



JIMMA UNIVERSITY
JIMMA INSTITUTE OF TECHNOLOGY
FACULTY OF COMPUTING AND INFORMATICS
DATA SCIENCE CHAIR

**DETECTION AND IDENTIFICATION OF HARASSMENT AND HATE SPEECH ON
SOCIAL MEDIA BASED ON PROTECTED CHARACTERISTICS FOR THE AFAAN
OROMO LANGUAGE USING DEEP LEARNING.**

By

ASMELASH G/EYESUS

**A FINAL THESIS SUBMITTED TO DATA SCIENCE CHAIR, FACULTY OF COMPUTING AND
INFORMATICS, JIMMA INSTITUTE OF TECHNOLOGY, JIMMA UNIVERSITY, FOR PARTIAL
FULFILLMENT OF THE AWARD OF MASTER'S IN DATA SCIENCE.**

PRINCIPAL ADVISOR: GELETAW SAHLE (PhD)

CO-ADVISOR: HAMBISA MITIKU (Ass.Prof)

SUBMITTED TO

DECEMBER 2023

DATA SCIENCE CHAIR

JIMMA, ETHIOPIA

Declaration

I hereby declare that this Master Thesis entitled “**Detection and Identification of Harassment and Hate Speech on Social Media Based on Protected Characteristics for the Afaan Oromo Language Using Deep Learning**” is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

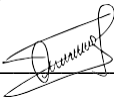
Asmellash G/yesus

Name of Student

Signature

Date

Dr. Geletaw Sahle



Major Advisor

Signature

Date

Mr. Hambisa Mitiku

Co-Advisor

Signature

Date

Acknowledgment

First of all, I would like to thank the almighty god who made possible, this Thesis possible. My gratitude and appreciation also go to my advisors **Dr. Geletaw Sahle** and **Mr. Hambisa Mitiku** for giving constructive comments, and suggestions in conducting this Thesis. Lastly, I would like to thank the Jimma University Institute of Technology, and the data science chair, for coordinating and giving all the necessary information and support.

Table of contents

| | |
|---|-----|
| List of Figures | i |
| List of Tables | ii |
| Acronyms and Abbreviations | iii |
| Abstract..... | iv |
| CHAPTER ONE..... | 1 |
| 1.1 Introduction | 1 |
| 1.2 Statement of the problem..... | 4 |
| 1.3 Research question | 5 |
| 1.4 Objective..... | 5 |
| 1.4.1 General objective | 5 |
| 1.4.2 Specific objective..... | 5 |
| 1.5 Significance of the study | 6 |
| 1.6 Scope and limitation | 6 |
| 1.7 Structure of the Thesis | 7 |
| 2 CHAPTER TWO..... | 8 |
| LITERATURE REVIEW AND RELATED WORKS..... | 8 |
| 2.1 Hate Speech | 8 |
| 2.1.1 Definition of Hate Speech..... | 8 |
| 2.2 Social Media | 10 |
| 2.3 Hate Speech on Social Media..... | 10 |
| 2.4 Techniques Used for Hate Speech Detection and harassment identification | 11 |
| 2.4.1 Methods of machine learning..... | 11 |
| 2.4.2 Deep learning methods | 12 |
| 2.5 The benefit of comparing Multiple algorithm | 15 |
| 2.6 Methods for feature extraction | 15 |
| 2.6.1 Bag of words..... | 15 |
| 2.6.2 Term frequency-inverse document frequency (TF-IDF) | 16 |
| 2.6.3 Word Embedding..... | 16 |
| 2.6.4 Continuous Bag of Words (CBOW)..... | 17 |

| | | |
|-------|---|----|
| 2.7 | Afaan oromo language..... | 18 |
| 2.7.1 | The writing system of Afaan Oromo | 18 |
| 2.7.2 | Word categories of afaan oromo..... | 18 |
| 2.7.3 | Afaan Oromo Sentence Structure | 19 |
| 2.7.4 | Afaan Oromo Punctuation | 19 |
| 2.8 | Related Works on Hate Speech Detection..... | 20 |
| 3 | CHAPTER THREE..... | 27 |
| | RESEARCH METHOD | 27 |
| 3.1 | Data Collection | 27 |
| 3.1.1 | Dataset Annotation Guide line..... | 29 |
| 3.1.2 | Dataset description..... | 30 |
| 3.2 | Research Design | 31 |
| 3.3 | Data Preprocessing | 32 |
| 3.4 | Feature Extraction Methods..... | 33 |
| 3.4.1 | Bag of words | 33 |
| 3.4.2 | TF-IDF (Term Frequency-Inverse Document Frequency) | 33 |
| 3.4.3 | Word Embedding | 34 |
| 3.5 | Model Selection Techniques | 34 |
| 3.6 | Deep Learning Algorithms for the Harassment Identification | 35 |
| 3.6.1 | Recurrent Neural Network (RNN)..... | 35 |
| 3.6.2 | Long Short-Term Memory (LSTM) | 36 |
| 3.6.3 | Bidirectional Long Short-Term Memory (BiLSTM)..... | 37 |
| 3.6.4 | Gated recurrent unit (GRU) | 37 |
| 3.6.5 | Convolutional Neural Network (CNN)..... | 38 |
| 3.7 | BERT Pre-trained Model..... | 39 |
| 3.8 | Advantages of deep learning over classical machine learning | 40 |
| 3.9 | Model Evaluation Techniques | 40 |
| 3.9.1 | Cross-Validation | 40 |
| 3.9.2 | Evaluation Metrics | 41 |
| 3.10 | Natural language processing..... | 42 |
| 3.11 | Model Overfit Handling Techniques | 42 |
| 3.12 | Hyperparameter techniques | 42 |
| 3.13 | Ethics of the Research | 43 |

| | | |
|--------|--|----|
| 3.14 | The Proposed Architecture | 43 |
| 3.15 | Proposed Deep Learning Model | 44 |
| 3.16 | BERT Pre-Trained Language Model..... | 45 |
| 3.17 | Proposed Feature Representation | 47 |
| 3.17.1 | Word Embeddings | 47 |
| 3.18 | Saving the Model for Future Use..... | 48 |
| 3.19 | Tools | 48 |
| 3.19.1 | Data preparation and preprocessing tools | 48 |
| 3.19.2 | Package managers and environments | 49 |
| 3.19.3 | Modeling tools and packages..... | 49 |
| 3.20 | Hardware Tools | 51 |
| 4 | CHAPTER FOUR | 52 |
| | RESULTS AND DISCUSSION..... | 52 |
| 4.1 | Result of Approach-1..... | 52 |
| 4.1.1 | Results of the CNN model | 52 |
| 4.1.2 | Results of the LSTM model..... | 53 |
| 4.1.3 | Results of the BiLSTM model | 54 |
| 4.1.4 | Results of the GRU model | 55 |
| 4.2 | Result of Approach-2..... | 56 |
| 4.2.1 | Results of BERT Pre-trained Model | 57 |
| 4.2.2 | Results of CNN Pre-trained Model..... | 58 |
| 4.3 | Hyperparameter Tuning..... | 59 |
| 4.4 | Discussions | 60 |
| 5 | CHAPTER FIVE..... | 63 |
| | CONCLUSION AND FUTURE WORKS | 63 |
| 5.1 | Conclusion..... | 63 |
| 5.2 | Future Works | 63 |
| | Reference | 64 |
| 1 | APPENDICES..... | 70 |
| | Appendix 1: Sample Code..... | 70 |
| 1.1 | Loading the dataset..... | 70 |

