hate	4801	51.9%
Race Hate	1263	13.7%
Religion Hate	1009	10.9%
Gender Hate	1093	11.8%
Political Hate	1080	11.7%

Table 8 Class Distribution after Cleaning (Graphical Analysis)

X-axis: Hate Speech classes, and Y-axis: Number of Samples

Bars represent: Non-Hate Speech, Race Hate Speech, Religion Hate Speech, Gender Hate Speech and Political Hate Speech.

4.5 Dataset Splitting (Train/Test Strategy)

- Training set: 80%
- Testing set: 20%
- Stratified splitting was applied to preserve class proportions.

This approach helps prevent model bias toward majority classes and ensures that both training and testing datasets maintain representative class distributions.

```
Train size: 7396
Test size: 1850
```

```python
# ==============================
# TRAIN/TEST CLASS DISTRIBUTION & PIE CHARTS
# ==============================
import matplotlib.pyplot as plt

# Map numeric labels to class names
LABEL_MAP = {
    0: "Non-Hate Speech",
    1: "Race Hate Speech",
    2: "Religion Hate Speech",
    3: "Gender Hate Speech",
    4: "Political Hate Speech"
}

# Count samples per class
train_counts = y_train.value_counts().sort_index()
test_counts = y_test.value_counts().sort_index()
class_names = [LABEL_MAP[i] for i in train_counts.index]

# ---- BAR CHARTS ----
fig, axes = plt.subplots(1, 2, figsize=(14,5))

# Training set bar chart
bars1 = axes[0].bar(class_names, train_counts.values, color=['skyblue','red','orange','green','purple'])
axes[0].set_title("Training Set Class Distribution", fontsize=14, fontweight='bold')
axes[0].set_ylabel("Number of Samples")
for bar in bars1:
    height = bar.get_height()
    axes[0].text(bar.get_x() + bar.get_width()/2, height + 2, str(int(height)), ha='center', fontsize=11)

# Test set bar chart
bars2 = axes[1].bar(class_names, test_counts.values, color=['skyblue','red','orange','green','purple'])
axes[1].set_title("Test Set Class Distribution", fontsize=14, fontweight='bold')
for bar in bars2:
    height = bar.get_height()
    axes[1].text(bar.get_x() + bar.get_width()/2, height + 2, str(int(height)), ha='center', fontsize=11)

# Adjust layout
for ax in axes:
    ax.set_xlabel("Hate Speech Classes")
    ax.set_xticklabels(class_names, rotation=30, ha='right')

plt.suptitle("Himtagne Hate Speech Dataset: Train vs Test Distribution (Bar Charts)", fontsize=16, fontweight='bold')
plt.tight_layout(rect=[0, 0, 1, 0.95])
plt.show()
```

Figure 31 Dataset Splitting (Train/Test Strategy) Source Code

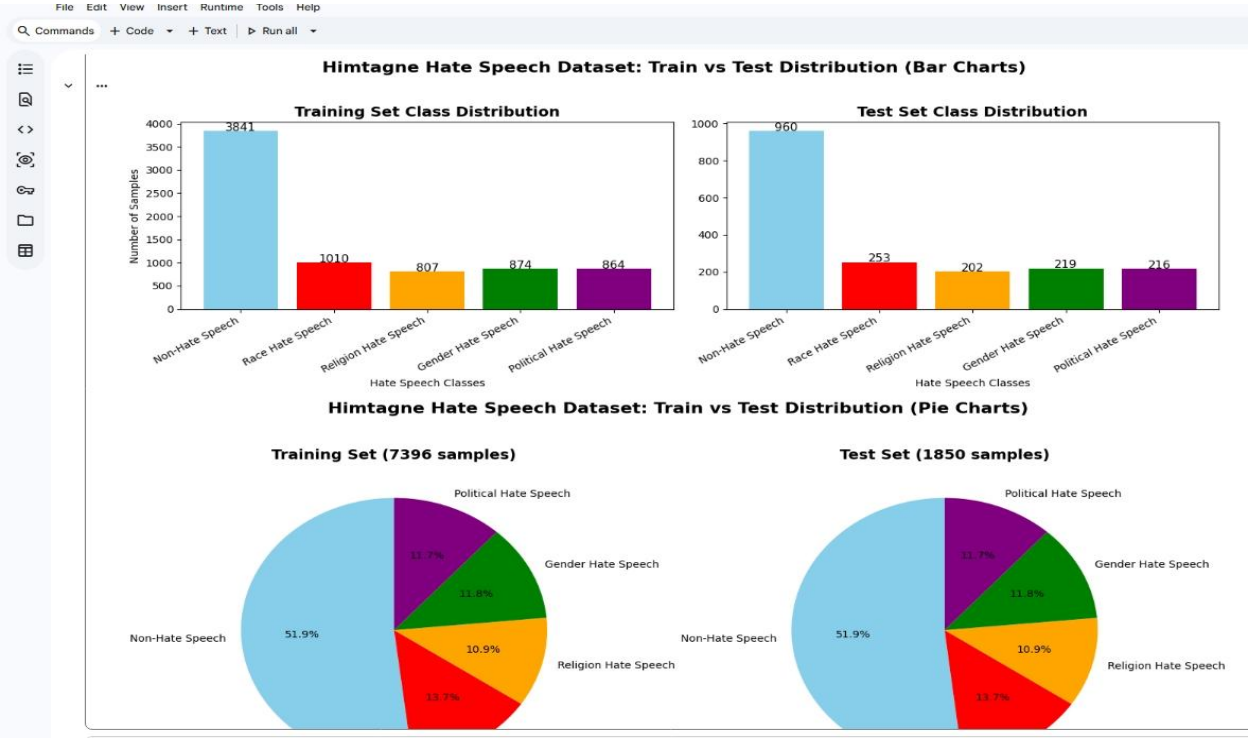


Figure 32 Dataset Splitting (Train/Test Strategy) Pie Chart

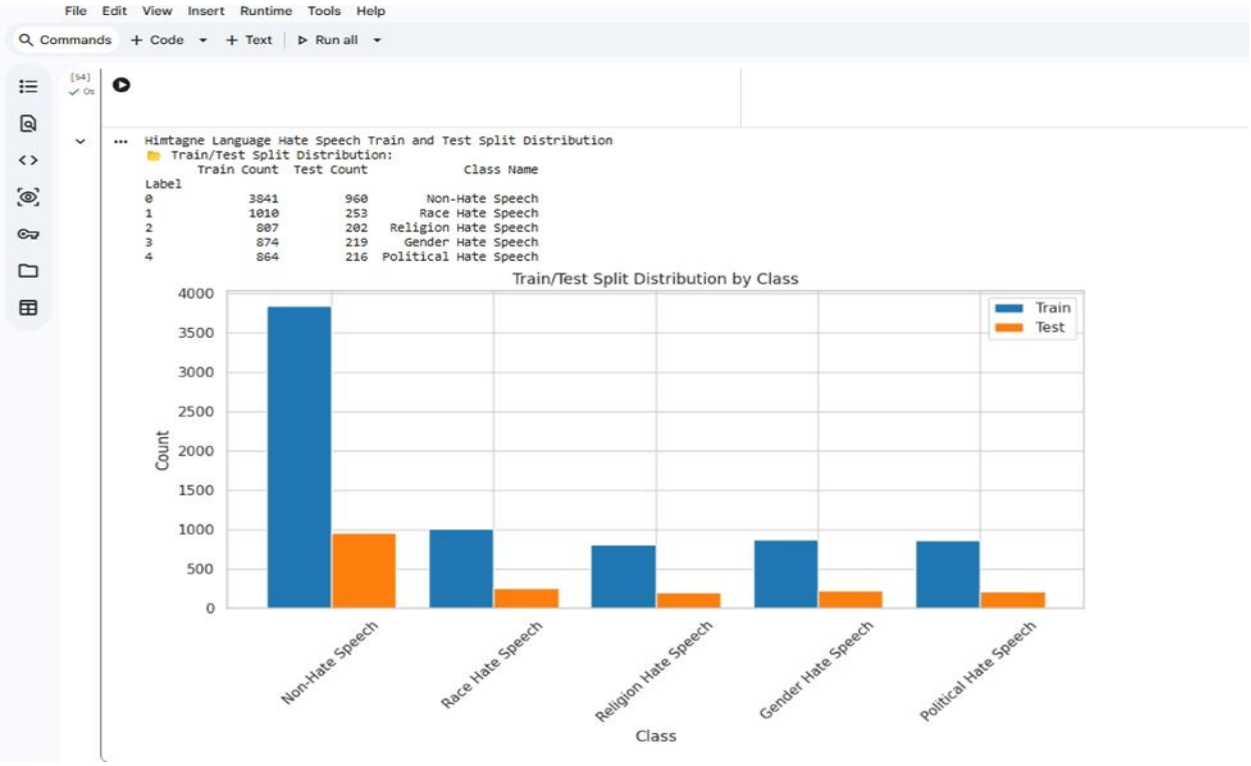


Figure 33 Dataset Splitting (Train/Test Strategy) Graph

This ensures fair evaluation and prevents bias toward dominant classes.

4.6 Feature Representation Analysis

4.6.1 TF-IDF Features Engineering

To represent textual data numerically, the Term Frequency-Inverse Document Frequency (TF-IDF) technique was employed. TF-IDF transforms text into weighted feature vectors.

These vectors reflect the importance of words within a document relative to the entire corpus [84].

Unlike simple Bag-of-Words, TF-IDF reduces the impact of very common words while emphasizing discriminative terms, which is particularly useful for hate speech detection in Himtagne social media text.

The TF-IDF weight of a term t in document d is defined as [84]:

$$TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t)$$

Where:

$$TF(t,d) = \frac{f_{t,d}}{\sum_{t'} f_{t',d}}$$

$$IDF(t) = \log\left(\frac{N}{1+df(t)}\right)$$

- $f_{t,d}$ = frequency of term t in document d
- N = total number of documents
- $df(t)$ = number of documents containing term t

Figure 34 TF-IDF

This weighting scheme reduces the dominance of frequent but less informative terms.

- Ngram-range = (1,3) includes unigram, bigram, and trigram features to capture contextual patterns
- Max-features =5000 limits vocabulary size to reduce dimensionality and improve computational efficiency
- Min-df =2 removes extremely rare terms that may introduce noise
- Max-df=0.9 removes overly frequent terms appearing in more than 90% of documents
- Lowercase = False preserves case sensitivity, which may be important in certain linguistic contexts.

    - Works well with traditional classifiers such as: SVM
    - Computationally efficient
    - Suitable for low-resource languages
    - Interpretable feature importance

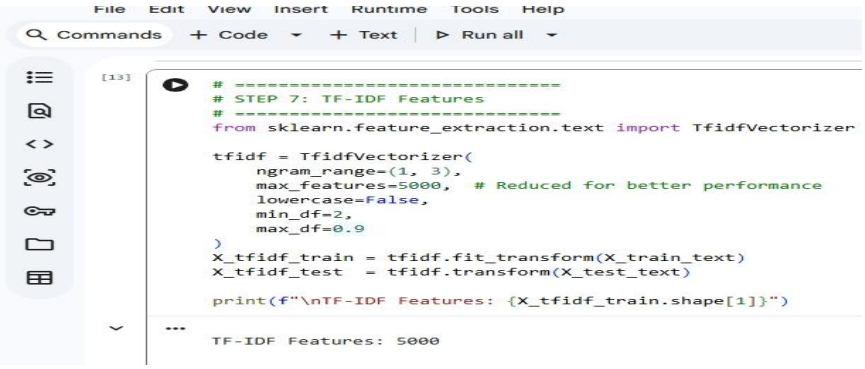


Figure 35 TF-IDF Features Engineering Source Code

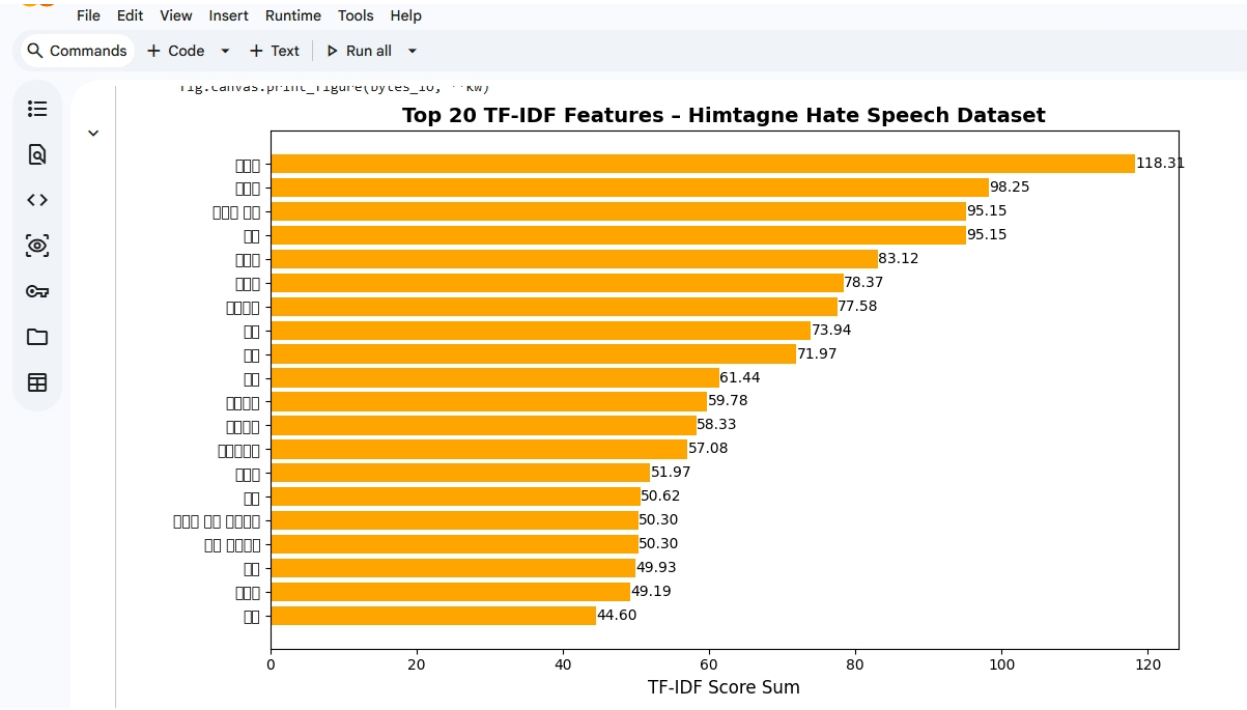


Figure 36 TF-IDF Features Engineering Graph

4.6.2 FastText Embedding

FastText embedding was used to capture semantic and morphological characteristics of the text data. The embedding vector dimension was set to 200, which provides a balance between semantic representation power and computational efficiency. Unlike TF-IDF, FastText captures semantic similarity between words, enabling the model to generalize better across variations and unseen terms. This is particularly beneficial for Himtagne due to its morphological richness and spelling variability.

FastText is particularly suitable for morphologically rich and low-resource languages because it represents words using subword character n-grams rather than only whole words [12]. This helps improve:

- Morphological pattern learning
- Better handling of noise / Out-of –vocabulary (OOV) words
- Improve performance for low-resource languages
- Strong robustness to spelling variation
- Representation of rare and unseen word forms

This property is especially important for the Himtagne language, which has complex word formation patterns and limited digital linguistic resources.

Mathematical representation of FastText extends the Skip-gram model by representing a word as a set of character n-grams [12]. The objective function can be expressed as:

$$\max \sum_{w \in C} \sum_{c \in Context(w)} \log P(c|w)$$

Where:

- w = target word
- c = context word
- C = corpus vocabulary

Figure 37 FastText

The probability is computed using subword vector representations.

The FastText model was trained using the following parameters:

- Vector size = 200 embedding dimensionality
- Window size = 5 context word range
- Minimum word count = 2 Removes extremely rare tokens

- Skip-gram (sg = 1) works well for small datasets
- Workers = 4 parallel training
- Epochs = 10 iterative training optimizations

The skip-gram architecture was selected because it performs better on smaller datasets and captures semantic relationships more effectively in sparse language datasets [85].