| | | |
|-------|---|----|
| 1.1 | Text Normalization..... | 70 |
| 1.2 | Sample code of data cleaning | 71 |
| 1.3 | Sample code of text tokenization..... | 71 |
| 1.4 | Sample code of word2vec implementation | 72 |
| 1.5 | Sample of List of stop words in afaan oromo Language..... | 73 |
| 1.6 | Sample Code for all Proposed Model..... | 74 |
| 1.6.1 | CNN Model Sample Code | 74 |
| 1.6.2 | LSTM model sample code | 75 |
| 1.6.3 | BiLSTM model sample code | 75 |
| 1.6.4 | GRU model sample code | 76 |
| 2 | Appendix 2: BERT Pre-trained Model..... | 77 |
| 2.1 | Sample code for BERT Pre-trained Model..... | 77 |
| 2.2 | Sample of Embedding Particular words in word2vec Representation | 77 |
| 2.3 | Sample code for saving the Model | 78 |
| 2.4 | Sample code for Hate speech Detection and Harassment Classification | 78 |
| 2.5 | Sample form data collection using Google forms | 79 |

List of Figures

| | |
|--|----|
| Figure 2-1: Artificial Neural Networks..... | 12 |
| Figure 2-2: Recurrent neural network..... | 13 |
| Figure 2-3: Architecture of Bidirectional LSTM..... | 14 |
| Figure 2-4: The architecture of CBOW | 17 |
| Figure 2-5: Skip-gram Model Architecture | 18 |
| Figure 3-1: Dataset preparation procedure | 28 |
| Figure 3-2: Research Design workflow | 32 |
| Figure 3-3: The Architecture of the RNN model..... | 36 |
| Figure 3-4: The LSTM Model's architecture. | 37 |
| Figure 3-5: The GRU Model's Architecture..... | 38 |
| Figure 3-6: The Architecture of CNN Model | 39 |
| Figure 3-7: Bert Pre-trained model and fine-tuned architecture..... | 39 |
| Figure 3-8: The General architecture of Hate Speech Detection and harassment identification Model | 44 |
| Figure 3-9: The proposed architecture for hate speech detection and harassment identification model..... | 46 |
| Figure 3-10: Proposed Data Preparation, Preprocessing, and Word Embedding Techniques..... | 47 |
| Figure 4-1: CNN Model Accuracy and Loss through Epochs..... | 52 |
| Figure 4-2: LSTM Model Accuracy and Loss through Epochs..... | 53 |
| Figure 4-3: BiLSTM Model Accuracy and Loss through Epochs | 54 |
| Figure 4-4: GRU Model Accuracy and Loss through Epochs..... | 55 |
| Figure 4-5: BERT Pre-trained model Accuracy and Loss through Epochs..... | 57 |
| Figure 4-6: CNN with pre-trained model Accuracy and Loss through Epochs..... | 58 |

List of Tables

| | |
|---|----|
| Table 2-1 : Afaan Oromo word class | 19 |
| Table 2-2: Afaan Oromo Language Punctuation Mark | 20 |
| Table 2-3: Summary of binary class hate speech detection using machine learning approaches..... | 23 |
| Table 2-4: Summary of Multi-class hate speech detection using Machine learning Approaches..... | 24 |
| Table 2-5: Summary of Binary and Multi-class Hate Speech Detection using Deep Learning Approaches. | 25 |
| Table 2-6: Afaan Oromoo Hate Speech Detection Related Work..... | 26 |
| Table 3-1: Selected Pages | 29 |
| Table 3-2: Dataset Description | 30 |
| Table 4-1: Summary result of each model | 56 |
| Table 4-2: Hyperparameter for all Proposed Models | 59 |

Acronyms and Abbreviations

| | |
|---------|--|
| AI | Artificial Intelligence. |
| BERT | Bidirectional Encoder Representation from Transformers |
| BiLSTM | Bi-Directional Long Short-Term Memory. |
| BLR | Beaconless Routing. |
| BOW | Bag of Word. |
| CNN | Convolutional Neural Network. |
| GRU | Gated Recurrent Unit. |
| LSTM | Long Short-Term Memory. |
| NB | Nave Bayes. |
| NLP | Natural language processing. |
| NLTK | Natural language Toolkit. |
| RFDT | Random Forest Decision Tree. |
| RF | Random Forest. |
| RNN | Recurrent Neural Network. |
| SVM | Support Vector Machine. |
| TF-IDF | Term Frequency Inverse Document Frequency. |
| XGBoost | Extreme Gradient Boosting. |
| QA | Question and Answering |
| LSVM | Linear Support Vector Machine |

Abstract

Social media today affects a nation's social, political, and economic facets in both positive and negative ways. Positive effects include the facilitation of digital opinion exchanges and the rapid and broad dissemination of information. The spread of hate speech, which includes disparaging individuals based on shared traits like gender (sexism), race, religion, color, disability, and nationality, has a negative effect. Protected characteristics are defined as being against the law to discriminate against someone because of gender (sexism), race, religion, color, disability, or nationality. The use of social media platforms, like Facebook and Twitter, to organize hateful events and spread hate speech has become more common. The unstructured nature of social media data makes manual tracking more challenging. Thus, we are motivated to continue developing the detection of hate speech and harassment identification based on protected characteristics. The study aims to develop a method for harassment and hate speech detection and identification on social media based on protected characteristics of the Afaan Oromo language using deep learning. In this study, we have used an experimental research design approach. Facepacer and Google Forms were used for data collection. Normalization, data cleaning, and tokenization were utilized for data preprocessing. We employed two-step approaches for the experimentation. The primary dataset was used for experimentation using the BERT-pretrained model. To examine and identify the best performing deep learning techniques in our dataset, a convolutional neural network (CNN), long short-term memory (LSTM), bi-directional long short-term memory (BiLSTM), and gated recurrent unit (GRU) were used and executed. However, overfitting was encountered due to the limited size of our dataset. To address the overfitting issue within the dataset, methods of cross-validation and L2 regularization were employed. To solve the scarcity of the trained data, the second approach, the BERT-pretrained model, was applied. The researcher used the model's accuracy and loss to evaluate the performance of the model. After all the preprocessing activities and training were performed, the performance of each model was: a convolutional neural network (CNN) with an accuracy of 98.44% and a loss of 0.0396 and a bidirectional encoder representation from transformers (BERT) with an accuracy of 98.83% and a loss of 0.0952. Finally, through experimentation, the BERT model outperformed other algorithms with 98.83% accuracy. The study used Afaan Oromo language features to detect harassment and hate speech on social media. Future research could use social media data to create unique word embeddings and assess the CapsNet model's effectiveness on non-textual data.

Keywords: BERT, Deep learning, Harassment, Hate speech, and Protected characteristics

CHAPTER ONE

1.1 Introduction

People's engagement with social media has grown over the past few decades, and advancements in mobile computing and the internet have led to an increase in social media usage [4]. Web-based and mobile-based Internet applications that facilitate the production, exchange, and availability of widely accessible user-generated content are referred to as social media[1]. The use of social media platforms like Facebook and Twitter to organize and spread hate speech has grown. However, hate speech is spreading on social media these days, disrupting the social lives of most people due to posts containing hate speech and convicts who incite such posts [2].

Social media can have both beneficial and detrimental effects on a nation's political, social, and economic spheres[2]. Positive effects include the facilitation of digital opinion exchanges and the rapid and widespread dissemination of information. The spread of hate speech, which includes disparaging individuals based on protected traits like gender (sexism), race, religion, color, disability, and nationality, has a detrimental effect [3]. In this study, protected characteristics are defined as being against the law to discriminate against someone because of gender (sexism), race, religion, color, disability, and nationality [4]. The rise in social media usage in all contemporary societies has fundamentally altered interpersonal interactions[5].

Contrarily, harassment is defined as any undesired or unwanted behavior, such as upsetting, frightening, obnoxious, or threatening a person or a group. It entails consistent or recurrent acts that put the target(s) in a hostile, intimidating, or offensive situation. There are many different ways that harassment can happen, but some examples are as follows: (i) verbal harassment, which consists of calling someone names, threatening them, or using derogatory, offensive, or abusive language; (ii) physical harassment, which consists of unwanted physical acts like pushing, hitting, or other physical aggression. (iii) Unwelcome sexual advances, requests for sexual favors, and other verbal, nonverbal, or physical acts of a sexual character that incite hostility or discomfort are all considered forms of sexual harassment. (iv) Cyber harassment is when someone uses social media, email, or messaging apps to send offensive or threatening messages, start rumors, stalk someone, or engage in other online harassment; (v) Psychological or emotional harassment is when someone engages in actions that denigrate, threaten, or otherwise harass someone emotionally. Constant criticism, embarrassment, loneliness, or emotionally distressing threats are a few examples of this[6]. All things considered, it's crucial to remember that different jurisdictions may have different laws and definitions of harassment and that some legal definitions may only apply in particular situations. Generally speaking, harassment is defined as unwelcome and damaging behavior directed towards a person or group

and resulting in an uncomfortable or hostile atmosphere. Policies and guidelines are frequently implemented by organizations and societies to address and prevent harassment, fostering an environment that is safe and courteous for everyone. Promote abusive behavior, and the large user base of platforms has the ability to rapidly disseminate harmful content. Language nuances and cultural differences can make it difficult to identify hate speech, and content moderation is a complex and resource-intensive process. Memes or coded language are two ways that offenders evade moderation and detection. Self-harm, depression, and anxiety are psychological effects. Different laws and regulations across international borders create legal and jurisdictional challenges. It can be challenging to strike a balance between the need to stop harassment and the right to free speech. In order to overcome these obstacles, platforms, governments, civil society organizations, and users must work together. They also need to implement strict content moderation policies, improve technology, offer support, and encourage digital literacy.

According to the global conflict trackers, 5.1 million Ethiopians were internally displaced in 2021 alone, setting a record for the greatest number of internally displaced persons in a single year in any nation at the time. In addition, thousands of people fled to Sudan and other nearby nations. Approximately 600,000 people had died as a result of the Tigray War and the ensuing humanitarian crisis by the time the Pretoria agreement went into effect. For the first time since November 2020, humanitarian organisations were allowed to operate in Tigray beginning in late 2022.

The situation in Ethiopia was getting worse, and it was happening at the same time as the conflict in Tigray. Following a string of violent attacks against ethnic Oromo residents, the government of Amhara State declared a state of emergency early in the conflict. While militants from Amhara and Afar, two regions that border Tigray, were accused of aiding federal troops and even attacking civilians they suspected to be Tigrayans or connected to the TPLF, Oromia's regional army sided with the Tigrayans in the civil war.

Fearing that the TPLF's rising influence would pose a threat to the state, Ethiopia detained more than 4,000 people in Amhara in May 2022 to undermine a nationalist militia that assisted the government in repelling the group. The following month, hundreds of Amhara people were killed in Oromia, and government forces were accused of failing to act appropriately.

According to the Minority Rights Group International (MRG) highlights evidence of ethnic cleansing and denounces the recent acts of violence, intimidation, and harassment directed towards minorities in Ethiopia's Oromia region.

Recently, hate speech and fake news have been held accountable in Ethiopia, particularly for ethnic violence that has occurred throughout the nation[2]. Facebook and Twitter are widely used social media platforms [2]. More than 36 million people (33.8% of the Ethiopian population) speak the Afaan Oromo language [7].

Facebook and Twitter users in Ethiopia use a variety of languages to spread hate speech, which is supported by Facebook and leads to deadly ethnic conflicts between people through ugly Facebook content[8]. Additionally, according to the Ethiopian government, interactions on social media intensify hate speech and hinder the nation's progress[9]. Ethiopia's government keeps an eye on social media posts to stop offensive messages from being spread. The country has also seen disruptions in internet service, including the blocking of websites and their inaccessibility[9]. Additionally, proposes a bill to expand anti-terrorism laws to include online hate speech. The bill forbids the distribution of terrorizing messages and imposes a maximum eight-year prison sentence on offenders[10].

Furthermore, A harassment law is a speech restriction imposed by the government that forbids the expression of certain opinions or that awards significant compensatory and punitive damages for speech that violates someone's race, color, religion, sex, nationality, or disability[11].

Hate speech detection has been an increasingly trending subject over the past few years. Several studies have been conducted to address this issue, including [12] for English Languages, [13] for Italian Languages, [15] for Amharic Languages, and [16] for Chinese Hate Speech Detection.

Currently, social media offers localization, enabling users to use various world languages on their websites. Among these languages is Afaan Oromoo, which is also the working language of the Oromia regional state and one of the most widely spoken languages. The Afaan Oromoo language still lacks many computing tools and inadequate resources. Widespread hate speech and acts of violence against people or groups in Oromia are motivated by protected traits such as gender, race, religion, color, disability, and nationality[14]. To address the hate speech detection challenge of the Afaan Oromoo language, a hate speech framework was proposed using a support vector machine, logistic regression, and decision tree [8]. For feature extraction, bag of word, TF_IDF, word embedding, Bert, and feature selection techniques such as frequency-based feature selection, and chi-square feature selections were employed and achieved an accuracy of 96 % with the support vector machine algorithm. Increasing the amount of data, preparing another dataset rather than text, and adding several classes, are mentioned by the researcher for further improvement of the research. The G. O. Ganfure [15] additionally experimented with five deep learning model approaches, including CNN, LSTM, GRU, BILSTM, and CNN-LSTM, for comparative hate speech detection for the afaan oromo language was conducted. using those algorithms, a model that detects afaan oromo text into four classes was developed, and applying classical ensembles and meta-learning tasks for improvement, and misclassification of the text were also mentioned as challenging issues by the authors. Furthermore, [19] tried to detect hate speech from social media using machine learning approaches, and the linear support vector classifier scored the highest f1-score value of 64 %.