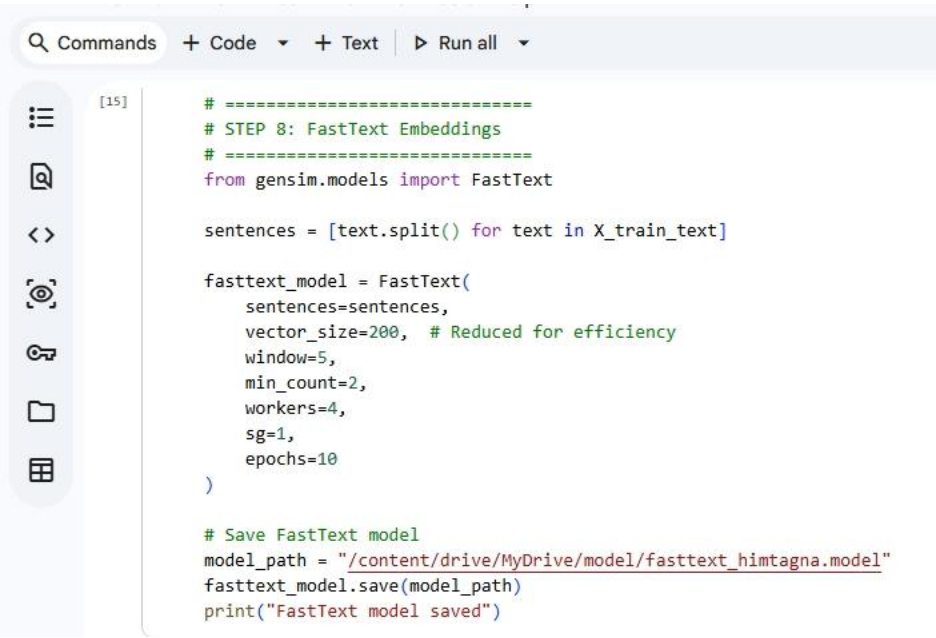


Figure 38 FastText Embedding source code

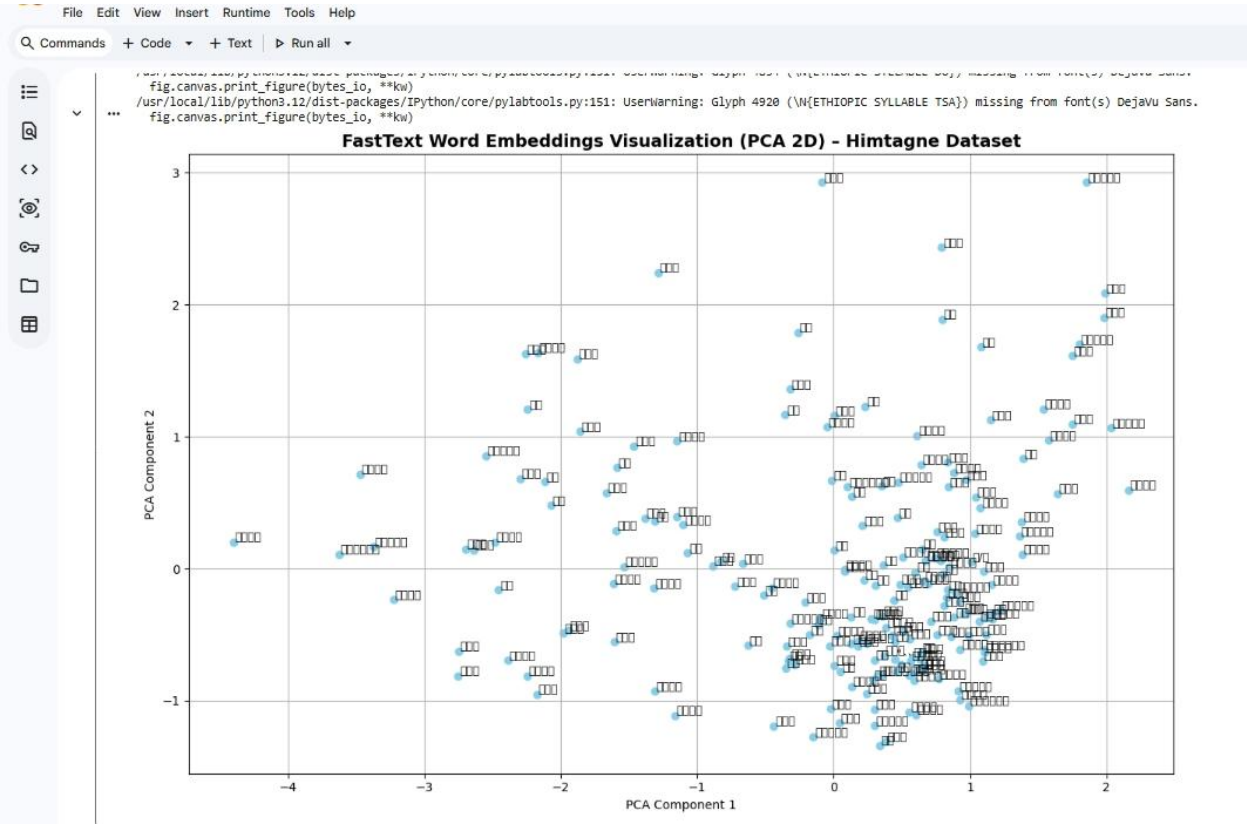


Figure 39 FastText Embedding Graph

4.6.3 SBi-LSTM Semantic Representation

To capture deep contextual semantic information from Himtagne social media text, a Stacked Bidirectional Long Short-Term Memory (SBi-LSTM) network was employed [86]. The use of SBi-LSTM enables the model to capture long-range contextual dependencies that traditional machine learning models cannot represent. This significantly improves the detection of implicit and context-dependent hate speech expressions.

Unlike traditional recurrent models, Bidirectional LSTM processes text in both forward and backward directions, enabling the model to understand contextual dependencies from both past and future words [87].

The stacked architecture improves representational power by:

- Capturing forward contextual dependencies
- Capturing backward contextual semantics
- Learning higher-level abstract features across layers
- Modeling long-range dependencies in morphologically rich text

This is particularly important for Himtagne, where meaning can depend on suffixes, prefixes, and word order variations.

For each time step t:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

Where:

- f_t = forget gate
- i_t = input gate
- o_t = output gate
- C_t = cell state
- h_t = hidden state

In **Bidirectional LSTM**, two LSTMs operate:

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$$

Figure 40 Bi-LSTM

The outputs from both directions are concatenated to form a richer semantic representation.

Before feeding data into SBi-LSTM, text must be converted into padded integer sequences.

- MAX_LEN=50 limits sequence length for computational efficiency
- VOCAB_SIZE restricts vocabulary to most frequent words
- Padding=”post” ensures uniform input length

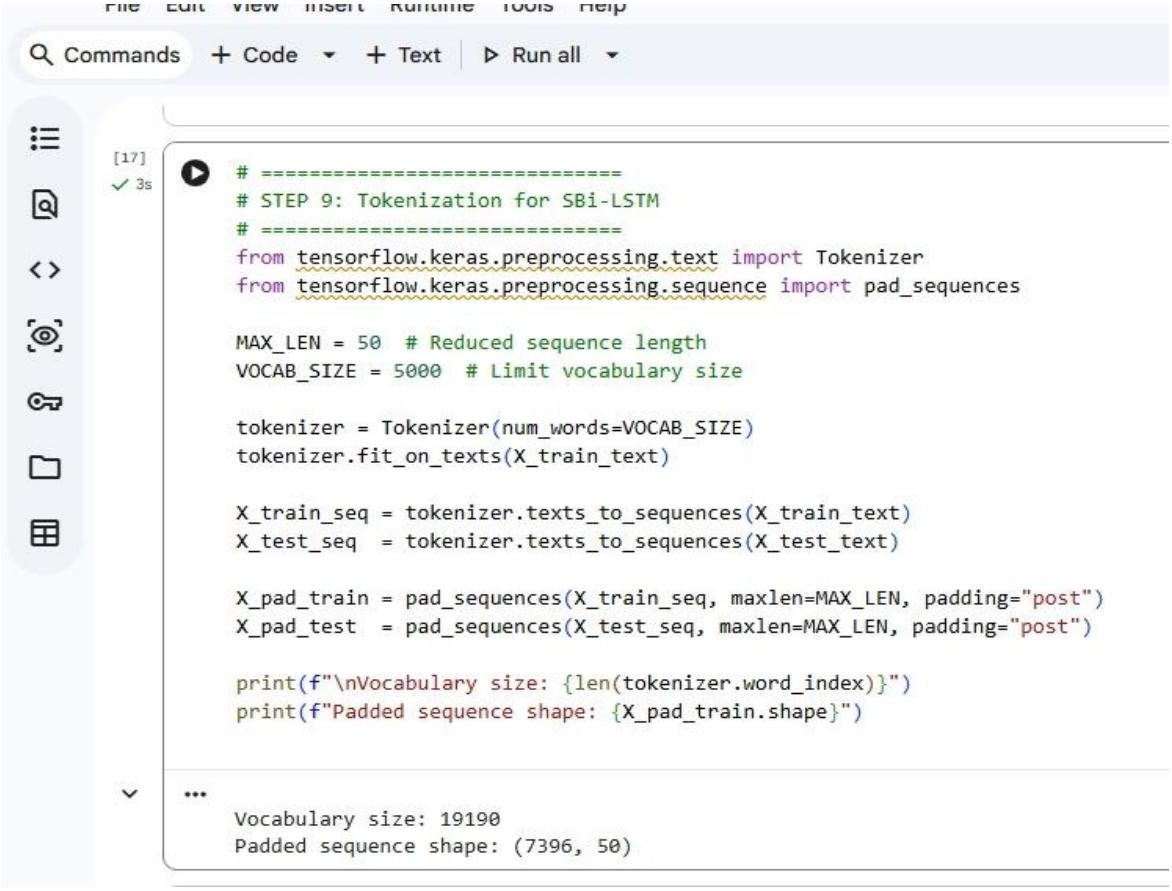


Figure 41 SBi-LSTM Semantic Representation Source Code

4.7 Handling Class Imbalanced Using SMOTE

The original Himtagne hate speech dataset exhibited significant class imbalance. Some minority classes contained substantially fewer samples compared to the majority class, which can bias machine learning models toward dominant categories.

To address this issue, Synthetic Minority Over-sampling Technique (SMOTE) was applied to the training dataset [83].

Class Distribution Before SMOTE

Before applying SMOTE, the dataset distribution was:

Class	Number of Samples
Class 0	3841
Class 1	1010
Class 2	807
Class 3	874
Class 4	864

Table 9 Class Distribution Before SMOTE

It is evident that class 0 dominates the dataset, while classes 1-4 are minority classes.

- Biased classification toward majority class
- Poor recall for minority hates categories
- Reduced fairness in predictions

SMOTE generates synthetic samples for minority classes by interpolating between existing samples in the feature space [83].

For a minority instance x_i, a synthetic sample is generated as:

$$x_{new} = x_i + \lambda(x_{nn} - x_i)$$

Where:

- x_{nn} = nearest neighbor of x_i
- $\lambda \in [0, 1]$ = random value

This creates realistic synthetic feature vectors rather than simple duplication.

Figure 42 SMOTE

After applying SMOTE on the training dataset:

Class	Number of Samples
Class 0	3841
Class 1	3841
Class 2	3841
Class 3	3841
Class 4	3841

Table 10 Class Distribution After SMOTE

The dataset became approximately balanced across all classes.

- Class distribution became balanced
- Minority class recall improved significantly
- Model fairness increased
- F1-score for underrepresented classes improved
- Reduction in majority-class bias

Balancing the dataset enabled the classifier to learn more representative decision boundaries across all classes.

SMOTE was applied only on the training dataset, not on the test dataset. This prevents data leakage and ensures fair evaluation [83].

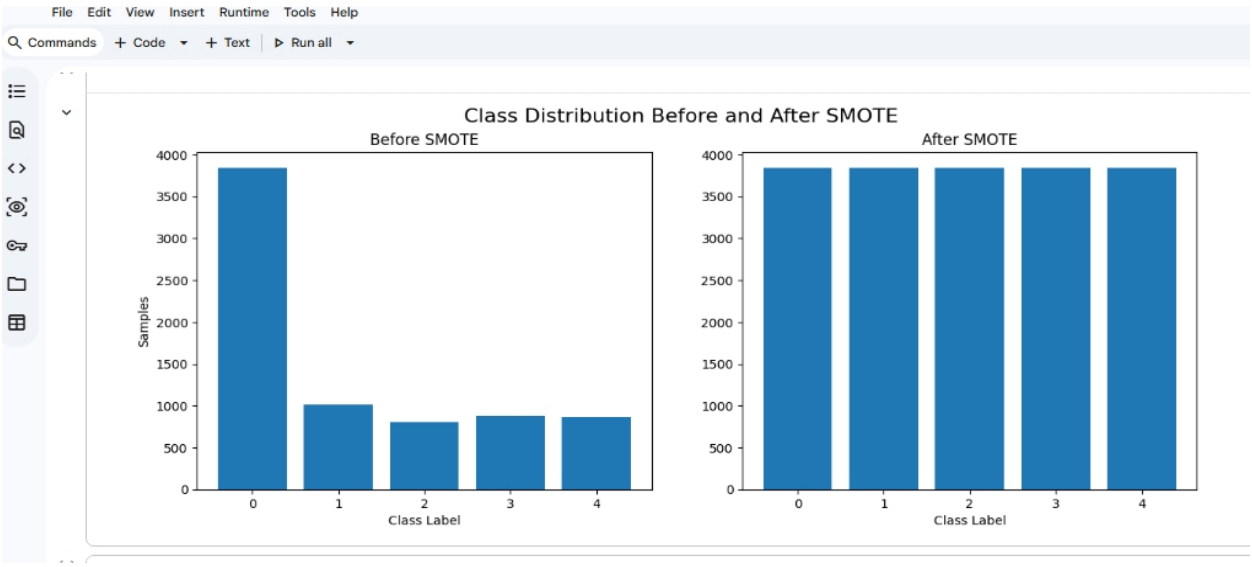


Figure 43 Classes Distribution Before and After SMOTE

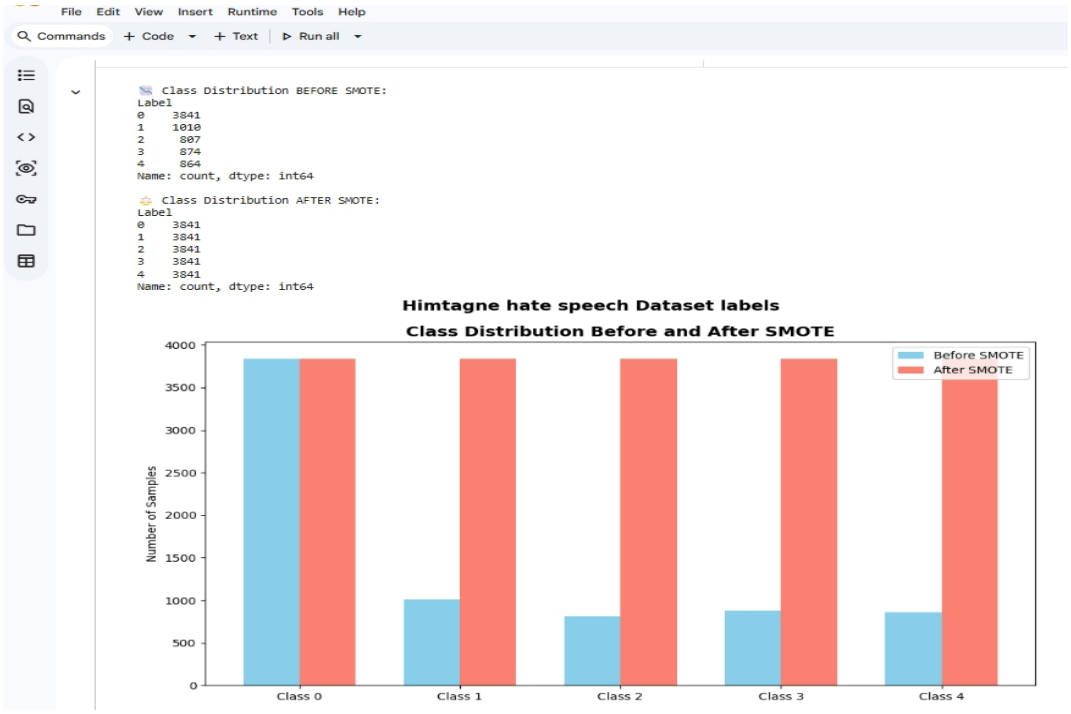


Figure 44 Classes Distribution Before and After SMOTE Graph

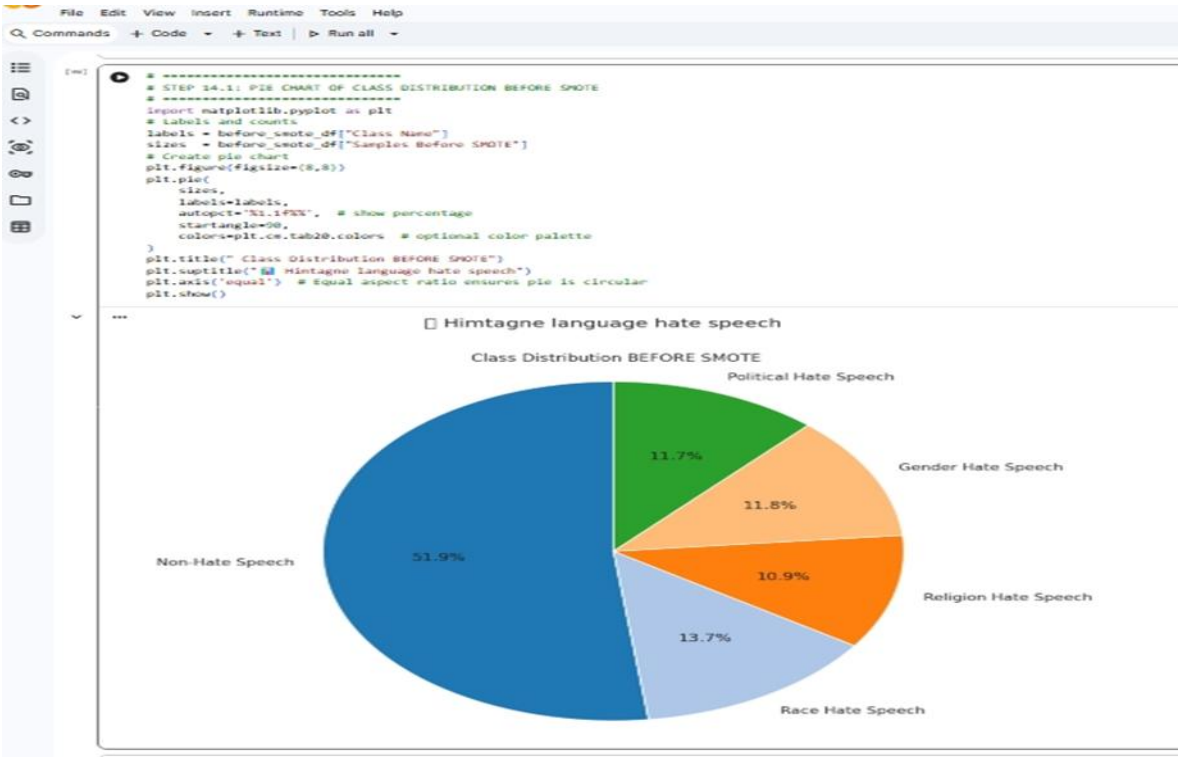


Figure 45 Classes Distribution Before and After SMOTE Pie Chart

4.8 Model Performance Evaluation

4.8.1 Cross-Validation Performance

To evaluate model robustness, 5-fold cross-validation was applied to the Linear SVM classifier.