Moreover, (T. M. Ababu and M. M. Woldeyohannis) [9] attempted to create a model that can identify and categorize hate speech as race, religion, gender, and offensive classes. The classical and ensemble machine learning algorithms SVM perform with 0.82% accuracy, while the deep learning algorithm BiLSTM achieves better performance with 0.84% accuracy. However, most of the research that has been done only detects the text as hate, normal, neutral, and offensive. Not only that most of the researchers did not mention what types of hyperparameter tuning and overfitting handler techniques have applied.

Therefore, using a deep learning approach, this study attempted to identify hate speech texts and harassment on social media based on protected characteristics, such as gender (sexism), race, religion, color, disability, and nationality. The novelty of the this study is that the dataset was collected from scratch by the researcher and the number of classes also were increased from the previous performed activities , and applying of the transfer learning methods. Twelve label datasets have been gathered, processed, and annotated for this study.

1.2 Statement of the problem

On social media, hate speech, harassment, and threats of violence are directed towards a person, community, or group based on protected traits like gender (sexism), race, religion, color, disability, or nationality[7]. Hatred and derogatory speech on social media is a widespread issue that harms communities, particularly for underserved languages like the Afaan Oromo language, and fuels conflict and violence in the community[2]. Furthermore, A proclamation for the prevention of hate speech was posted by the Ethiopian government, requiring social media companies should remove texts that promote hate speech from their platforms [7]. This is due to the fact that hate speech on social media has the capacity to greatly disrupt society.

The majority ethnic group in Ethiopia, the Oromo, speaks Afaan Oromo as their first language. Other Ethiopian ethnic groups also speak it as a first or second language [7]. This largest ethnic group in Ethiopia expresses opinions and feelings about socioeconomic and political issues via Afaan Oromo on social media. Disagreement, agreement, and clashing of individual or group ideas occur during this process. When they disagree, it becomes problematic because they are using hate speech as a weapon against one another. There is nothing wrong with agreement. The majority of languages worldwide continue to lack adequate resources and devices for handling languages, a situation that is particularly severe for nations in sub-Saharan Africa. Because it lacks the same tools and techniques for language processing as other languages like English, the Afaan Oromoo language is considered under-resourced. Because word formation and grammatical arrangement vary among languages, there are differences in hate speech detection and harassment identification techniques. For the Afaan Oromoo Language, some research has been done on hate speech text detection on social media. [19] classified the text into binary classes as hate or normal, while (S. G. Tesfaye and K. K. Tune) [8] classed as hate or neutral. Whereas (T. M. Ababu and M. M. Woldeyohannis) [9] were classified into four areas, such as gender, race, religion, and offensive. However, hate speech is propagated

based on nationality; color and disability were not considered [9]. In addition, hyperparameter techniques were not applied to handle overfitting and improve accuracy [9]. Dues to this reason, this study is focused on detecting hate speech and harassment identification based on the six protected characteristics such as gender (sexism), race, religion, color, disability, and nationality.

As social media users, we saw that many people in Ethiopia have made posting hate speech, particularly against particular ethnic groups, a daily habit[16]. Overall, despite their efforts to create AI for hate speech identification, social media corporations such as Facebook and Twitter continue to face difficult challenges. This is because social media text hate speech detection and harassment identification remain tedious tasks that for hate speech to be taken down from a social media platform, the user must report it to the relevant social media companies. Additionally, the unstructured nature of social media data makes manual tracking more challenging. Thus, we are motivated to continue developing the detection of hate speech and harassment identification based on protected characteristics. In addition, we must contribute to the development of the system for identifying harassment and hate speech, as well as to the preparation of benchmark datasets for use by researchers in the future [17].

1.3 Research question

The research was answer the following question

1. Which feature extraction technique is better to use for the detection and identification of hate speech and harassment on social media based on protected characteristics?
2. Which algorithm will be best suited for developing the detection and identification of hate speech and harassment on social media based on protected characteristics?
3. How do we improve the overall performance of detection and identification of hate speech and harassment on social media based on the protected characteristics model?

1.4 Objective

1.4.1 General objective

The general objective of this research was to develop a model that detects hate speech and identifies harassment on social media based on protected characteristics for the Afaan Oromo language using deep learning.

1.4.2 Specific objective

To achieve the general objectives, the following specific tasks were performed: -

- ✓ To prepare hate speech and harassment dataset based on protected characteristics using afaan Oromo texts.
- ✓ To select the better feature extraction technique, used for detection and identification of hate speech and

harassment on social media based on protected characteristics.

- ✓ To identify the best-suited algorithm for developing the detection and identification of harassment and hate speech on social media based on protected characteristics.
- ✓ To build a model that detects hate speech and harassment identification based on protected characteristics.
- ✓ To evaluate the performance of the developed model and tune parameters for the selected algorithm for further improvement.

1.5 Significance of the study

The study's implications are multifaceted. First, to keep online communities safe for their members, it's critical to recognize hate speech and harassment. The development of technologies for social media harassment and hate speech detection based on Afaan Oromo language-protected characteristics will be crucial in making the task of tracking online hate speeches easier. To guarantee the nation's long-term security and tranquility. In their day-to-day operations, social media platforms can use it as a control mechanism to recognise hate speech and harassment based on protected characteristics. However, it also aids in shielding users of social media from hate speech both during and after their time on these platforms. Hateful texts can be blocked by using it as part of their extension. By blocking hateful posts targeted at particular groups or people, the hate speech detection system is essential to maintaining the nation's values of democracy and dignity. Additionally, it helps businesses on social media sites like Facebook and Twitter.

1.6 Scope and limitation

Protected characteristics: It is illegal to treat someone unfairly based on their gender (sexism), race, religion, color, nationality, handicap, or other factors. [4]. This study focused on the Hate speech concept given by Facebook and Twitter social media platforms and the Ethiopian government Proclamation [4]. Hate speech is defined by the Ethiopian House of People Representative as expressions that deliberately incite hatred, discrimination, and attacks against a person or group of people because of their gender, race, ethnicity, religion, or disability. It also covers suppression, harassment, and disinformation avoidance [18]. Facebook and Twitter define hate speech as verbally attacking someone on the basis of one or more protected characteristics, such as race, national origin, ethnicity, sexual orientation, religion, sex, serious illness or disability, gender, gender identity, or caste. Only textual Facebook posts and comments written in the Afaan Oromo language are included in this study because there is no well organized dataset to deal with dataset like video, picture and other image datasets. It is restricted to hate speech text detection and harassment identification on the social media platforms Facebook and Twitter based on a protected characteristics model. Using face pager software, a new dataset was gathered from Afaan Oromo text posts and comments

on popular public Facebook and Twitter pages between January 2021 and April 2023. The posts and comments were then annotated into 12 classes. Videos, pictures, or emoji gestures are not considered in this study.

1.7 Structure of the Thesis

The research papers is structured into five chapters:

The first chapter, which was previously discussed, includes the study's introduction, the statement of problems, the research questions , objective ,scope and limitations, and significance.

Chapter 2 literature review and related works on the subject cover the following topics: the definition of hate speech, the definition of social media, the techniques for detecting hate speech (feature extraction, deep learning models, an overview of the Afaan Oromo languages, and finally, the related work).

Chapter 3 the third chapter covers the research methodology, dataset construction techniques, detection model development techniques, feature extraction techniques, deep learning algorithms, assessment techniques, the suggested method for identifying and detecting hate speech and harassment in the Afaan Oromo language, the deep learning model's architecture, feature representation, and system architecture and tools used in the study.

In Chapter 4 the results of experimenting with different deep learning models are presented in Chapter 4. Additionally, this chapter covers and analyses the experiment's primary findings.

Chapter 5: In this chapter, we wrap up the study and provide some essential or helpful context for the important suggestions, Recommendations, and next steps.

2 CHAPTER TWO

LITERATURE REVIEW AND RELATED WORKS

To explore the research problem and give a deeper understanding of the concept, this chapter reviews pertinent literature. The review includes definitions of hate speech, a list of application domains for hate speech detection systems, and details on the newest methods of hate speech detection. First, let's define social media and hate speech. The chapter then moves on to discussing hate speech on social media, including how to identify hate speech, a brief history of the Afaan Oromo language, and relevant research on the subject of identifying harassment and hate speech on social media.

2.1 Hate Speech

The phrase "hate speech" has many definitions provided by various national laws, as well as definitions provided by social media companies. Any form of expression that incites hatred or makes personal attacks against a target due to their identity is referred to as hate speech. However, there's no universally accepted definition of what hate speech is. As a result, it is challenging to determine whether or not texts contain hate speech. Even for humans, some obstacles make identifying hate speech challenging. As a result, even though the dataset contains fewer instances of hate speech than all other social media data, its creation is nonetheless noteworthy. Hate speech is content created by users in a variety of writing styles[19]. Hate speech has become an increasingly serious crime in recent times, particularly in online communications. Because of social media, the internet, and people's growing willingness to express their opinions online, hate speech is spreading swiftly [20].

2.1.1 Definition of Hate Speech

2.1.1.1 Definition of Social Media Platforms

Distinct social media networks define hate speech differently. However, the majority of definitions share a similar set of elements. Social media sites like Facebook, Twitter, and YouTube are the most popular places for hate speech to be posted and distributed. Below is a definition of hate speech as provided by Twitter, Facebook and YouTube:

Definition 1: Twitter, "A person's race, ethnicity, national origin, sexual orientation, gender, gender identity, religious affiliation, age, disability, or serious illness are not grounds for encouraging violence against, directly attacking, or threatening other people. Additionally, accounts that primarily aim to incite

harm towards others based on these categories are prohibited by us.”

Definition 2: Facebook, “Content deemed objectionable is that which targets people on the basis of protected characteristics, including but not limited to race, ethnicity, national origin, religion, sexual orientation, caste, sex, gender, gender identity, or a serious illness or disability. Additionally, we provide some immigration status protections.. Attacks are defined as calls for exclusion or segregation, statements of inferiority, or violent or dehumanizing speech.”

Definition 3: YouTube, “On YouTube, hate speech is not permitted. Age, disability, ethnicity, gender, nationality, race, immigration status, religion, sex, sexual orientation, or veteran status are just a few of the characteristics that cause us to remove content that incites violence or hatred toward specific people or groups. We use the definitions of gender, sex, and sexual orientation while keeping in mind that societal perceptions of these concepts are constantly changing.”

2.1.1.2 Definition from Other Sources

Furthermore, there exist definitions of "hate speech" from multiple organisations, a few of which are mentioned below:

Definition 4: UN, International Committee on the Elimination of Racial Discrimination, “Hate speech is defined as any expression of intolerance-based hatred, such as xenophobia, anti-Semitism, or racial hatred, that is promoted, supported, or encouraged. This includes prejudice and animosity towards minorities, migrants, and people of immigrant origin, as well as intolerance manifested through strong nationalism and ethnocentrism.”[21].

Definition 5: American Convention on Human Rights (ACHR), “The goal and the target of hate speech can both be used to define it. As far as intent goes, hate speech is any speech that is intended to oppress, intimidate, or inspire violence or hatred. Additionally, the speech must be directed specifically at an individual or group based on attributes such as race, religion, nationality, gender, sexual orientation, disability, or any other attribute shared by that group”[22]. We employed the Ethiopian Hate Speech and Disinformation Prevention and Suppression Proclamation's definition of hate speech for this study No.1185 /2020[23]. The following is the definition:

Hate Speech: “Hate speech is defined as any intentional incitement of hatred, discrimination, or attacks on a recognised identity or group of people on the basis of race, ethnicity, gender, religion, or disability.”.

The following characteristics of hate speech set it apart from other types of speech, according to the definitions given above:

- ✓ The goal of hate speech is to stir up animosity or violence.

- ✓ Encouraging prejudice against a person or group on the basis of their identities, such as gender, religion, or handicap.
- ✓ Hate speech targets particular characteristics that a person or group possesses, like gender, race, or religion.
- ✓ A protected identity can be disparaged or attacked through hate speech.

2.2 Social Media

People use social media as a forum to voice their concerns and ideas. Social media platforms facilitate the sharing and exchange of ideas, information, texts, images, videos, and much more among users within a specific network.[24].These days, social media greatly aids in the simplification of tasks like online business, email, and tutorials or education. Social media has drawbacks as well as benefits that may have a detrimental impact on society.The possibility that social media platforms offer for individuals to publish and disseminate various illegal information could be detrimental to social media users as well as other societies. Hate speech and misinformation are being disseminated via social media sites like Facebook, Twitter, and YouTube, which directly affects society's daily functioning.

2.3 Hate Speech on Social Media

People can participate online and create and share content thanks to social media's nature. People now have the chance to post and express their thoughts and emotions online thanks to it. People can post and distribute unlawful content, including hate speech, cyberbullying, and offensive speech, by taking advantage of this opportunity. These internet resources are frequently abused and misused to disseminate information that can denigrate or attack specific individuals or groups, or that encourages violence or hate crimes against them. The built-in security and privacy features of social media platforms enable their users to hide their true identities behind a screen and express or spread hateful content more than they otherwise could[16].

Recently, several national authorities have deemed hate speech to be a serious issue. Hate speech on social media harms an inclusive, egalitarian society in addition to its users' well-being. These days, billions of people are connected through social media sites like Facebook and Twitter, which enable them to instantly share their thoughts and opinions. However, there are also several frightening repercussions, including hate speech, cyberbullying, trolling, and online harassment[25].

With more people using social media in Ethiopia, hate speech on these platforms is growing to be a major issue. The problem is made more difficult by the absence of legislation or recommendations that define or address hate speech indirectly. However, there is a law—the anti-terrorism law—that is utilized inadvertently

about matters of hate. The use of social media and other communication platforms to spread terrorizing messages is prohibited by law as is " The dissemination of any frightening information or lewd message via any kind of communication device or network" [25].