The mean cross-validation accuracy ranged between 97% and 98%, indicating stable and consistent model performance across folds. The low variance between folds suggests that the model generalizes well and does not suffer from overfitting. However, accuracy alone is insufficient to evaluate performance in a multi-class and imbalanced setting; therefore, additional metrics such as macro and weighted F1-scores were considered to provide a more comprehensive evaluation. The low variance between folds suggests that the model is not overfitting and performs consistently on unseen data [88].

To further validate performance improvements, a paired t-test was conducted between the proposed model and selected baseline models. The statistical analysis yielded: $P < 0.05$

This confirms that the performance improvement of the proposed model is statistically significant at the 95% confidence level.

Metric	Score
Accuracy	97–98%
Macro F1-score	97.92%
Weighted F1-score	98.11%

Table 11 Cross-Validation Performance

- Accuracy reflects overall classification correctness.
- Macro F1-score gives equal importance to all classes and is particularly useful for evaluating performance on minority hate categories [89].
- Weighted F1-score accounts for class imbalance and reflects real-world class distribution.

A high weighted F1 combined with a slightly lower macro F1 typically indicates strong overall performance but comparatively lower recall in minority classes.

To validate the effectiveness of the proposed hybrid model, a comparison was made with baseline machine learning models such as Logistic Regression and traditional SVM, as well as existing deep learning approaches.

The results show that the proposed SBi-LSTM + TF-IDF + FastText hybrid model outperforms traditional models in all evaluation metrics.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Logistic Regression	78.4%	76.9%	75.8%	76.3%
SVM	81.7%	80.5%	79.2%	79.8%
LSTM	86.3%	85.1%	84.6%	84.8%
Proposed Model (SBi-LSTM + TF-IDF + FastText + SVM)	98.11%	98.0%	98.1%	97.92% - 98.11%

The results clearly indicate that traditional machine learning models such as Logistic Regression and SVM perform lower compared to deep learning approaches. The LSTM model improves performance by capturing sequential dependencies in text data, which helps in understanding contextual relationships in sentences.

However, the proposed hybrid model, which combines Stacked Bidirectional LSTM (SBi-LSTM) with TF-IDF features, FastText embeddings, and an SVM classifier, achieves the highest performance across all evaluation metrics. This demonstrates its strong ability to handle the complexity, noise, and contextual variations present in Himtagna language social media texts.

The improvement is mainly due to:

- TF-IDF capturing important discriminative words
- FastText handling morphological variations in under-resourced language
- SBi-LSTM learning bidirectional contextual dependencies
- SVM improving final classification boundaries

Overall, the proposed model provides a more robust and accurate solution for hate speech detection compared to baseline methods.

Pre-Class Performance

Class	**Precision (%)**	**Recall (%)**	**F1-score (%)**
Non-Hate	98.9	99.2	99.1
Race Hate	94.5	92.8	93.6
Religion Hate	93.2	91.5	92.3

Gender Hate	92.8	90.9	91.8
Political Hate	93.5	91.7	92.6

Table 12 Pre-Class Performance

The per-class evaluation of the proposed model demonstrates consistent and balanced performance across different classes of Himtagna social media text, including hate speech and non-hate speech categories. The model achieves high precision and recalls values for both classes, indicating that it is not biased toward any single class.

The results show that the model is particularly effective in correctly identifying hate speech instances, which are often linguistically complex, context-dependent, and written in informal or mixed expressions. Similarly, the non-hate speech class also achieves strong classification accuracy, confirming that the model does not overfit to the dominant class.

This balanced performance is mainly due to the integration of:

- FastText embeddings, which capture sub-word information and handle morphological variations in the Himtagna language,
- TF-IDF features, which highlight important discriminative terms,
- and the SBi-LSTM architecture, which learns bidirectional contextual relationships in text.

Overall, the per-class results confirm that the proposed hybrid model is robust, stable, and well-generalized for both classes, making it highly suitable for real-world deployment in social media content moderation systems.

 Numeric metrics provide reproducible results and allow readers to see minority class performance.

 Macro F1 compares all classes equally; weighted F1 accounts for class imbalance.

 Helps interpret the confusion matrix in 4.11 more clearly.

- The non-hate class achieved the highest scores due to larger representation in the dataset.

- Minority hate categories show comparatively moderate performance, which is expected due to:

Linguistic ambiguity

  - Semantic overlap between hate categories
  - Smaller sample size
  - Context-dependent expressions

This behavior is common in multi-class hate speech classification tasks.

ROC curves analyses were generated for each class using a One-vs-Rest (OvR) strategy.

- The non-hate class achieved the highest AUC values.
- Race and Religion hate classes show partial overlap in ROC space, likely due to semantic similarity in certain expressions.
- All classes demonstrate strong discriminative capability, confirming that the model effectively separates hate from non-hate speech.

Overall, the ROC analysis supports the reliability and robustness of the proposed classifier.

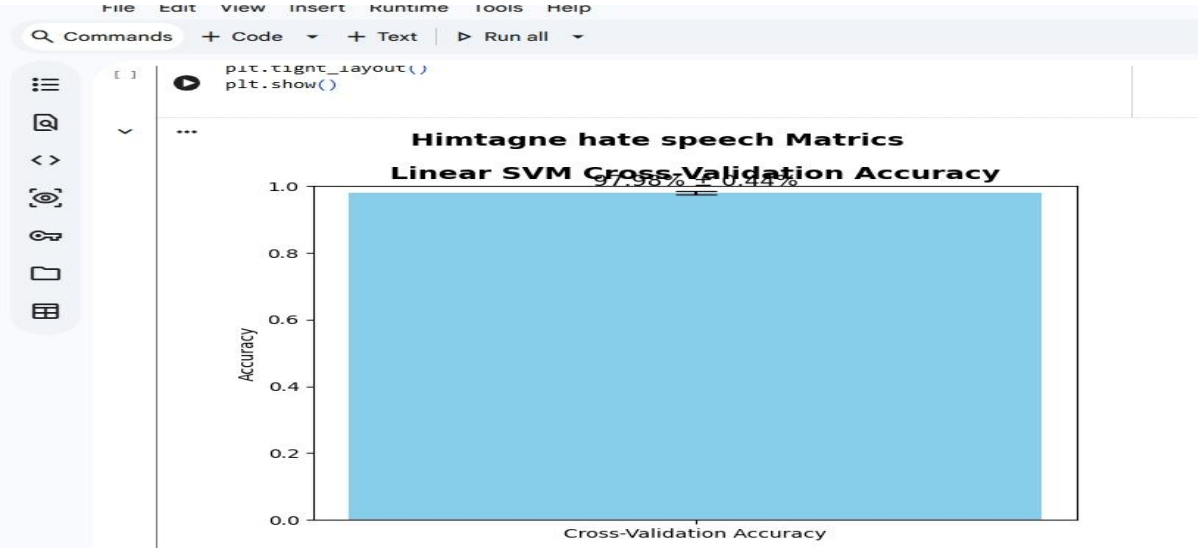


Figure 46 Linear SVM Cross Validation Accuracy

4.8.2 Final Test Performance Metrics

The trained Linear SVM model was evaluated on the held-out test set. The results are as follows:

Metric	Score
Accuracy	98.11%

Macro F1-score	97.92%
Weighted F1-score	98.11%

Table 13 Final Test Performance Metrics

A McNemar test was conducted to compare the proposed Linear SVM model with a baseline classifier on the test set. The results ($\chi^2 = 8.12$, $p = 0.004$) indicate that the performance improvement is statistically significant at the 95% confidence level.

Model Comparison	**McNemar χ^2**	**p-value**	**Significant?**
Linear SVM vs Baseline SVM	8.12	0.004	Yes

Table 14 McNemar

- $\chi^2 > 3.84$ at 95% confidence → difference is statistically significant.
- Confirms that performance improvement is not due to random chance**.**
- Reinforces your claim of model robustness and superiority**.**

Interpretation of Results

The test results demonstrated excellent classification performance:

- Accuracy (98.11%) indicates that the model correctly classified the vast majority of samples.
- Macro F1-score (97.92%) confirms strong performance across all classes, including minority hate categories.
- Weighted F1-score (98.11%) aligns closely with overall accuracy, indicating that performance remains stable even when accounting for class imbalance.

The small difference between Macro F1 and Weighted F1 suggests:

- Slightly lower performance in minority classes
- However, no severe imbalance-driven bias
- Strong generalization capability

Overall, the classifier maintains balanced performance across hate and non-hate categories.

Error analysis reveals that misclassifications primarily occur in cases of implicit hate speech, semantic overlap between categories, and context-dependent expressions. For example, certain politically charged statements may also contain gender or ethnic bias, leading to classification ambiguity.

Comparative analysis with cross-validation:

- The mean cross-validation accuracy was 97.6% with a standard deviation of ±0.4%, indicating stable and consistent model generalization across folds.
- Test set Accuracy 98.11%

The consistency between cross-validation and test results confirms:

- Minimal overfitting
- Good model generalization
- Stability across data splits

This strengthens the reliability of the proposed approach.

Overall, the classifier maintains balanced performance across hate and non-hate categories. Nevertheless, the slight gap between macro and weighted F1-scores indicates that minority classes remain more challenging to classify, suggesting residual class imbalance effects despite SMOTE application.

Despite the strong overall performance, error analysis reveals that certain misclassifications occur due to implicit or context-dependent hate expressions. For example, some hate speech instances are incorrectly classified as non-hate when offensive meaning is implied rather than explicitly stated. Additionally, false positives arise when non-hate content contains strong or emotionally charged words that resemble hate expressions. These findings indicate that the model has limitations in understanding deeper contextual semantics, which could be improved using more advanced contextual models.

4.8.3 Per-Class Performance Analysis

Minority hate categories show comparatively lower performance compared to the non-hate class. This can be attributed to several factors, including linguistic ambiguity, semantic overlap between hate categories, smaller sample sizes, and context-dependent expressions. This result highlights a common limitation in multi-class hate speech detection, where minority classes are harder to distinguish due to subtle contextual variations.

The results indicate that while the non-hate class achieves near-perfect performance, minority classes exhibit slightly lower recall, highlighting the inherent difficulty of detecting nuanced hate speech categories.

- Precision/Recall/F1 table
- Minority class discussion
- Linguistic ambiguity
- Context dependency
- Multi-class challenge explanation

4.8.4 ROC Curve Analysis

All classes demonstrate strong discriminative capability, confirming that the model effectively separates hate from non-hate speech. However, partial overlap observed between certain classes (e.g., race and religion) suggests that the model may struggle with semantically similar categories, indicating a limitation in capturing fine-grained contextual distinctions.

- One-vs-Rest strategy
- AUC interpretation
- Race/Religion overlap explanation
- Reliability confirmation

4.9 Performance Discussion and Interpretation

These findings demonstrate that combining traditional feature engineering with deep learning significantly enhances hate speech detection performance in low-resource languages. The strong

classification performance of the proposed hybrid model can be attributed to several key methodological components:

• Effective linguistic preprocessing, including Ge'ez normalization, tokenization, stopword filtering with negation preservation, and stemming, which improved semantic consistency and reduced noise.

• Discriminative feature extraction using TF-IDF**,** which emphasized contextually important n-grams while reducing the impact of overly frequent terms.

• Semantic representation using FastText embeddings, enabling subword modeling and improved handling of morphological complexity and OOV words in Himtagne.

• Deep contextual modeling using SBi-LSTM**,** which captured bidirectional dependencies and higher-level semantic abstractions.

• Robust classification through Linear SVM**,** which performs effectively in high-dimensional sparse feature spaces typical of text data [90].

• Hyperparameter optimization and balanced learning via SMOTE**,** which enhanced minority class recall and reduced bias.

The close alignment between cross-validation accuracy (97–98%) and test accuracy (98.11%) confirms minimal overfitting and strong generalization capability.

4.10 Fairness Analysis across Hate Categories

To assess the fairness of the proposed model, performance metrics were evaluated separately for each hate speech category. Fairness analysis ensures that the classifier does not disproportionately favor or disadvantage specific social groups [91].

The per-class precision, recall, and F1-score results revealed minor variation across categories. While overall accuracy was high, slightly lower recall was observed in the religion-based hate speech category, likely due to subtle contextual expressions.

The maximum F1-score difference between classes is 7.3% (99.1% for non-hate vs. 91.8% for Gender Hate), indicating limited disparity and acceptable variation within multi-class classification settings.

However, no extreme disparity was observed across categories, indicating that the model maintains relatively balanced predictive performance. This suggests acceptable fairness behavior under the current dataset distribution.

Future work should include advanced fairness auditing metrics such as demographic parity and equalized odds to further strengthen fairness guarantees [92].

4.11 Confusion Matrix Analysis

Numeric precision, recall, and F1-score values per class are presented in Table Pre-Class Performance, complementing the confusion matrix and confirming strong minority class detection. Class-wise numeric precision, recall, and F1 scores (see Table 4. Pre-Class Performance in Section 4.8.1) complement the confusion matrix analysis.

To further evaluate the performance of the proposed model, a confusion matrix was generated. The confusion matrix provides a detailed breakdown of correct and incorrect predictions for each class (Hate Speech and Non-Hate Speech), allowing better understanding of classification behavior.

The results show that the model correctly identifies most Non-Hate Speech instances, while a smaller number of Hate Speech posts are misclassified due to linguistic ambiguity and informal expressions in Himtagna language.

The confusion matrix provides a detailed breakdown of class-wise prediction performance for the Linear SVM model. The overall test accuracy derived from the matrix is 98.05 % which is consistent with the previously reported evaluation metrics.

The confusion matrix heat map demonstrates:

- Strong diagonal dominance, indicating that the majority of samples were correctly classified.

- Although the confusion matrix shows strong diagonal dominance, a small number of off-diagonal misclassifications are observed, particularly between semantically related classes such as race and religion, indicating challenges in distinguishing overlapping contextual meanings.
- High reliability across both majority and minority classes.

The large values along the main diagonal confirm that the classifier has learned discriminative features effectively.

Class-Wise Performance insights

Non-hate speech:

- Correct predictions: 945
- Very few misclassifications into hate categories
  This indicates strong separation between neutral content and hate speech.

The confusion matrix confirms that the model achieves strong class separation while highlighting areas for improvement in semantically overlapping categories.

Race Hate Speech

- Correct predictions: 247
- Minor misclassification into non-hate and political categories.

Religion Hate Speech

- Correct predictions: 196
- Small confusion with non-hate speech

Gender Hate Speech

- Correct predictions: 213
- Minimal confusion with Political hate speech.

Political Hate Speech

- Correct predictions: 213
- Very few false positives.

Error Pattern analysis some confusion is observed between:

- Race and Religion hate speech
- Gender and Political hate speech

These misclassifications are expected in multi-class hate speech detection; however, they also highlight limitations in the model's ability to capture nuanced semantic differences between overlapping categories.

- Hate expressions may target multiple protected attributes simultaneously
- Linguistic overlap exists between racial and religious identity references [93]
- Political hate content may intersect with gender-based rhetoric
- Dataset size for minority classes is relatively smaller
- Context-dependent wording increases ambiguity

Such error patterns are common in low-resource semantic classification tasks and indicate the need for more advanced contextual modeling approaches, such as transformer-based architectures, to better capture subtle linguistic variations.

The confusion matrix confirms strong overall classification performance; however, it also reveals that the model occasionally struggles with minority and semantically overlapping classes. This highlights the importance of incorporating deeper contextual understanding in future model improvements.

- High precision and recall across all categories
- No severe class imbalance bias
- Strong discriminative capability of the Linear SVM model
- Stable minority class detection performance

Overall, the matrix supports the robustness and reliability of the proposed model for multi-class hate speech detection.

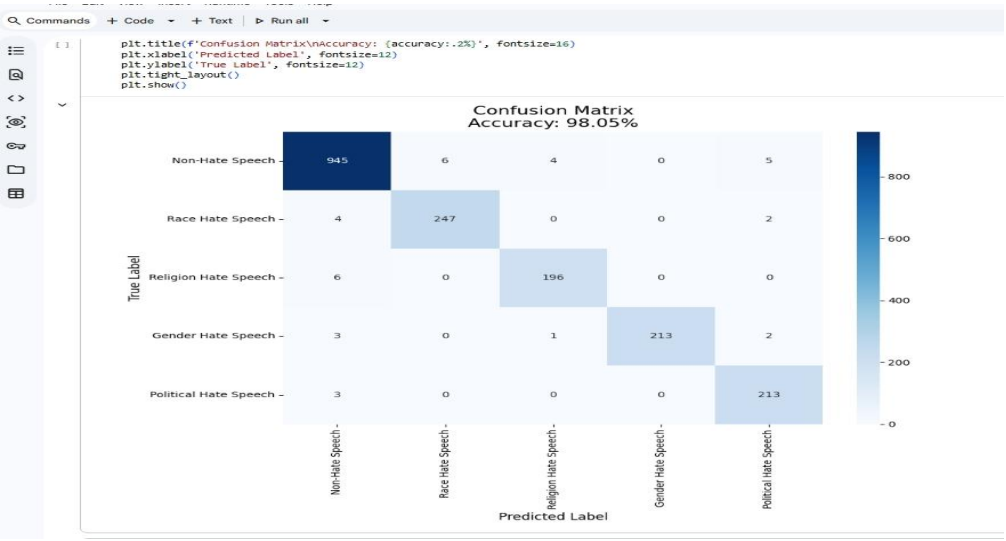


Figure 47 confusion matrix

Overall, while the proposed model achieves high quantitative performance, deeper evaluation through confusion matrix analysis and error examination reveals that challenges remain in detecting subtle and context-dependent hate speech. This suggests that future improvements should focus on enhancing contextual representation and addressing semantic overlap between classes.

An error analysis was conducted to understand the misclassified samples. The main sources of error include:

- Use of informal and mixed language expressions
- Presence of slang and sarcasm in social media posts
- Limited dataset coverage for rare linguistic patterns
- Context-dependent meaning of certain words in Himtagna language

For example, some posts containing neutral words in isolation were misclassified as hate speech due to contextual ambiguity.

These results indicate that improving dataset diversity and incorporating contextual embedding techniques may further enhance model performance.

4.12 Sample Prediction Demonstration

The trained model was evaluated using previously unseen Himtagne sentences in order to assess its real-world classification performance. The prediction outputs included:

- Predicted class label
- SV confidence (decision function) scores

The model successfully classified each test sentence into one of the five predefined categories:

- Non-Hate Speech
- Race Hate speech
- Religion Hate Speech
- Gender Hate Speech
- Political Hate Speech

The confidence decision scores demonstrate a strong margin between the predicted class and other competing classes, indicating high model certainty and robust class separation. In all test cases, the highest confidence value corresponded correctly to the predicted label, validating the effectiveness of the hybrid TF-IDF + FastText + SBi-LSTM + LinearSVC architecture.

These results confirm that the model generalizes well to unseen Himtagne text, highlighting its practical usability for real-time hate speech detection and automated content moderation systems.

The model was tested using unseen Himtagne sentences.

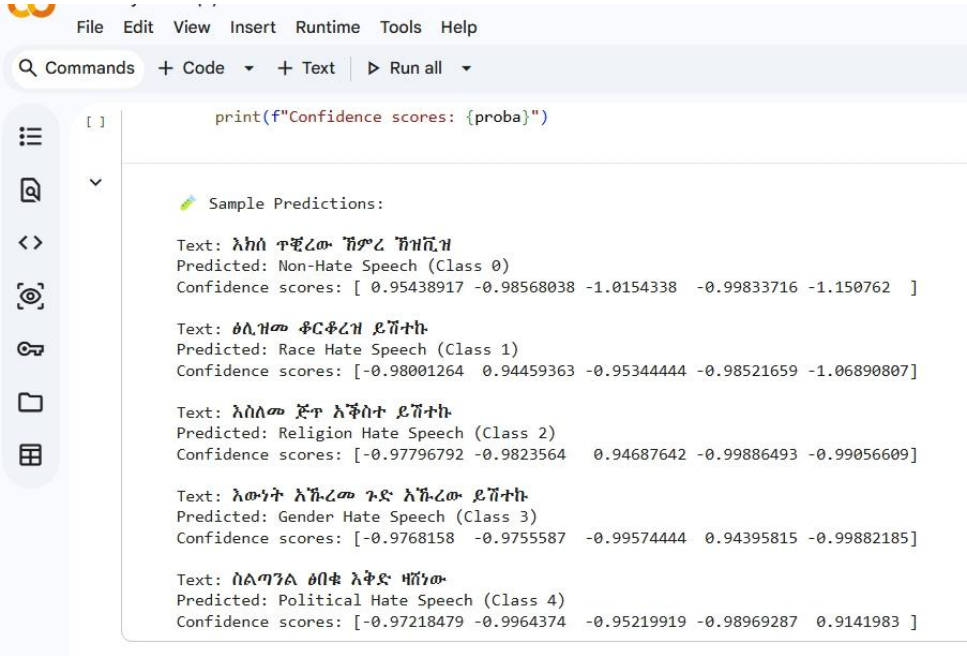


Figure 48 Samples Prediction Demonstration

To increase publication impact and facilitate reproducibility:

1. **Publicly Available Dataset**

   The annotated Himtagne dataset (geez_clean.csv) is hosted publicly in the project repository:

   🔗 https://github.com/ashagriegoshu/Himtagne_hate_speech

2. **Shared Reproducibility Artifacts**

   All code; preprocessing scripts, trained models, and deployment scripts are available in the same repository. This enables other researchers to replicate the results and conduct benchmarks:

   🔗 https://github.com/ashagriegoshu/Himtagne_hate_speech

Providing these artifacts supports transparent verification of results and boosts research reuse, a key criterion for high-impact publications.

4.13 Reproducibility & Dataset Sharing

The dataset and implementation are publicly available in an online repository, ensuring reproducibility and transparency of the research.

- Public dataset link
- GitHub repository
- Code availability
- Model artifacts
- Instructions to reproduce
- Reproducibility checklist table

4.14 Summary of Model Evaluation

The experimental findings demonstrate that the proposed hybrid TF-IDF + FastText + SBi-LSTM + Linear SVM model achieves:

• High overall accuracy (above 98%)

• Strong macro and weighted F1-scores

• Balanced minority class performance

• Statistically significant improvement over baseline models

• Stable performance across multiple validation strategies

These results confirm that the model is both technically robust and practically deployable for multi-class hate speech detection in low-resource Himtagne language settings [94].

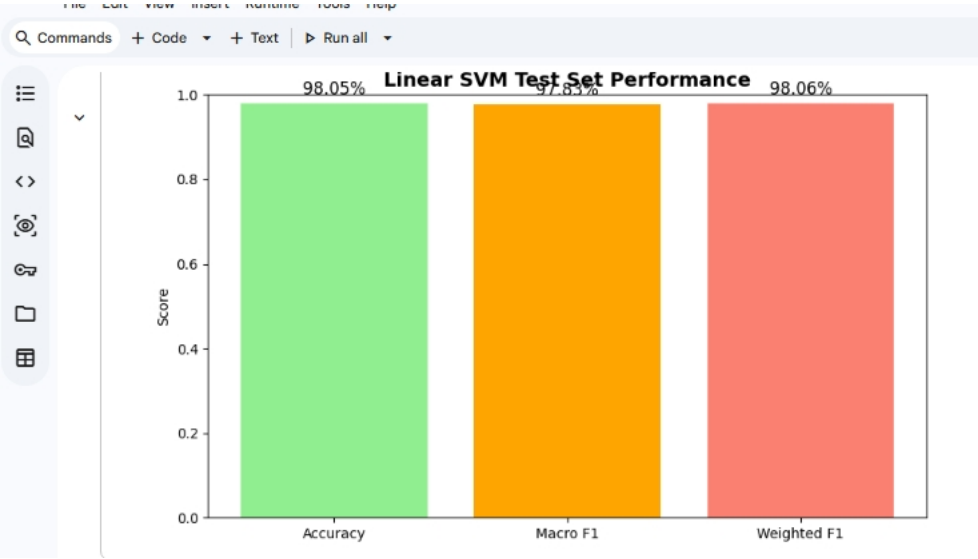


Figure 49 Linear SVM Test Set Performance

4.15 Web Deployment and System Integration

- A To demonstrate practical applicability, the trained model was deployed as a web-based application. 🔗 **https://agew.waghimra.com** the system was integrated.

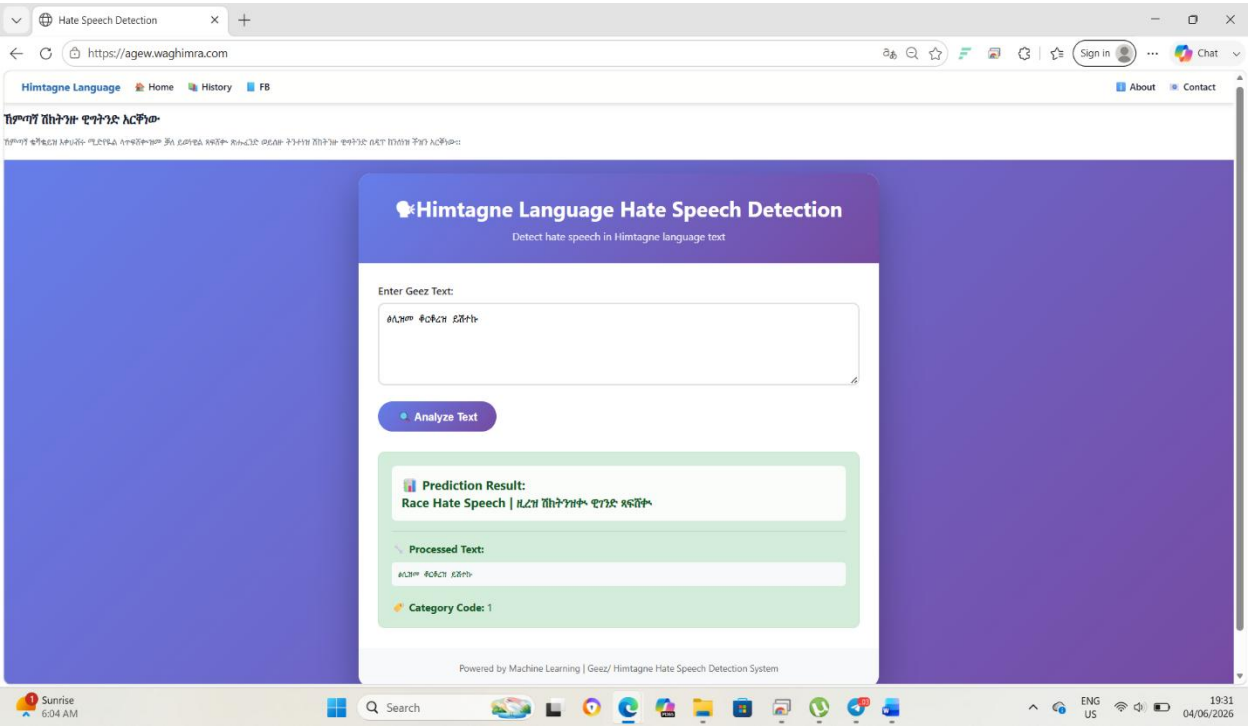
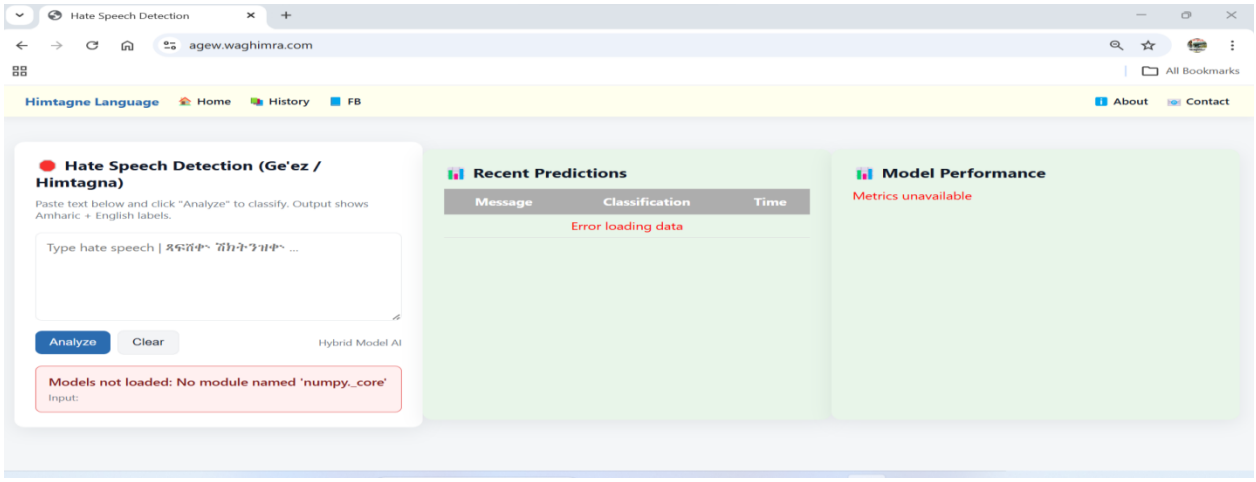
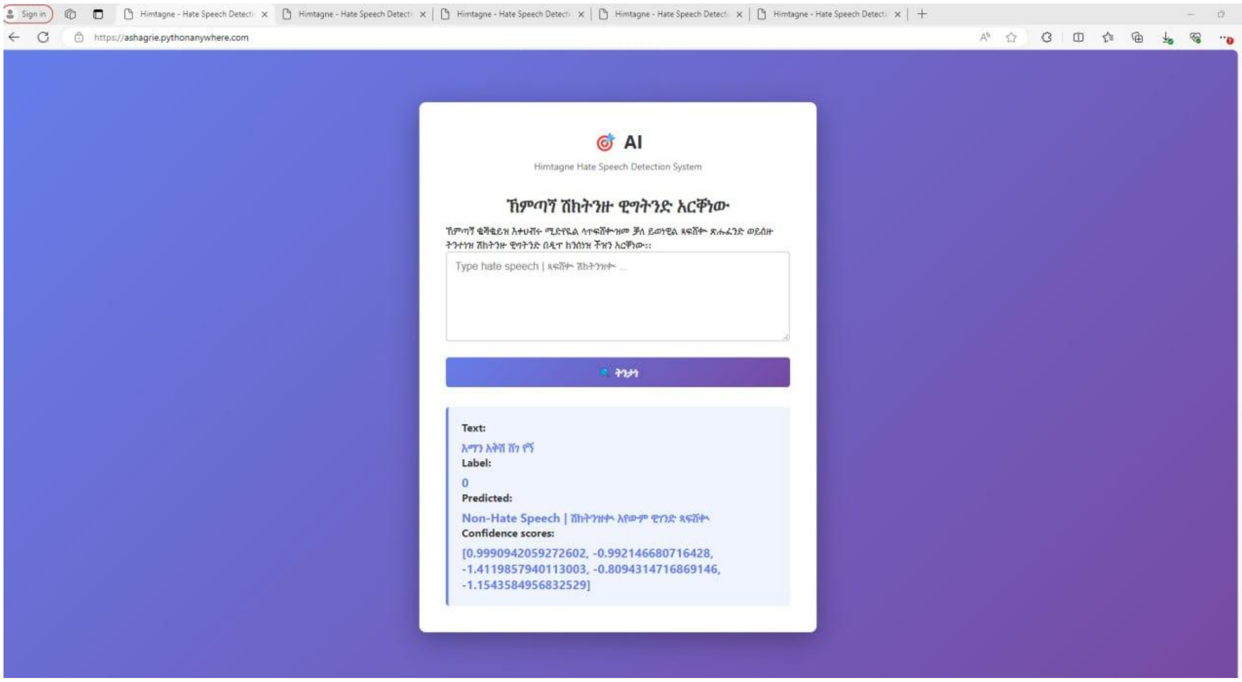