Punishing offenders with a maximum eight-year prison term. Nonetheless, the law has been applied to suppress speech or messages that are critical of public officials or policies. The academic community, as well as national and international organizations, have responded to this law because it violates human rights legislation about freedom of speech. Politicians, government employees, and legislators in Ethiopia are currently aware of hate speech on social media and are attempting to resolve the problem . New laws against hate speech and fake news are being drafted by the nation's lawmakers and will soon go into effect.

2.4 Techniques Used for Hate Speech Detection and harassment identification

Since text classification tasks are related to the problems of harassment identification and hate speech detection, many researchers are utilizing machine learning and deep learning techniques to address the issue of online hate speech detection.

2.4.1 Methods of machine learning

Machine learning enables computers to gain knowledge and experience from data and carry out tasks expertly. For machine learning algorithms to learn, data is necessary, and data discipline and database discipline must be connected. Algorithms that can adapt and learn from large data sets are crucial because of the nature of some problems and the exponential growth of digital information. Tasks that are too complex to program are the two main issues that require machine learning algorithms due to their capacity to learn and improve from experience. The other issue is that adaptability necessitates a working knowledge of user data. To better understand the data and assist users in making decisions for their daily activities, machine learning algorithms look for patterns in the data. Machine learning can be divided into two categories: supervised and unsupervised.

Supervised Learning: Labeled data is used to train this kind of machine learning. It makes use of a labeled dataset made up of matching sets of observed inputs (X) and their corresponding outputs (Y). To determine the patterns between the inputs and outputs, the machine learning algorithm is applied to the dataset.

Unsupervised Learning: In the case of an unsupervised approach, we must let the model work on its own to find information rather than using labeled data. A dataset that only contains inputs and analyses to make sense of, organize, or group the data is used to train a model. By giving the data in the group some structure, it makes recommendations to users based on unlabeled and uncategorized data. It uses input data without output data to draw determinations from the dataset. However, because it lacks labeled outputs, its objective is to infer the inherent structure found in a collection of data points[26].

2.4.2 Deep learning methods

Neural networks are used in deep learning techniques to automatically extract multiple layers of features from the provided data. It is a subset of machine learning techniques that aims to discover the inputs' layered model. It allows data representations with multiple abstraction levels to be learned by computational models made up of various processing layers. Deep learning algorithms use a backpropagation algorithm to update their internal parameters, which allows them to find complex structures in large datasets[25]. Text classification tasks appear to be a promising domain for neural network models. Deep learning models have additional depth but still rely solely on the ANN [27].

2.4.2.1 Neural networks

Neural networks, or artificial neural networks, are pattern-recognition models fascinated by the ideas behind how the human brain works. Its mode of operation appears to be similar to how neurons in the human brain communicate with one another. Deep learning algorithms are based on artificial neural networks (ANNs), a subset of machine learning. An input layer, an output layer, and one or more hidden layers make up an artificial neural network (ANN).

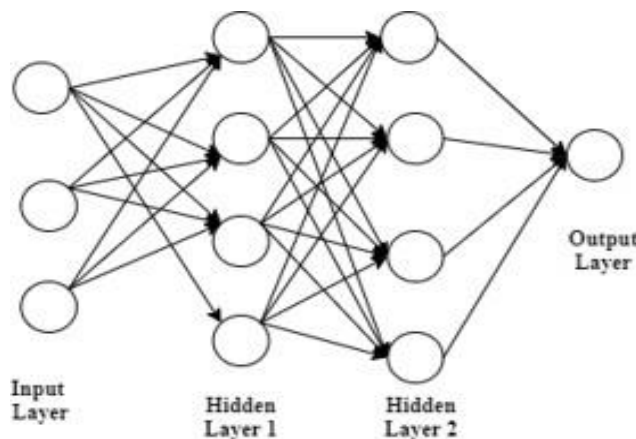


Figure 2-1: Artificial Neural Networks

Because of their high accuracy, deep learning algorithms have recently gained a lot of attention in the text classification task. For text classification, the following deep-learning algorithms are employed:

2.4.2.2 Recurrent neural network (RNN)

Sequential or time-series data are used by an ANN type known as an RNN; the output from one step is used in the current one. Because of its design, it can recognize patterns in the data that indicate sequential features and use those patterns to predict the next event. As a result, RNNs ought to have a hidden state where they can remember details about a sequence and store some of that information.

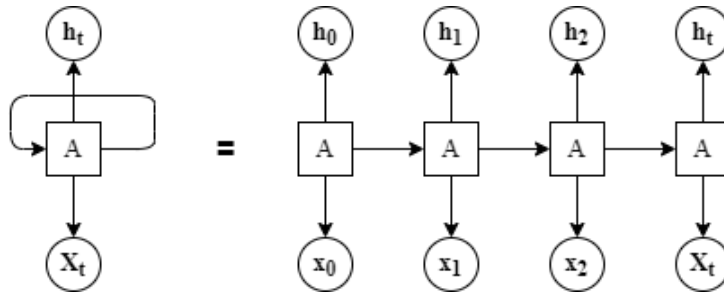


Figure 2-2: Recurrent neural network

The hidden layer of an RNN is a recurrent layer because all of the neurons in it are connected. The input to the hidden layer comes from both the input layer (x_t) and the hidden layer from the previous state (h_{t-1}). Unlike feed-forward neural networks, which accept fixed-size vectors as input and generate fixed-size vector outputs, recurrent neural networks are designed to model sequences and have the ability to recall past data. Short-term memory is a challenge for recurrent neural networks (RNNs), though. They will struggle to transfer the data from the earlier timesteps to the later ones if the input sequence is very long. This issue was resolved by GRU and LSTM.

2.4.2.3 Long-short Term Memory (LSTM)

Hochreiter & Schmidhuber (1997) introduce LSTMs. Long-term dependencies can be learned by short-term memory, which is a unique type of RNN network. Long-term dependency is a problem that LSTMs are meant to avoid. The input gate i_t , the output gate o_t , and the forget gate f_t are the three gates of an LSTM. The first stage in the Long Short-Term Memory (LSTM) process is the forget gate (f_t), which determines what data from the cell state to discard by examining h_{t-1} and x_t . For each number in the cell state, it then outputs a number between 0 and 1, where 1 means "keeping this completely" and 0 means "ignore this." The first step is to choose which data to store in the cell state. Selecting which values to update is then the responsibility of the input gate layer. Then, the tanh layer generates a vector of fresh candidates. The new cell state is then c_t , which was previously the old state c_{t-1} . Since the output C_t will be a filtered version, it is ultimately dependent on the cell state. The decision of which cell component to output is made by the

sigmoid layer.