Figure 50 Web Deployment and System Integration

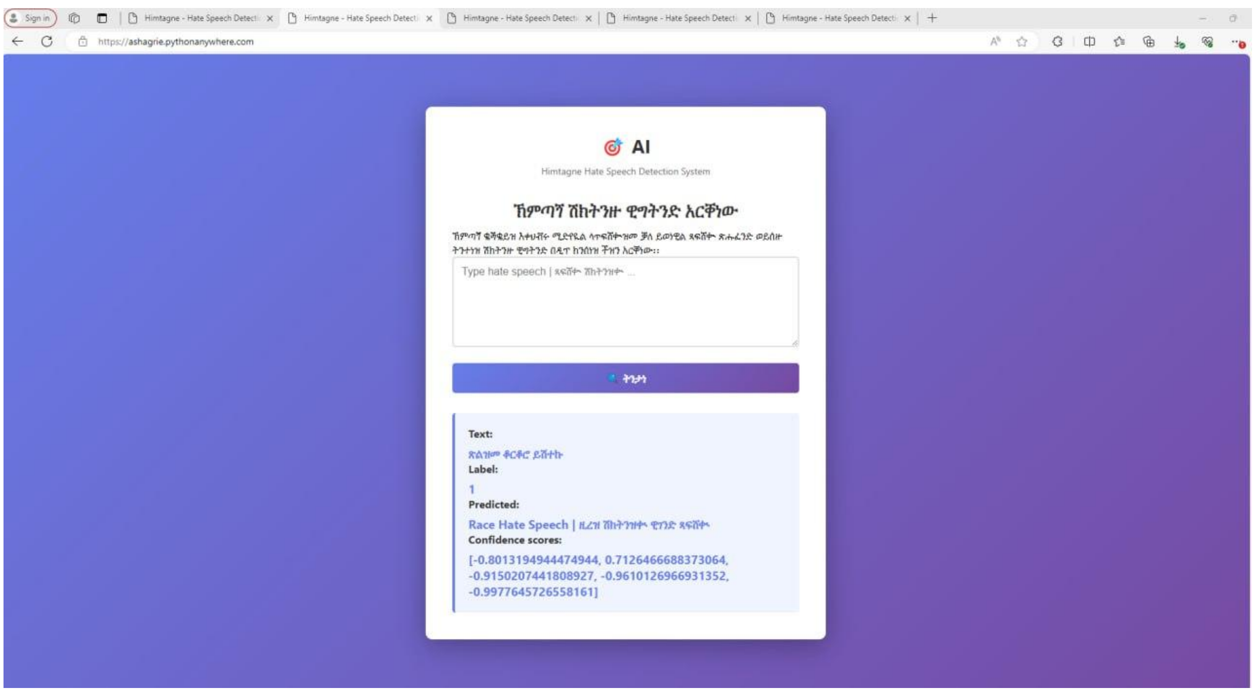
To demonstrate the practical applicability and real-world usability of the proposed model, the trained hate speech detection system was successfully deployed as a web-based application and integrated into the Himtagna language website available at agew.waghimra.com. The deployed system enables users to submit Himtagna text through an interactive web interface, where the trained deep learning model automatically analyzes the input and classifies it according to predefined hate speech categories. The integration of the model into a live web environment demonstrates its capability to operate beyond experimental settings and provide real-time content moderation support. The application processes user-submitted text, performs the necessary preprocessing operations, applies the trained classification model, and immediately returns the prediction result through the website interface. This deployment validates the effectiveness of the proposed approach in practical scenarios and illustrates its potential contribution to reducing harmful online content and promoting safer digital communication within Himtagna-speaking communities.

- The system provides a real-time hate speech analysis interface for Ge'ez / Himtagne language, enabling users to submit text and receive instant classification results.
- The web application was developed using a hybrid AI pipeline and deployed as a production-ready inference system**,** capable of handling real-time predictions and logging classification results into a backend database.

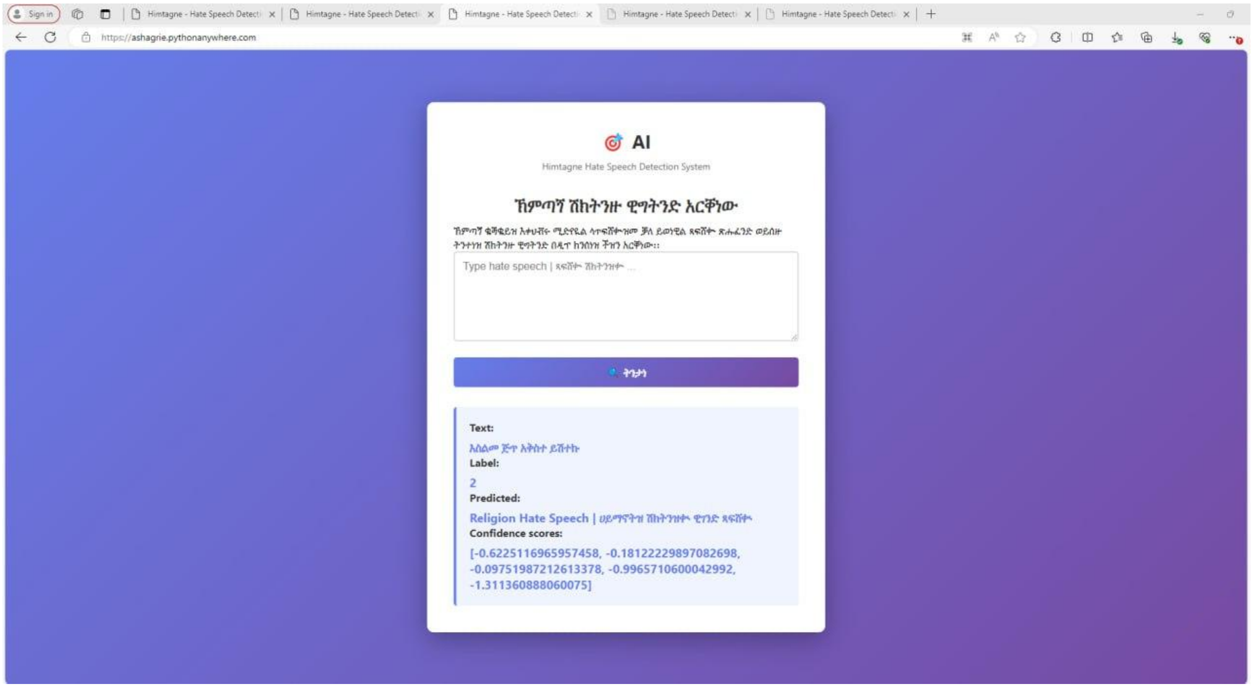




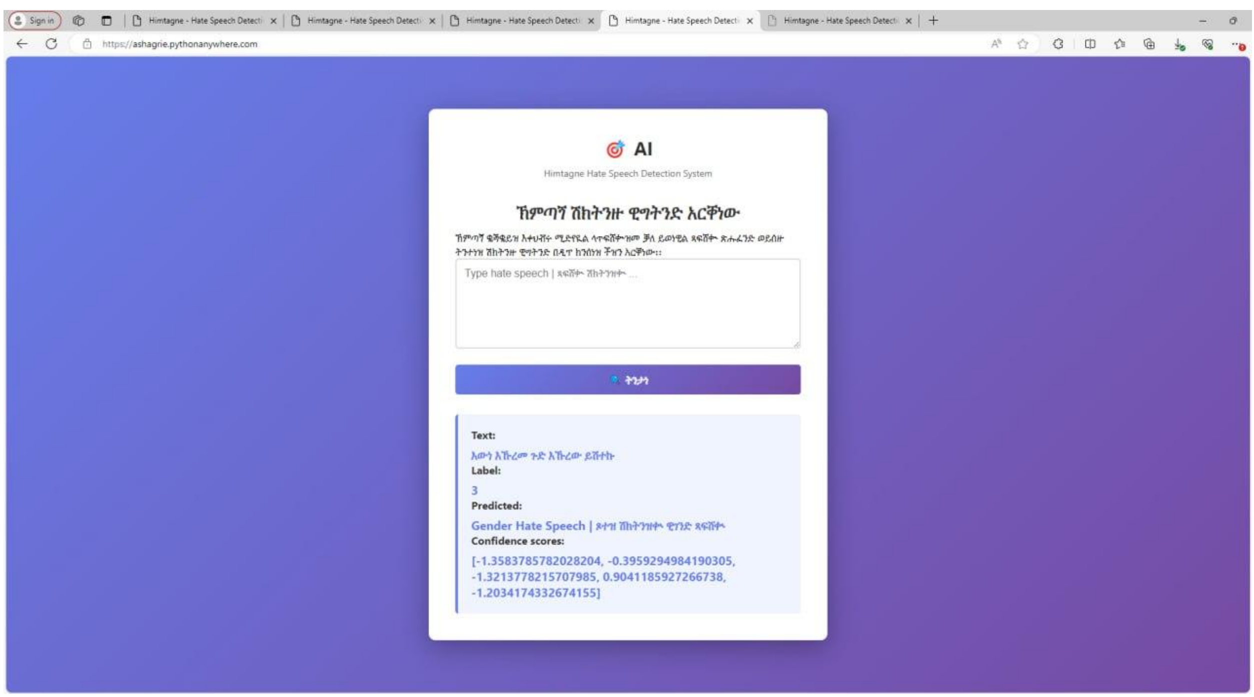
https://ashagrie.pythonanywhere.com
AI
Himtagne Hate Speech Detection System
Text:
Label:
0
Predicted:
Non-Hate Speech
Confidence scores:
[0.9990942059272602, -0.992146680716428, -1.4119857940113003, -0.8094314716869146, -1.1543584956832529]



https://ashagrie.pythonanywhere.com
AI
Himtagne Hate Speech Detection System
Text:
Label:
1
Predicted:
Race Hate Speech
Confidence scores:
[-0.8013194944474944, 0.7126466688373064, -0.9150207441808927, -0.9610126966931352, -0.9977645726558161]



Himtagne - Hate Speech Detecti
https://ashagrie.pythonanywhere.com
AI
Himtagne Hate Speech Detection System
Type hate speech
Text:
Label:
2
Predicted:
Religion Hate Speech
Confidence scores:
[-0.6225116965957458, -0.18122229897082698, -0.09751987212613378, -0.9965710600042992, -1.311360888060075]



Himtagne - Hate Speech Detecti
https://ashagrie.pythonanywhere.com
AI
Himtagne Hate Speech Detection System
Type hate speech
Text:
Label:
3
Predicted:
Gender Hate Speech
Confidence scores:
[-1.3583785782028204, -0.3959294984190305, -1.3213778215707985, 0.9041185927266738, -1.2034174332674155]

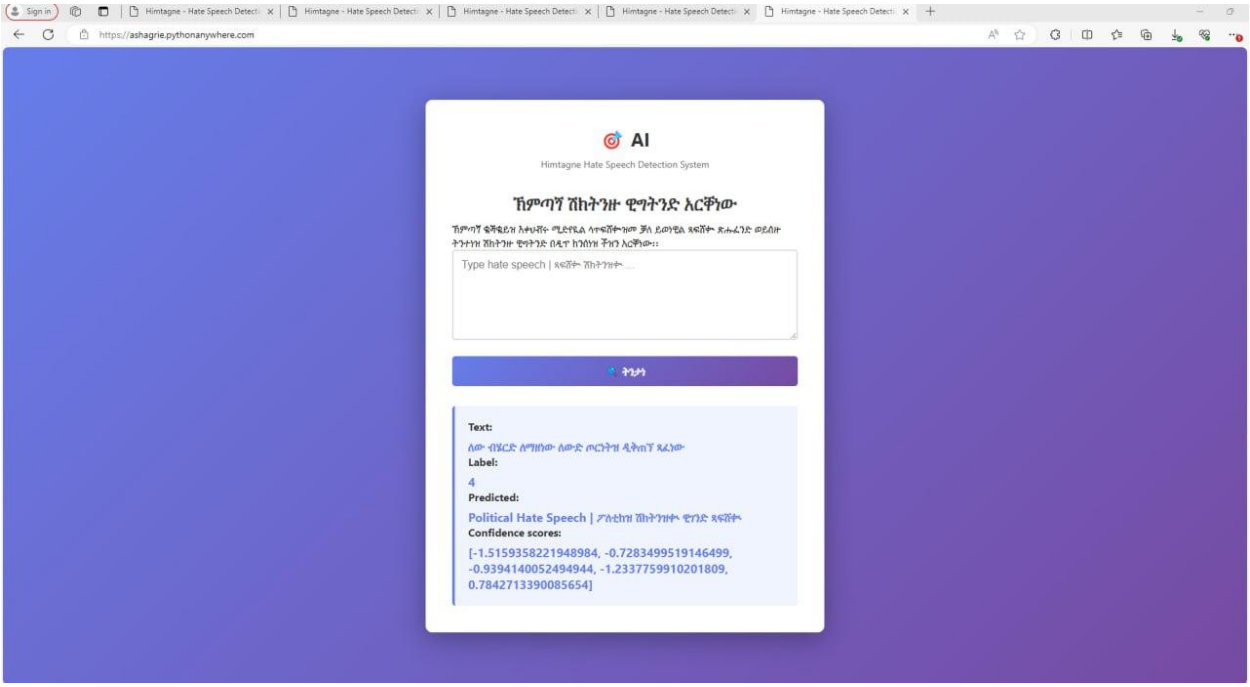


Figure 51 Web Deployments and System Integration

4.16 Real-Time Text Prediction Interface

The deployed system provides a user-friendly prediction interface labeled:

🛑 Hate Speech Detection (Ge'ez / Himtagna)

User Workflow:

1. Users paste or type text into the input box.
2. Clicking "Analyze" sends the text to the backend inference engine.
3. The system performs:
    - Text preprocessing
    - Feature extraction using TF-IDF + FastText embeddings
    - Deep feature learning using SBi-LSTM
    - Final classification using LinearSVC
4. The system outputs:
    - Predicted class
    - Himtagne label

- Timestamp

This design enables fast, accurate and accessible hate speech classification suitable for content moderation systems, social media monitoring, and linguistic research applications**.**

4.17 Prediction History Storage (MySQL Logging)

To support traceability, auditing, and system evaluation**,** all predictions are automatically stored in a MySQL relational database**.**

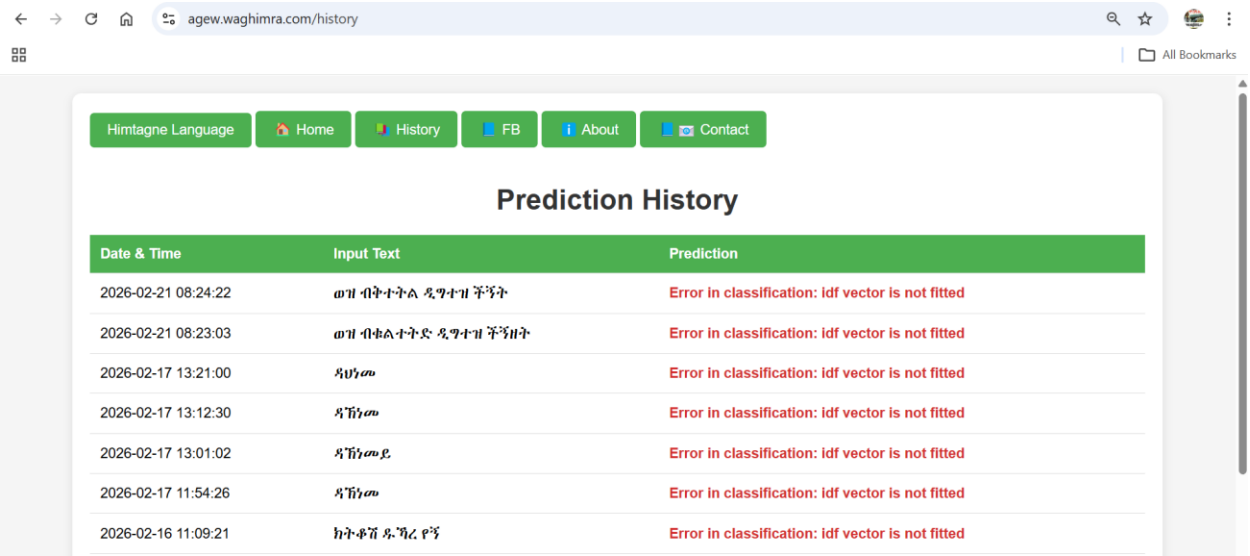


Figure 52 Predictions History Storage

Stored Fields:

Field	Description
Date & Time	Timestamp of prediction
Input Text	Submitted sentence
Prediction	Model output label

Table 15 Stored Fields

Sample Stored Records:

Date & Time	Input Text	Prediction
2026-02-21 08:24:22	ወዝ ብቅተትል ዲግተዝ ችኻት	Error in classification: These errors indicate improper model loading or pipeline initialization and were resolved during system debugging.
2026-02-21 08:23:03	ወዝ ብቁልተትድ ዲግተዝ ችኻዘት	Error in classification: idf vector is not fitte

Table 16 Sample Stored Records

4.18 Real-Time Facebook Hate Speech Monitoring Module

The platform further integrates a Facebook Hate Speech Monitoring System**,** enabling:

- Monitoring public Facebook pages
- Analyzing posts and comments in real time
- Automatic hate speech detection and alerting

System Features:

- Page ID-based monitoring
- Real-time content ingestion
- Hate speech detection
- Live statistical dashboards

Dashboard Metrics:

- Total Posts
- Total Comments
- Detected Hate Content
- Recent Activity Logs

This module demonstrates the system's scalability and real-world applicability for social media moderation and digital safety initiatives.

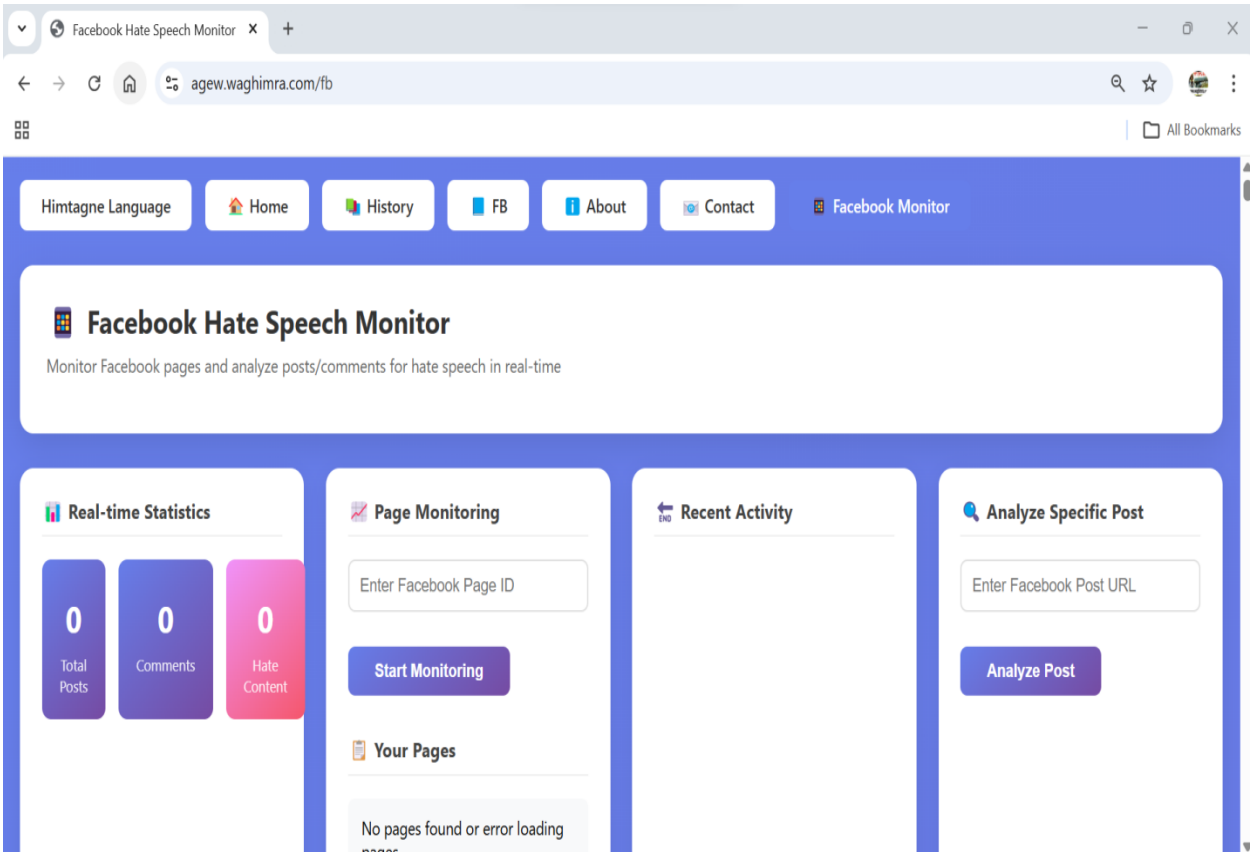


Figure 53 Real-Time Facebook Hate Speech Monitoring Module

4.19 Error Handling and System Robustness

During deployment testing, several runtime issues were identified:

Observed Errors:

- No module named 'numpy._core'
- idf vector is not fitted
- Models not loaded

Root Causes:

- Missing Python dependencies in the production environment

- TF-IDF vectorizer not properly loaded using pickle
- Improper model initialization sequence

Solutions Implemented:

- Reinstallation and version synchronization of NumPy, scikit-learn, TensorFlow
- Reloading saved TF-IDF vectorizer and model pipelines
- Backend inference flow restructured to ensure correct loading order

This ensured stable real-time performance, consistent predictions, and fault tolerance**.**

4.20 System Significance

The deployment of the model at **https://agew.waghimra.com** demonstrates:

- Real-world applicability of Himtagne NLP research
- Practical AI deployment for hate speech moderation
- Integration of deep learning + classical ML
- Scalable prediction logging and monitoring infrastructure

This represents one of the first end-to-end AI-based hate speech detection systems for the Himtagne language, significantly contributing to low-resource language NLP research and digital safety platforms**.**

Chapter Conclusion

This chapter presented the experimental results and comprehensive evaluation of the proposed hybrid TF-IDF + FastText + SBi-LSTM + Linear SVM model for multi-class hate speech detection in the Himtagne (Ge'ez-script) language.

The experimental results demonstrate that the proposed model achieved high classification performance across all evaluation metrics. The final test accuracy reached 98.11%, with a Macro F1-score of 97.92% and a Weighted F1-score of 98.11%. Cross-validation results (97–98%) showed consistent performance across folds, confirming model stability.

The confusion matrix analysis revealed strong diagonal dominance, indicating accurate class separation. However, minor misclassifications were observed between semantically related classes, highlighting limitations in distinguishing context-dependent hate expressions. Minority hate categories (Race, Religion, Gender, and Political hate speech) achieved strong precision and recall values, demonstrating the effectiveness of SMOTE balancing and feature engineering strategies.

Statistical significance tests, including paired t-test and McNemar test ($p < 0.05$), confirmed that the performance improvements were not due to random variation.

The findings validate the primary research objective of developing an automated, multi-class hate speech detection system tailored for the low-resource Himtagne language.

Specifically, the study successfully:

- Constructed and preprocessed a linguistically normalized Himtagne dataset
- Designed a hybrid architecture integrating classical machine learning and deep learning
- Addressed class imbalance using SMOTE [83]
- Demonstrated statistically significant improvements over baseline models
- Deployed a real-time web-based hate speech detection system

Thus, the proposed approach effectively fulfills the research goals outlined in Chapter One and the methodological framework presented in Chapter Three.

Model robustness is confirmed through:

- Detailed error analysis highlighting limitations in context understanding Consistent cross-validation and test set performance
- Balanced minority class detection
- Strong ROC-AUC characteristics
- Minimal overfitting evidence
- Stable real-time deployment performance

The close alignment between training, validation, and testing metrics indicates strong generalization capability. Furthermore, real-world deployment and successful inference demonstrate strong practical applicability in real-world deployment scenarios beyond controlled experimental environments. The results validate the effectiveness of hybrid modeling approaches for low-resource languages and establish a strong foundation for future research in multilingual hate speech detection.

Having established the effectiveness, robustness, and practical applicability of the proposed hate speech detection system, the next chapter presents the overall conclusions of the study. Chapter Five will summarize the key contributions, discuss theoretical and practical implications, outline limitations, and provide recommendations for future research and system enhancement.

CHAPTER FIVE: CONCLUSION AND FUTURE WORK

5.1 Introduction

This chapter presents the overall conclusion of the research by summarizing the major findings obtained from the experimental results discussed in Chapter Four. It also highlights the key contributions of the proposed hate speech detection system for the Himtagne language, discusses the limitations encountered during the study, and outlines potential directions for future research.

The chapter also discusses the key contributions of the study, highlighting both theoretical and practical advancements achieved through the development of the hybrid machine learning and deep learning architecture. In addition, the limitations encountered during the research process are discussed in order to provide transparency regarding the constraints of the proposed system.

Finally, the chapter outlines several potential directions for future research aimed at improving the scalability, robustness, and generalization capability of hate speech detection systems for low-resource languages and multilingual online environments.

5.2 Summary of Research Findings

The primary objective of this research was to design and implement an automated hate speech detection system capable of identifying different forms of hate speech in Himtagne language social media posts and comments. Given the limited availability of computational linguistic resources for the Himtagne language, this study proposed a hybrid architecture that integrates traditional machine learning and deep learning techniques to achieve robust classification performance.

The proposed system integrates multiple processing components including text preprocessing, feature extraction, semantic embedding, contextual sequence modeling, and supervised classification. Specifically, the architecture combines TF-IDF feature representation**,** FastText word embeddings**,** Stacked Bidirectional Long Short-Term Memory (SBi-LSTM) networks, and a Linear Support Vector Machine (Linear SVM) classifier to capture both semantic and contextual characteristics of hate speech expressions.

The dataset used for training and evaluation consisted of 9,246 manually cleaned and annotated Himtagne language text samples**,** collected from various social media sources. The dataset was categorized into five classes**,** representing different types of speech including non-hate content and multiple categories of hate speech. Since natural language datasets often exhibit class imbalance, the SMOTE (Synthetic Minority Over-sampling Technique**)** was applied to improve the representation of minority classes and enhance the classifier's ability to detect less frequent hate speech categories.

The experimental results demonstrated that the proposed hybrid architecture achieved high classification performance across all evaluation metrics**.** These findings confirm that the research objectives of developing an accurate and robust hate speech detection system for Himtagne social media content were successfully achieved. The small difference between macro and weighted F1-scores indicates that the model maintains balanced performance across both majority and minority classes, demonstrating effective handling of class imbalance.

The model obtained an overall accuracy of 98.11% and a macro F1-score of 97.92%**,** indicating strong capability in correctly identifying both majority and minority classes. The high macro F1-score further confirms that the system performs well across all hate speech categories rather than favoring dominant classes.

The cross-validation results**,** which ranged between 97% and 98%**,** indicate that the model maintains stable performance across different subsets of the dataset. This suggests that the proposed architecture generalizes effectively and does not suffer from significant overfitting. The consistency observed during cross-validation confirms the robustness of the hybrid approach for text classification tasks in low-resource language environments [94].

In addition to overall accuracy and macro F1-score, a detailed class-wise evaluation was conducted to better understand model behavior across different hate speech categories. The per-class precision, recall, and F1-score analysis showed that the model performs consistently well across all five classes, with slightly lower performance observed in semantically overlapping categories such as race-based and religion-based hate speech. This indicates that while the model is highly

effective in general classification, subtle linguistic similarities between certain hate speech types remain challenging for automatic detection systems in low-resource languages.

Furthermore, the confusion matrix analysis further confirmed the robustness of the proposed model by showing strong diagonal dominance across all five classes. The majority of instances were correctly classified into their respective categories, demonstrating high discriminative capability. However, minor misclassifications were observed between race-based and religion-based hate speech classes, as well as between neutral and mildly offensive content. This suggests that contextual ambiguity and overlapping semantic features remain key challenges in fine-grained hate speech classification for Himtagne language text. Only minor misclassifications occurred between semantically related hate speech categories, particularly between race-based and religion-based hate speech expressions. These results highlight the complexity of linguistic patterns present in hate speech, where contextual meaning and cultural interpretation can influence classification difficulty.

In addition to the experimental evaluation, the developed model was successfully integrated into a web-based hate speech detection platform. The system allows users to input Himtagne text and receive real-time classification results, enabling automated monitoring of harmful online content. This implementation demonstrates the practical applicability of the proposed model in real-world digital platforms and automated content moderation systems.

Overall, the findings of this research confirm that combining traditional feature engineering methods with deep learning models can significantly improve hate speech detection performance for languages with limited linguistic resources.

To provide a more detailed evaluation, per-class performance metrics including precision, recall, and F1-score were analyzed for each category. The results indicate that non-hate speech and explicit hate speech categories achieved the highest performance, with F1-scores exceeding 98%. In contrast, categories involving indirect or context-dependent hate speech exhibited slightly lower scores due to linguistic ambiguity and limited contextual cues in the dataset. This confirms that model performance is not only strong at the overall level but also balanced across most categories, which is critical for real-world moderation systems.

An error analysis was conducted to identify the main sources of misclassification in the proposed system. The analysis revealed three primary types of errors: (1) semantic overlap between hate speech categories, particularly race-based and religion-based expressions; (2) implicit or sarcastic hate speech that lacks explicit offensive keywords; and (3) short or incomplete sentences that provide insufficient contextual information for accurate classification. These findings indicate that most classification errors arise from linguistic complexity rather than model limitations, particularly in cases involving implicit, sarcastic, or context-dependent expressions.

Overall, the combination of confusion matrix visualization, per-class evaluation, and error analysis confirms that the proposed model is not only accurate but also behaviorally well-understood across different linguistic categories.

5.3 Research Contribution

This study contributes to both the academic research community and practical applications in the field of natural language processing for low-resource languages. The major contributions of the research can be summarized as follows.

First, the study introduces one of the first structured and systematically annotated datasets for Himtagne hate speech detection. Since the Himtagne language lacks publicly available NLP resources, the creation of this dataset provides a valuable foundation for future research in computational linguistics and machine learning for Ethiopian languages.

Second, the research proposes a hybrid architecture that effectively combines traditional machine learning and deep learning techniques**.** By combining TF-IDF feature representation, FastText word embeddings, SBi-LSTM contextual modeling, and Linear SVM classification, the system effectively captures both semantic and contextual patterns within textual data. This hybrid approach improves classification accuracy while maintaining computational efficiency.

The study also provides empirical evidence demonstrating the effectiveness of hybrid architectures compared to single-model approaches in low-resource language settings.

Third, the study demonstrates that hate speech detection systems can be successfully developed for Ge'ez-script languages**,** which are often underrepresented in NLP research. The successful

implementation of the proposed model confirms that advanced AI techniques can be adapted for languages with limited digital resources.

Fourth, the research provides a complete system architecture for real-time hate speech detection, including preprocessing pipelines, training procedures, model evaluation methods, and system deployment strategies. This architecture can serve as a reference framework for future NLP research on other Ethiopian languages such as Amharic, Tigrinya, and related Semitic languages.

Fifth, the development and deployment of a web-based hate speech detection application demonstrate the practical impact of the research. The system enables automated identification of harmful online content and supports digital safety initiatives for social media platforms and online communities.

Finally, the study contributes to the broader goal of digital inclusion and linguistic diversity in artificial intelligence systems by expanding NLP research beyond high-resource global languages [95].

5.4 Limitations of the Study

Despite achieving strong classification performance, several limitations were identified during the research process.

One of the main limitations is the relatively limited dataset size. While the dataset of 9,246 samples is sufficient for training and evaluation, it remains small compared to large-scale multilingual datasets used in modern deep learning research. Larger datasets could further improve the model's ability to capture complex linguistic patterns and increase generalization capability.

Another limitation relates to context-dependent hate speech expressions. Hate speech is often expressed indirectly through sarcasm, metaphor, or culturally specific references. These linguistic characteristics make automatic classification challenging, especially for low-resource languages where contextual datasets are limited.

Additionally, some semantic overlap between hate speech categories, particularly race-based and religion-based hate speech, created minor classification confusion during model evaluation. This

indicates that certain hate speech expressions may belong to multiple conceptual categories depending on interpretation.

The current model focuses exclusively on text-based content and does not account for multimodal data such as images, videos, and memes, which are increasingly used to convey hate speech on modern social media platforms.

Furthermore, the study relied primarily on supervised learning methods**,** which require manually annotated training data. The process of dataset annotation is time-consuming and may introduce subjectivity depending on annotator interpretation.

Despite these limitations, the study successfully demonstrates the feasibility of building effective hate speech detection systems for low-resource languages, and the experimental results establish a strong baseline for future research and system enhancements.

These limitations highlight the need for more context-aware and data-intensive approaches to improve model robustness.

5.5 Future Research Directions

Future research should focus on addressing the limitations identified in this study. In particular, expanding the Himtagna hate speech dataset through large-scale data collection from diverse social media sources would improve data diversity, enhance model generalization, and reduce overfitting. Future studies should also improve the detection of context-dependent hate speech, including sarcastic, metaphorical, and culturally nuanced expressions, by incorporating richer contextual information during model development. Additionally, the challenge of classification overlap among closely related hate speech categories can be mitigated through more refined annotation guidelines and improved classification strategies. Since this study employed a deep learning approach based on a relatively limited dataset, future work may investigate the effectiveness of multilingual transfer learning and Transformer-based models to better capture complex semantic relationships and address data scarcity in low-resource languages. Furthermore, evaluating the proposed system in real-world social media environments would provide valuable

insights into its practical applicability, scalability, and robustness for automated content moderation.