2.4.2.4 Bidirectional long short-term memory

BiLSTMs are an extension of standard LSTMs designed to capture data on a sequential dataset while preserving contextual features from the past and future. This network uses two sub-layers to process input sequences in both directions to account for the entire input context. The two RNN sub-layers handle the computation of the forward and backward hidden sequences[28]. The output sequence is then calculated by combining these. It can take advantage of context in two ways. The forward and backward hidden layers of BiLSTM, which can recognize both the prior and subsequent contexts, are combined. However, LSTM is limited to using historical context. The following illustrates the BiLSTM architecture:

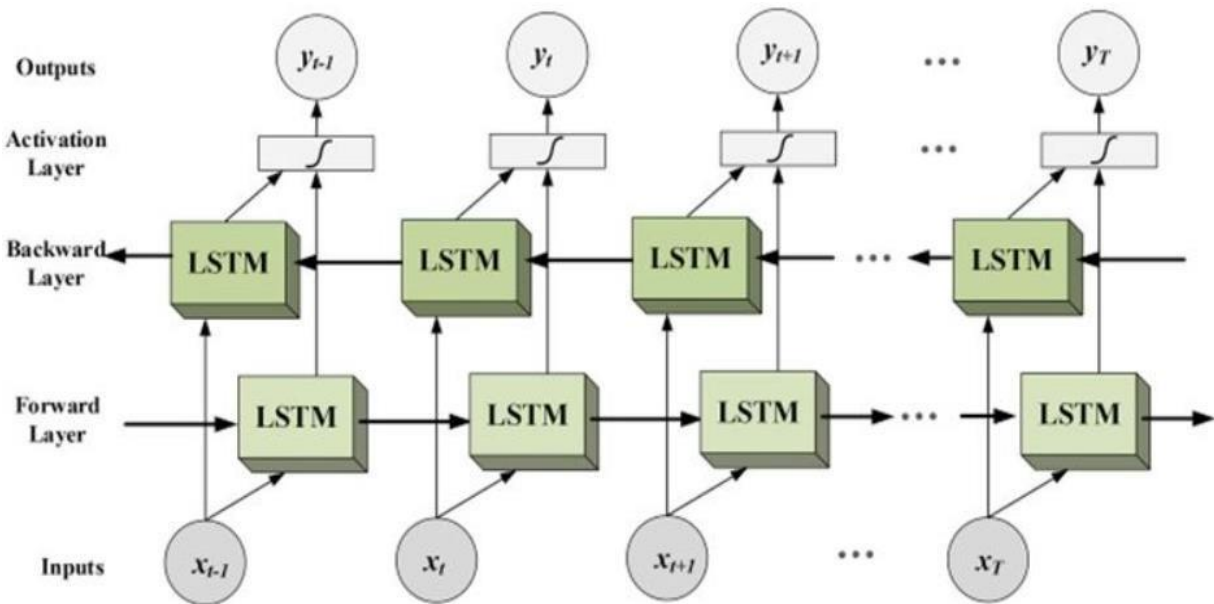


Figure 2-3: Architecture of Bidirectional LSTM

2.4.2.5 Convolutional Neural Network (CNN)

The connections between nodes in a convolutional neural network do not form a cycle because it operates using a "feed-forward" strategy. The model is primarily utilized in computer vision applications, but it has also demonstrated encouraging outcomes in a range of natural language processing tasks. Originally designed for image processing, convolution layers allow CNNs to take advantage of the 2D structure of image data by having discrete computation units respond to discrete areas of the input. Nonetheless, CNN has recently drawn a lot of attention for its text classification.

To use the model for text data, the first words in a sentence are transformed into low-dimensional word vectors using various methods, like word embedding. CNN consists of three layers: input, convolution, and max-pooling. Words are embedded into low-dimensional vectors using the input layer. Subsequently, the convolution layer operates by performing convolutions over the embedded word vectors with various filter sizes. However, in text classification modeling, a good prediction requires an understanding of the data's context. For instance, for the model to function well on the new test data, it must comprehend the word contexts from the training data in the case of hate speech detection. Nevertheless, because the dataset is represented by one-hot encoded vectors and the model is a feed-forward neural network, RNNs are better at modeling text sequences, while CNN is not able to model the context of words.

2.5 The benefit of comparing Multiple algorithm

Several models are employed in hate speech detection and harassment identification, including CNN, LSTM, BiLSTM, GRU, and BERT. CNN speeds up training and inference times by identifying local patterns and features in textual data. Long-range dependencies and sequential information are captured by LSTM networks, which improves gradient flow during training. BiLSTM offers a thorough representation of the input sequence by fusing the advantages of LSTM with contexts from the past and future. GRU models are computationally efficient because they capture long-range dependencies and require fewer parameters. BERT improves hate speech detection performance by capturing fine-grained contextual information and semantic relationships after being pre-trained on large-scale corpora[15].

2.6 Methods for feature extraction

2.6.1 Bag of words

A text representation that shows word locations within a document is called a "bag of words." In contrast to dictionaries, which offer a predetermined list of words, this technique builds a vocabulary list using the terms found in the training set. The procedures for developing a BOW model for a text are as follows:

1. Store the tokens in a list and convert the text to token form.
2. Build a lexicon with the tokens.
3. Determine how many times each token appears in a sentence, then record the total.
4. By converting the text into vectors and counting each word in the vocabulary, you can create a bag of words model.

Nevertheless, the word sequence in this method ignores its syntactic and semantic content in addition. This implies that misclassification could happen if the words are used in various contexts. The model of the bag of words uses a sentence's word count as its representation. Thus, the BOW model does not take sentence structure into account.

2.6.2 Term frequency-inverse document frequency (TF-IDF)

The computation of a word's relevance to a text within a corpus or series is known as TF-IDF.

Term Frequency (TF): Frequency, the number of times the given word t appears in the document (given document d). As a result, it makes sense that when a word appears in the text, it becomes more relevant.

$$TF(t) = \frac{\text{(Number of times term } t \text{ appears in a document)}}{\text{(Total number of terms in the document)}} \quad \text{Eq.2-1}$$

Inverse Document Frequency (IDF): This IDF's primary function is to determine the word's relevance. Finding the relevant records that meet the demand is the main goal of the search. Since TF gives all terms equal weight, Utilizing term frequencies is not the only way that gauge a term's importance within the document.

$$IDF(t) = \text{Log} \frac{\text{(Total number of documents)}}{\text{(Number of document with term } t \text{ in it)}} \quad \text{Eq. 2-2}$$

Term frequency is only the number of times a term appears in a single document and depends on the entire corpus, whereas document frequency is the total number of unique documents that contain a term. Now let's look at the definition of the frequency of the inverse paper. The total number of documents in the corpus, divided by textual frequency, is known as the IDF.

Part-of-speech tag (POS): It is a method for enhancing the significance of the context and identifying a word's function within a sentence. It entails identifying the word's class, such as determiners, verb base forms, adjectives, present tense singular present verbs, and personal pronouns. In general, none of the methods mentioned above can fully convey the meaning and context.

2.6.3 Word Embedding

By capturing the contextual hierarchy, real-number vectors are mapped to words in the vocabulary using a feature-learning technique called word embedding. It is a method of representing texts in which terms with the same meaning are represented similarly. This indicates that two similar words are represented in a vector

space by nearly identical vectors that are positioned very near to one another. These are necessary to resolve the majority of issues with natural language processing[29]. Because it uses a coordinate system to represent words, related words—mostly those with a relationship to one another in the corpus—are positioned closer to one another. Among the most widely used methods for learning word embeddings is Word2Vec, which was created at Google in 2013 by Tomas Mikolov. Word embeddings became the newest thing in natural language processing after the great interest in the field was sparked by the release of the word2vec toolkit. The Word2vec model creates a collection of vectors known as feature vectors that represent the words in a text corpus as input. The main goal of Word2vec is to enable words with similar contexts to have similar embeddings. Word2vec provides two models for architecture: The Skip-gram model and the CBOW (Continuous Bag of Words) model.