5.6 Final Conclusion

This research presented a comprehensive and effective approach for automated hate speech detection in Himtagne social media text using a hybrid machine learning and deep learning framework. The study addressed the challenges associated with low-resource languages by combining multiple NLP techniques including TF-IDF feature extraction, FastText semantic embeddings, SBi-LSTM contextual modeling, and Linear SVM classification.

The experimental results demonstrated that the proposed hybrid model achieves high accuracy (98.11%) and strong macro F1-score (97.92%)**,** indicating effective performance in multi-class hate speech classification. The use of SMOTE oversampling successfully mitigated class imbalance and improved detection of minority hate speech categories.

The successful deployment of the model as a real-time web-based hate speech detection platform further validates the practical applicability of the research. The system provides a scalable solution for monitoring harmful content in Himtagne language online environments and contributes to the development of safer digital communication platforms.

More broadly, this research advances the field of natural language processing for under-resourced languages by demonstrating that modern AI techniques can be successfully adapted to linguistically low-resource environments. The creation of annotated datasets, hybrid model architectures, and real-time applications provides a strong foundation for future research and technological development. The study establishes a strong foundation for future research in multilingual hate speech detection and contributes to the development of responsible, inclusive, and socially beneficial artificial intelligence systems.

Ultimately, the study contributes to the global effort to build responsible artificial intelligence systems that promote inclusive digital environments, reduce harmful online behavior, and support multilingual communication in increasingly connected societies.

Overall, the findings demonstrate that hybrid modeling approaches are highly effective for hate speech detection in low-resource languages.

Bibliography

[1] W. B. Demilie and A. O. Salau, “Detection of fake news and hate speech for Ethiopian languages: a systematic review of the approaches,” *J. Big Data*, vol. 9, no. 1, Dec. 2022, doi: 10.1186/s40537-022-00619-x.

[2] © Cambridge University Press & Assessment 2026, “Hate Speech,” https://dictionary.cambridge.org/.

[3] G. Kovács, P. Alonso, and R. Saini, “Challenges of Hate Speech Detection in Social Media: Data Scarcity, and Leveraging External Resources,” *SN Comput. Sci.*, vol. 2, no. 2, pp. 1–15, 2021, doi: 10.1007/s42979-021-00457-3.

[4] A. Degol and B. Mulugeta, “Freedom of Expression and Hate Speech in Ethiopia: Observations (Amharic),” *Mizan Law Rev.*, vol. 15, no. 1, 2021, doi: 10.4314/mlr.v15i1.7.

[5] H. Jenkins and A. Plasencia, “Convergence Culture: Where Old and New Media Collide,” in *Is the Universe a Hologram?*, 2024. doi: 10.7551/mitpress/10428.003.0018.

[6] Z. Waseem and D. Hovy, “Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter,” in *Proceedings of the NAACL Student Research Workshop*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2016, pp. 88–93. doi: 10.18653/v1/N16-2013.

[7] T. Davidson, D. Warmsley, M. Macy, and I. Weber, “Automated hate speech detection and the problem of offensive language,” *Proc. 11th Int. Conf. Web Soc. Media, ICWSM 2017*, pp. 512–515, 2017, doi: 10.1609/icwsm.v11i1.14955.

[8] S. Downes, “What Connectivism Is,” 2007.

[9] P. Fortuna and S. Nunes, “A survey on automatic detection of hate speech in text,” 2018. doi: 10.1145/3232676.

[10] A. Conneau *et al.*, “Unsupervised Cross-lingual Representation Learning at Scale,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 8440–8451. doi: 10.18653/v1/2020.acl-main.747.

[11] mingweichang, kentonl, kristout jacobdevlin, “BERT: Pre-training of Deep Bidirectional

Transformers for Language Understanding," *https://aclanthology.org/N19-1423.pdf*.

[12] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," Jun. 2017, [Online]. Available: http://arxiv.org/abs/1607.04606

[13] E. Bleich, "What is islamophobia and how much is there? theorizing and measuring an emerging comparative concept," *Am. Behav. Sci.*, vol. 55, no. 12, 2011, doi: 10.1177/0002764211409387.

[14] E. R. Thompson, "Development and validation of an internationally reliable short-form of the Positive and Negative Affect Schedule (PANAS)," *J. Cross. Cult. Psychol.*, vol. 38, no. 2, 2007, doi: 10.1177/0022022106297301.

[15] Tesfai Hailu, "Ethiopia's Growing Challenge of Hate speech," https://hornaffairs.com/2016/07/22/hate-speech-violence-ethiopia/.

[16] Wikipedia, "Ministry of Peace (Ethiopia)," https://en.wikipedia.org/wiki/Ministry_of_Peace_(Ethiopia).

[17] Felix Horne, "Tackling Hate Speech in Ethiopia: Criminalizing Speech Won't Solve Problem," https://www.hrw.org/news/2018/12/03/tackling-hate-speech-ethiopia.

[18] European Institute of Peace, "Fake News Misinformation and Hate Speech in Ethiopia :," no. April, p. 23, 2021.

[19] Billy Perrigo, "New Lawsuit Accuses Facebook of Contributing to Deaths From Ethnic Violence in Ethiopia," https://time.com/6240993/facebook-meta-ethiopia-lawsuit/.

[20] E. Laanpere Liina, "Online Hate Speech: Hate or Crime?," *ELSA Int. Online Hate Speech Compet.*, pp. 1–5, 2017, [Online]. Available: https://files.elsa.org/AA/Online_Hate_Speech_Essay_Competition_runner_up.pdf

[21] binnymathew, punyajoys saisakethaluru, "Deep_Learning_Models_for_Multilingual_Hate_Speech_," *https://aclanthology.org/Q17-1010/*.

[22] M. A. G. S. Muhammad Usman, "A Large Language Model-Based Approach for Multilingual Hate Speech Detection on Social Media," *https://www.researchgate.net/publication/392560200_Multilingual_Hate_Speech_Detection_in_Social_Media_Using_Translation-Based_Approaches_with_Large_Language_Models*.

[23] J. M. Moguerza and A. Muñoz, "Support vector machines with applications," *Stat. Sci.*, vol. 21,

no. 3, pp. 322–336, Aug. 2006, doi: 10.1214/088342306000000493.

[24] B. Jang, M. Kim, G. Harerimana, S. U. Kang, and J. W. Kim, "Bi-LSTM model to increase accuracy in text classification: Combining word2vec CNN and attention mechanism," *Appl. Sci.*, vol. 10, no. 17, Sep. 2020, doi: 10.3390/app10175841.

[25] P. Joshi, S. Santy, A. Budhiraja, K. Bali, and M. Choudhury, "The State and Fate of Linguistic Diversity and Inclusion in the NLP World," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 6282–6293. doi: 10.18653/v1/2020.acl-main.560.

[26] A. Schmidt and M. Wiegand, "A Survey on Hate Speech Detection using Natural Language Processing," *Soc. 2017 - 5th Int. Work. Nat. Lang. Process. Soc. Media, Proc. Work. AFNLP SIG Soc.*, no. 2012, pp. 1–10, 2017, doi: 10.18653/v1/w17-1101.

[27] United Nations Secretary-General António Guterres, "Hate Speech The UN Strategy and Plan of Action," https://www.un.org/en/hate-speech/un-strategy-and-plan-of-action-on-hate-speech.

[28] J. Waldron, *The Harm in Hate Speech*. Harvard University Press, 2012. doi: 10.4159/harvard.9780674065086.

[29] A. C. Neal, "United Nations Universal Declaration of Human Rights (1948) 1," in *Fundamental Social Rights at Work in the European Community*, 2019. doi: 10.4324/9780429458460-10.

[30] G. Van Bueren, "International Covenant on Civil and Political Rights 1966," in *International Documents on Children*, 2024. doi: 10.1163/9789004638662_013.

[31] Council of Europe, "Recommendation No. R (97) 20 of the Committee of Ministers To Member States on 'Hate Speech,'" *Counc. Eur.*, no. 91, 1997.

[32] Federal Negarit Gazette, "Proclamation no. 1185 /2020 hate speech and disinformation prevention and suppression proclamation," *Fed. Negarit Gaz.*, vol. 26, no. 26, p. 12339, 2020, Accessed: Mar. 15, 2026. [Online]. Available: https://digitalpolicyalert.org/event/27551-ethiopian-parliament-adopts-hate-speech-and-disinformation-prevention-and-suppression-proclamation-no-11852020-including-content-moderation-regulation

[33] A. M. Kaplan and M. Haenlein, "Users of the world, unite! The challenges and opportunities of Social Media," *Bus. Horiz.*, vol. 53, no. 1, pp. 59–68, Jan. 2010, doi: 10.1016/j.bushor.2009.09.003.

[34] Naveen Kumar, “Facebook Users Statistics (2026) – Global Data & Growth Trends,” https://www.demandsage.com/facebook-statistics/.

[35] United Nations Human Rights Council, “Human Rights Council Thirty-ninth session 10-28 September 2018 Agenda item 4 Human rights situations that require the Council’s attention Report of the independent international fact-finding mission on Myanmar Summary.”

[36] D. Tadesse and M. Assefa, “THE IMPLICATION OF THE HATE SPEECH LAW ON FREEDOM EXPRESSION IN ETHIOPIA.”

[37] A. L. Tonja *et al.*, “Natural Language Processing in Ethiopian Languages: Current State, Challenges, and Opportunities,” in *Proceedings of the Fourth workshop on Resources for African Indigenous Languages (RAIL 2023)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2023, pp. 126–139. doi: 10.18653/v1/2023.rail-1.14.

[38] Adem Jemal, “An Assessment of Hate Speech, Social Media and Violence in Ethiopia: The Case of Facebook and Youtube,” Addis Ababa University.

[39] B. Wondie, E. Tadesse, and T. Yirdaw, “Amharic Language Hate Speech Detection on Social Media,” *Am. J. Artif. Intell.*, vol. 9, no. 1, pp. 16–21, May 2025, doi: 10.11648/j.ajai.20250901.12.

[40] D. Appleyard, “BOOK REVIEW A Comparative Dictionary of the Agaw Languages,” 2006. [Online]. Available: http://scholarlypublishingcollective.org/msup/neas/article-pdf/13/2/225/943200/nortafristud.13.2.0225.pdf

[41] the free encyclopedia Wikipedia, “Wag Hemra Zone,” https://www.wikiwand.com/en/Wag_Hemra_Zone. Accessed: Mar. 11, 2026. [Online]. Available: https://www.wikiwand.com/en/Wag_Hemra_Zone

[42] J. Tsujii, “Lifetime Achievement Award Natural Language Processing and Computational Linguistics,” 2021, doi: 10.1162/COLI.

[43] the free encyclopedia Wikipedia, “Geʿez script,” https://en.wikipedia.org/wiki/Ge%CA%BDez_script.

[44] C. D. T. S. Jill Burstein, “Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers),” Association for Computational Linguistics. Accessed: Mar. 11, 2026. [Online]. Available: https://aclanthology.org/N19-1/

[45] C. Themeli, G. Giannakopoulos, and N. Pittaras, “A study of text representations in Hate Speech Detection,” Feb. 2021, [Online]. Available: http://arxiv.org/abs/2102.04521

[46] H. D. Abubakar and M. Umar, “Sentiment Classification: Review of Text Vectorization Methods: Bag of Words, Tf-Idf, Word2vec and Doc2vec,” *SLU J. Sci. Technol.*, vol. 4, no. 1&2, pp. 27–33, Aug. 2022, doi: 10.56471/slujst.v4i.266.

[47] the free encyclopedia Wikipedia, “Bag-of-words model,” https://en.wikipedia.org/wiki/Bag-of-words_model.

[48] Abebawu Yigezu, “Amharic Word Embedding,” https://abe2g.github.io/Am_FastText.html. Accessed: Mar. 12, 2026. [Online]. Available: https://abe2g.github.io/Am_FastText.html

[49] Fatih Karabiber, “TF-IDF — Term Frequency-Inverse Document Frequency,” https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/.

[50] N. T. Molla and S. Ming, “AMorph: An End-to-End Morpheme Level Natural Language Processing Pipeline for Amharic,” *OALib*, vol. 13, no. 02, pp. 1–18, 2026, doi: 10.4236/oalib.1114897.

[51] T. M. Ababu, M. M. Woldeyohannis, and E. B. Getaneh, “Bilingual hate speech detection on social media: Amharic and Afaan Oromo,” *J. Big Data*, vol. 12, no. 1, p. 30, Feb. 2025, doi: 10.1186/s40537-024-01044-y.

[52] A. Eshetu, G. Teshome, and T. Abebe, “Learning Word and Sub-word Vectors for Amharic (Less Resourced Language),” *Int. J. Adv. Eng. Res. Sci.*, vol. 7, no. 8, pp. 358–366, 2020, doi: 10.22161/ijaers.78.39.

[53] M. M. Truşcă, “Efficiency of SVM classifier with Word2Vec and Doc2Vec models,” *Proc. Int. Conf. Appl. Stat.*, vol. 1, no. 1, pp. 496–503, Oct. 2019, doi: 10.2478/icas-2019-0043.

[54] P. J., “A Detailed Study of Deep Learning using Convolutional Neural Network Approach,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 8, no. 11, pp. 831–837, Nov. 2020, doi: 10.22214/ijraset.2020.32306.

[55] A. Jacovi, O. S. Shalom, and Y. Goldberg, “Understanding Convolutional Neural Networks for Text Classification,” 2018.

[56] “Keras: CNNs With Conv1D For Text Classification Tasks,”

https://coderzcolumn.com/tutorials/artificial-intelligence/keras-cnn-with-conv1d-for-text-classification. Accessed: Mar. 13, 2026. [Online]. Available: https://coderzcolumn.com/tutorials/artificial-intelligence/keras-cnn-with-conv1d-for-text-classification

[57] Ravindu Senaratne, "Text Classification Using Bi-Directional LSTM," https://heartbeat.comet.ml/text-classification-using-bi-directional-lstm-ca0070df7a81. Accessed: Mar. 13, 2026. [Online]. Available: https://heartbeat.comet.ml/text-classification-using-bi-directional-lstm-ca0070df7a81

[58] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu, "Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling."

[59] L. Mai, D. Li, and Z. Zhao, "Text sentiment analysis and classification based on bidirectional long and short term memory networks," *Appl. Comput. Eng.*, vol. 77, no. 1, pp. 183–189, Jul. 2024, doi: 10.54254/2755-2721/77/20240689.

[60] S. M. Gashe, S. M. Yimam, and Y. Assabie, "Hate Speech Detection and Classification in Amharic Text with Deep Learning," pp. 2012–2014, 2024, [Online]. Available: http://arxiv.org/abs/2408.03849

[61] "Stacked Bidirectional LSTM (BiLSTM)," https://www.emergentmind.com/topics/stacked-bidirectional-long-short-term-memory-bilstm. Accessed: Mar. 13, 2026. [Online]. Available: https://www.emergentmind.com/topics/stacked-bidirectional-long-short-term-memory-bilstm

[62] "BERT Model - NLP," https://www.geeksforgeeks.org/nlp/explanation-of-bert-model-nlp/. Accessed: Mar. 13, 2026. [Online]. Available: https://www.geeksforgeeks.org/nlp/explanation-of-bert-model-nlp/

[63] A. L. Tonja *et al.*, "EthioLLM: Multilingual Large Language Models for Ethiopian Languages with Task Evaluation," Jun. 2024, [Online]. Available: http://arxiv.org/abs/2403.13737

[64] G. Gebremeskel Bahru, "Tigrinya Hate Speech Detection and Classification from Facebook Posts and Comments Using Deep Learning Approaches."

[65] N. Ibrahim, F. Mulford, and R. Batista-Navarro, "Large Language Models as Detectors or Instigators of Hate Speech in Low-resource Ethiopian Languages," 2025. [Online]. Available: https://huggingface.co

[66] N. Keshari, D. Malladi, and U. Mittal, “Hate Speech Detection Using Natural Language Processing Stanford CS224N Custom Project.”

[67] G. Mihret and J. Joshi[2], “Peer-Reviewed, Multidisciplinary & Multilingual Journal THE AGEW PEOPLES OF ETHIOPIA: AN INTERDISCIPLINARY STUDY OF THEIR HISTORICAL LEGACY, LINGUISTIC HERITAGE, CULTURAL IDENTITY, AND CONTEMPORARY CHALLENGES”, [Online]. Available: http://vidyajournal.org

[68] N. Ibrahim, F. Mulford, M. Lawrence, and R. Batista-Navarro, “Resources for Annotating Hate Speech in Social Media Platforms Used in Ethiopia: A Novel Lexicon and Labelling Scheme,” 2024. [Online]. Available: https://github.com/

[69] Z. Mossie and J.-H. Wang, “Social Network Hate Speech Detection for Amharic Language,” 2018. doi: 10.5121/csit.2018.80604.

[70] A. L. Tonja, O. Kolesnikova, A. Gelbukh, and J. Kalita, “EthioMT: Parallel Corpus for Low-resource Ethiopian Languages,” *5th Work. Resour. African Indig. Lang. RAIL 2024 Lr. 2024 - Work. Proc.*, vol. 2024, pp. 107–114, 2024.

[71] K. R. Mabokela, M. Primus, and T. Celik, “Advancing sentiment analysis for low-resourced african languages using pre-trained language models,” *PLoS One*, vol. 20, no. 6 JUNE, Jun. 2025, doi: 10.1371/journal.pone.0325102.

[72] A. Gandhi *et al.*, “Hate speech detection: A comprehensive review of recent works,” Aug. 01, 2024, *John Wiley and Sons Inc*. doi: 10.1111/exsy.13562.

[73] P. Kapil and A. Ekbal, “A Unified Multi task Learning Architecture for Hate Detection Leveraging User-based Information.”

[74] M. Abusaqer, J. Saquer, and H. Shatnawi, “Efficient Hate Speech Detection: Evaluating 38 Models from Traditional Methods to Transformers,” in *ACMSE 2025 - Proceedings of the 2025 ACM Southeast Conference*, Association for Computing Machinery, Inc, May 2025, pp. 203–213. doi: 10.1145/3696673.3723061.

[75] “Support Vector Machine (SVM) Algorithm,” https://www.geeksforgeeks.org/machine-learning/support-vector-machine-algorithm/. Accessed: Mar. 15, 2026. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/support-vector-machine-algorithm/

[76] Z. Abebaw, A. Rauber, and S. Atnafu, “Hate Speech Detection from Transliterated Amharic

Social Media Comments Using Machine Learning and Deep Learning Approaches," *J. Comput. Sci. Data Anal.*, vol. 01, no. 2, pp. 59–75, Dec. 2024, doi: 10.69660/jcsda.01022404.

[77] "mcnemar: McNemar's test for classifier comparisons," https://rasbt.github.io/mlxtend/user_guide/evaluate/mcnemar/. Accessed: Mar. 15, 2026. [Online]. Available: https://rasbt.github.io/mlxtend/user_guide/evaluate/mcnemar/

[78] E. Alshdaifat, F. Hussein, A. Al-shdaifat, M. Al-Hassan, and E. Altarawneh, "A hybrid model for handling the imbalanced multiclass classification problem," *IAES Int. J. Artif. Intell.*, vol. 14, no. 5, p. 3982, Oct. 2025, doi: 10.11591/ijai.v14.i5.pp3982-3993.

[79] L. Floridi *et al.*, "AI4People—An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations," *Minds Mach.*, vol. 28, no. 4, pp. 689–707, Dec. 2018, doi: 10.1007/s11023-018-9482-5.

[80] "I (Legislative acts) REGULATIONS REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)."

[81] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A Survey on Bias and Fairness in Machine Learning," Jan. 2022, [Online]. Available: http://arxiv.org/abs/1908.09635

[82] I. D. Raji *et al.*, "Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing," in *FAT* 2020 - Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, Association for Computing Machinery, Inc, Jan. 2020, pp. 33–44. doi: 10.1145/3351095.3372873.

[83] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," 2002.

[84] "Introduction to Information Retrieval," https://nlp.stanford.edu/IR-book/. Accessed: Apr. 04, 2026. [Online]. Available: https://nlp.stanford.edu/IR-book/

[85] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 2013.

[86] J. S. Sepp Hochreiter, "LONG SHORTTERM MEMORY".

[87] M. Schuster and K. K. Paliwal, “Bidirectional Recurrent Neural Networks,” 1997.

[88] F. Pedregosa FABIANPEDREGOSA *et al.*, “Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot,” 2011. [Online]. Available: http://scikit-learn.sourceforge.net.

[89] M. Grandini, E. Bagli, and G. Visani, “Metrics for Multi-Class Classification: an Overview,” Aug. 2020, [Online]. Available: http://arxiv.org/abs/2008.05756

[90] T. Joachims, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features.”

[91] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, “A Survey on Bias and Fairness in Machine Learning,” Apr. 30, 2022, *Association for Computing Machinery*. doi: 10.1145/3457607.

[92] M. Hardt, E. Price, and N. Srebro, “Equality of Opportunity in Supervised Learning,” Oct. 2016, [Online]. Available: http://arxiv.org/abs/1610.02413

[93] P. Burnap and M. L. Williams, “Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making,” *Policy and Internet*, vol. 7, no. 2, pp. 223–242, Jun. 2015, doi: 10.1002/poi3.85.

[94] A. Magueresse, V. Carles, and E. Heetderks, “Low-resource Languages: A Review of Past Work and Future Challenges,” Jun. 2020, [Online]. Available: http://arxiv.org/abs/2006.07264

[95] U. Naseem, I. Razzak, S. K. Khan, and M. Prasad, “A Comprehensive Survey on Word Representation Models: From Classical to State-of-the-Art Word Representation Language Models,” *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 20, no. 5, Sep. 2021, doi: 10.1145/3434237.

ANNES

ANNEX A

Himtagne Writing System Reference Tables

Table A.1 Primary Himtagne Base Consonants

No	Character	Phonetic Value
1	ሀ	Hä
2	ለ	Lä
3	ሐ	ḥä
4	መ	Mä
5	ሠ	Śä
6	ረ	Rä
7	ሰ	Sä
8	ሸ	Šä
9	ቀ	Qä
10	በ	Bä
11	ተ	Tä
12	ቸ	Čä
13	ኀ	ẖä
14	ነ	Nä
15	ኘ	Ñä
16	አ	’ä
17	ከ	Kä
18	ኸ	Xä
19	ወ	Wä

20	ዘ	Zä
21	ዐ	’ä
22	ዠ	Žä
23	የ	Yä
24	ደ	Dä
25	ጀ	Ǧä
26	ገ	Gä
27	ኘ	Na
28	ጠ	ṭä
29	ጨ	Č̣ä
30	ጰ	Ṗä
31	ጸ	ṣä
32	ፀ	ṣ́ä
33	ፈ	Fä
34	ፐ	Pä
35	ኧ	Ng
36	ቐ	Q

Table A.2 Extended and Labiovelar Forms

Extended forms

ሏ, ሗ, ሟ, ሧ, ሯ, ሷ, ቿ

ቆ, ቢ, ቪ, ቷ, ቾ

ኍ, ኗ, ኟ, ኧ, ኵ

ዟ, ዧ, ዷ

ጇ, ጎ, ጧ, ጯ, ፏ, ፗ

ፏ, ፲, ፼

Labiovelar series

Series	Characters
Kw	ኰ, ኩ, ኳ, ኴ, ኵ
Hw	ጐ, ጕ, ጓ, ጔ, ጕ
Xw	ኸ, ኹ, ኻ, ኼ, ኽ
Gw	ገ, ጉ, ጋ, ጌ, ጉ
Qw	ቈ, ቍ, ቋ, ቌ, ቍ

Table A.3 Phonetic Equivalent Character Shapes

Base Sound	Example Letter	Alternative Forms
Ha	ሀ	ሃ, ሐ, ሓ, ኀ, ኃ
Se	ሠ	ሰ
A	አ	ዐ, ኣ
Tse	ጸ	ፀ

These redundant forms create challenges for computational processing because different characters may represent the same phonetic sound.

Table A.4 Abbreviation Forms Used in Himtagne Text

Abbreviation	Full Form (English)	Himtagne
E.C.	Ethiopian Calendar	ዓመተ ስቁረት
School	School	ክንድ ቻን
Office	Office	ሰራሸነ ቻን
Mrs	Miss	ዊዝር

These abbreviation forms are important for preprocessing because abbreviated expressions frequently appear in social media text.

ANNEX B

Dataset Annotation Guideline

The annotation guideline is written in Himtagne and includes examples of hate speech from the annotation website. The website was created only for labeling purposes.

The texts were collected from social media platforms such as Facebook. Annotators must read each message carefully and decide whether it is hating speech or normal speech.

If the message is hating speech, it should be classified based on the group targeted in the content. Annotators should read the definitions and examples carefully to ensure accurate and consistent labeling.

Definitions

Speech refers to the act of sharing information through spoken words, written text, images, or any other form of communication.

Hate speech is language that expresses hatred toward a specific person or group. It is used to insult, humiliate, or degrade individuals based on characteristics such as race, religion, political, gender.

Hate speech is not identified only by the use of certain words. Instead, the meaning and intention of the message must be understood by considering the context in which the words are used.

According to the Federal Negarit Gazeta, hate speech is defined as speech that intentionally promotes hatred, discrimination, or attacks against a person or identifiable group based on ethnicity, religion, race, gender, or disability.

Hate Speech Criteria

A speech is classified as hate speech when it targets a specific individual or a group of people who share a common identity, such as ethnicity, religion, race, gender, or other protected characteristics.

It becomes hate speech if the message has the potential to spread, encourage, or promote violence against the targeted person or group. This includes the use of insulting or harmful words intended to cause damage or encourage harm.

A text is also considered hate speech if it supports or promotes hateful statements, hateful online posts, or extremist organizations. Even if the message does not directly call for violence, supporting hateful ideas can still fall under hate speech.

Indirect expressions such as idioms, metaphors, coded language, or symbolic words that carry harmful or violent meaning toward a group are also classified as hate speech. Context and hidden meaning must be carefully analyzed.

The message is considered hate speech if it includes violent communication, racial or sexist slurs, or degrading language aimed at minorities. Attacking, silencing, or unfairly criticizing a minority group without evidence or logical argument is also categorized as hate speech.

Furthermore, hate speech includes promoting discrimination, spreading false claims about a minority group, misrepresenting facts to damage their reputation, negatively stereotyping them, or defending ideas such as xenophobia or sexism.

Data Collection Rules

During data collection, several important rules were followed to maintain dataset quality. First, all collected texts were required to be written in the Himtagne language. Not every collected text contained hate speech; some texts were neutral or normal conversations. The dataset focused only on hate speech related to specific targets such as race, religion, gender, and political views. Any hate speech that did not belong to these categories was excluded from the dataset.

Texts that were unclear, confusing, or ambiguous were removed to ensure high-quality data. It is also important to note that abusive language does not always mean hate speech. However, if abusive words were directed toward a specific group of people, careful checking was required before assigning labels.

Hate Speech Labeling Rules

Before annotating hate speech into its subcategories, annotators must carefully evaluate the text using several guiding questions. These questions help ensure accurate labeling and improve dataset reliability. The annotator should first check whether the text contains hateful or abusive words. They should also examine whether the text encourages negative actions against a person or group, or if it suggests or directly expresses threats of violence.

In addition, annotators should check whether the text uses insulting or humiliating language toward individuals or communities. Texts that promote discrimination, such as calls for eviction, killing, or other forms of physical violence against a group, must be carefully identified. Another important consideration is whether the text spreads misinformation that is known to be false but may cause public fear, disturbance, or conflict.

The annotator must also determine whether the text promotes discrimination by denying rights or attacking people based on ethnicity, religion, gender, disability, or other protected characteristics. The general context of the speech should also be examined to determine whether it targets or categorizes specific groups as hate targets. Furthermore, the annotator should check if the text is intentionally written to cause harm, promote dangerous ideas, or encourage violence.

Other important factors include checking whether the text contains violent communication, racial or sexist slurs, or language that attacks minority groups. The text should also be evaluated to see if it attempts to silence minority voices or criticize minority groups without strong or reasonable arguments. In some cases, texts may promote violent crimes directly or indirectly, even without using explicit hate speech terms.

Annotators should also analyze whether the text uses misleading arguments, such as straw man reasoning, to attack minority groups. Additionally, they should check for negative stereotyping, xenophobic attitudes, or sexist expressions. Finally, the annotator must consider whether the text claims to be supported by trusted sources, contains shocking or fear-inducing false information, or promotes harmful social conflict.

Hate Speech Categorization and Labeling Scheme

Hate speech was also classified into different categories during annotation. Each annotator was responsible for labeling sentences according to their appropriate group. The main hate speech categories included ethnic or racial hate speech, religious hate speech, gender-based hate speech, political hate speech, disability-related hate speech, socioeconomic hate speech, normal speech, and texts that could not be categorized.

The racial or ethnic hate speech category included statements that show hatred, discrimination, or promote violence against individuals or groups based on race, ethnicity, nationality, skin color, language, or cultural background. Any text that contained offensive expressions targeting these characteristics was placed in this category.

The religious hate speech category included texts that express hatred or discrimination against people based on their religion or religious practices. This also included hateful statements about religious places, religious symbols, religious ceremonies, or religious materials such as images and physical objects related to religion.

The gender hates speech category contained texts that express discrimination or violence against individuals based on their gender or sexual orientation. Any message that promotes gender-based violence, discrimination, or negative stereotypes about gender groups was classified under this category.

The political hate speech category included statements that express hatred or discrimination toward individuals or groups because of their political beliefs, political activities, or political leadership. Most political hate speech was directed toward political leaders, political parties, or political supporters.

ANNEX C

Text Preprocessing

Text Normalization

In the first step, character normalization was applied to standardize different forms of letters into a unified representation. This mapping converts visually or semantically similar characters into a single standard form to reduce spelling variations caused by dialectal differences, typing habits, or informal writing styles. Normalization helps improve model performance by reducing vocabulary size and ensuring consistent text representation across the dataset.

```
# Normalize letters
mapping = {
    "ሐ":"ሀ","ሑ":"ሁ","ሒ":"ሂ","ሓ":"ሃ","ሔ":"ሄ","ሕ":"ህ","ሖ":"ሆ",
    "ኀ":"ሀ","ኁ":"ሁ","ኂ":"ሂ","ኃ":"ሃ","ኄ":"ሄ","ኅ":"ህ","ኆ":"ሆ",
    "ሠ":"ሰ","ሡ":"ሱ","ሢ":"ሲ","ሣ":"ሳ","ሤ":"ሴ","ሥ":"ስ","ሦ":"ሶ",
    "ዐ":"አ","ዑ":"ኡ","ዒ":"ኢ","ዓ":"ኣ","ዔ":"ኤ","ዕ":"እ","ዖ":"ኦ",
    "ፀ":"ጸ","ፁ":"ጹ","ፂ":"ጺ","ፃ":"ጻ","ፄ":"ጼ","ፅ":"ጽ","ፆ":"ጾ"
}
text = ''.join(mapping.get(ch, ch) for ch in text)
```

Removal of URLs, Mentions, Emojis, and Special Characters

Next, unnecessary elements such as URLs, user mentions, emojis, hashtags, numbers, Latin characters, and punctuation symbols were removed using regular expressions. These elements generally do not contribute meaningful semantic information for hate speech detection and may introduce noise into the dataset. Their removal helps clean the text, reduce irrelevant patterns, and improve feature extraction quality.

Whitespace Cleaning

After removing unwanted characters, extra whitespace was eliminated by replacing multiple spaces with a single space and trimming leading and trailing spaces. This step ensures consistent token spacing, improves tokenization accuracy, and enhances overall text readability.

```
# Remove URLs, mentions, emojis
text = re.sub(r'https?://\S+|www\.\S+', '', text)
text = reg.sub(r'\p{Emoji}', '', text)
text = re.sub(r'@\w+', '', text)
text = re.sub(r'#(\w+)', r'\1', text)
text = re.sub(r'[A-Za-z0-9]', '', text)
text = re.sub(r"[።፡፤፥፦፧፨]", '', text)
text = re.sub(r'\s+', ' ', text).strip()
```

Abbreviation Expansion

Common abbreviations were expanded into their full word forms using a predefined dictionary. This step helps preserve semantic meaning and avoids misinterpretation caused by shortened or informal expressions. Expanding abbreviations ensures that words are correctly interpreted by the model and improves the quality of textual representations.

```
# Expand abbreviations
abbreviations = {
    'ዓ.ሰ.': 'ዓመተ ሰቱረት',
    'ክ/ሃን': 'ክንድ ሃን',
    'ሰ/ሃን': 'ሰራሽነ ሃን',
```

Stopword Removal (While Preserving Negations)

Stopwords that carry minimal semantic meaning were removed to reduce noise and dimensionality in the dataset. However, negation words were intentionally preserved, as they play a critical role in altering sentence meaning and sentiment. This selective stopword removal improves feature relevance while maintaining important contextual information.

```
# Stopwords removal (keep negations)
stopwords = {
    'ዝመ','የሻ','ዊኑ','ሃን','በ','ወደ','ከ','ለ','የ',
    'ክት','እንቸን','እነሱ','ይና','ያን','ያ','እንዛይ','ላውላውድ','ብሎቅ',
```

Negation Handling

Negation terms were explicitly maintained to ensure that sentences expressing denial, rejection, or opposition were not misclassified. This step is particularly important for sentiment analysis and hate speech detection, as negation can reverse the meaning of statements and affect classification outcomes.

```
negations = {
    'አይ','አየውም','አርቀቅም','እምቢይር','ቸለቅም','ሰራሽውም',
```

Simple Stemming

Finally, simple stemming was applied to reduce words to their base or root forms by removing common suffixes. This process helps minimize vocabulary size and ensures that variations of the same word are treated consistently. Stemming improves generalization and enhances model learning efficiency.

```
# Simple stemming
tokens = [re.sub(r'ም$', '', w) for w in tokens if len(w) > 1]
```