2.6.4 Continuous Bag of Words (CBOW)

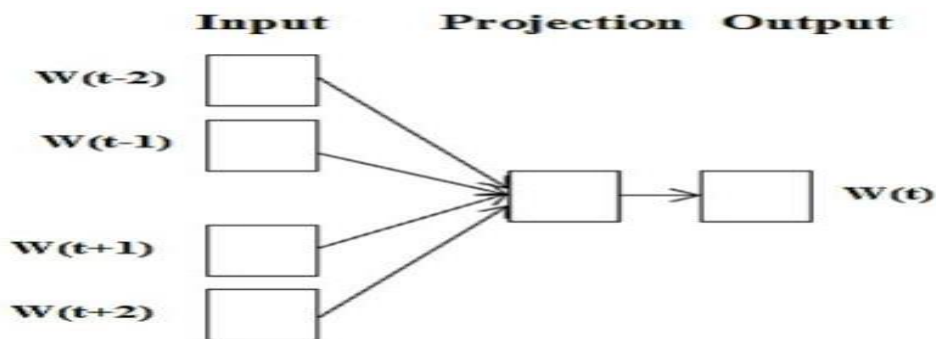


Figure 2-4: The architecture of CBOW

The likelihood of a word occurring in a given context is usually ascertained by the CBOW, as illustrated in Figures 2-4 above. Because of this, it applies to all possible combinations of words and context. It predicts a word's probability of occurring by looking at the words around it. The representation of a word is determined by the words surrounding it, which aids in capturing the semantics of the word. The output layer of the architecture holds the vector representation of that word after the model has been trained over the average vectors of words that accompany it.

Skip-gram model: The model typically seeks to achieve the CBOW model's opposite. It predicts the words present in the source context when a target word is entered. It can be inferred that the target is fed into the input, and the output layer is replicated multiple times to accommodate the chosen number of context words. The error vector from every output layer is added to adjust the weights using a backpropagation technique.

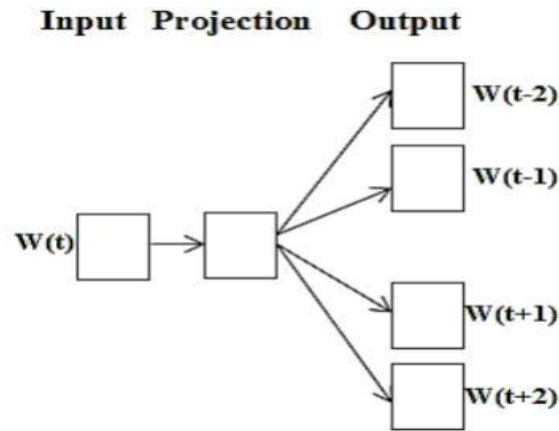


Figure 2-5: Skip-gram Model Architecture

GloVe (Global Vectors for Word Representation): GloVe uses collaborative filtering algorithms and matrix factorization, which are similar to Word2Vec's skip-gram model[30]. It is an unsupervised learning algorithm for vector-space word representation. Make a word-word co-occurrence matrix from the training document as a whole, and assign each word to a semantically appropriate location where the distance between related words is as small as possible. In practical terms, GloVe and Word2Vec perform similarly when it comes to embedding text.

2.7 Afaan oromo language

Afaan Oromo is one of the most widely used and spoken languages in Ethiopia and its neighboring countries, Kenya and Somalia. The Afaan Oromo language is acknowledged as the official language of the Oromia regional state. This language is essentially spoken by the Oromo people, who constitute the majority ethnic group in Ethiopia[8]. Since 1991, the Qubee alphabet, which is based on Latin, has been used as the official script for Afaan Oromo[31].

2.7.1 The writing system of Afaan Oromo

Afaan Oromo's writing system is nearly phonetic because it is written exactly as it is spoken, with one letter for each sound. The Afaan Oromo language uses the Latin alphabet Qubee, which was formally adopted in 1991. Seven of the 33 consonants in Qubee are combination consonant letters: ch, dh, ny, ph, sh, ts, and zh. We refer to the combination of consonant letters as qubee dacha. There are five short and five long vowels in the Afaan Oromo language. Like the English alphabet, the Afaan Oromo alphabet is made up of both capital and small letters. In Afaan Oromo, as in English, vowels are independent sounds that stand alone[32].

2.7.2 Word categories of afaan oromo

Words are the fundamental units of a language; they can be spoken or written, and they have meaning. In the

Afaan Oromo language, words are classified into two parts: the pattern, which is composed of prefixes and/or suffixes and gives the word its grammatical meaning, and the root, or base morpheme, which is usually composed of a single phoneme and gives the word its basic lexical meaning. The word "bare" (learned) is produced when the root "bar" and the pattern "-e" are combined, whereas the word "barte" (she learned) is produced when the root and the pattern "-te" are combined. The five grammatical categories of Afaan Oromo words are nouns, verbs, adverbs, adjectives, and adpositions[33].

Table 2-1 : Afaan Oromo word class

| | |
|----------------|----------------|
| “Gochima” | Verb |
| “Maqaa” | “Noun” |
| “Ibsa Maqaa” | “Adjective” |
| “Dabalgochima” | “Adverb” |
| “Raajeffannoo” | “Interjection” |
| “Walingaa” | “Conjunction” |
| “Durgala” | “Preposition” |
| “Maqdhaal” | “Pronoun” |

2.7.3 Afaan Oromo Sentence Structure

Afaan Oromo uses a subject-object-verb (SOV) structure, in contrast to English's SVO structure. For instance, in the Afaan Oromo sentence "Gaaddiseen barattuu dha," "Gadise" is the subject, "barattuu" is the object, and "dha" is the verb. "Gadise is a student" is the English translation of the statement. The way adjectives are formed in Afaan Oromo and English differs as well. Adjectives normally come after a noun or pronoun in the Afaan Oromo language; in English, on the other hand, adjectives usually come before the noun. For example, the adjective gaarii comes after the noun ilma in the phrase "ilma gaarii" (good boy)[8].

2.7.4 Afaan Oromo Punctuation

With the exception of apostrophes, punctuation is used exactly the same way in Afaan Oromo and English and has the same purpose. While the English language uses the apostrophe mark (,) to indicate possession, the Afaan Oromo language uses it to represent the glitch sound known as "hudhaa." In Afaan Oromo, the apostrophe mark (,) is crucial to the reading and writing systems. Punctuation is used in writing to improve readability and clarity of meaning. Afaan Oromo uses the same punctuation as English and other languages that are written using the Latin script.

Table 2-2: Afaan Oromo Language Punctuation Mark

| | |
|--|---|
| “Tuqaa (Full stop (.))” | “used in abbreviations and at the conclusion of sentences.” |
| “Mallattoo Gaaffii, (Question Mark (?))” | “used after a question or in an interrogative.” |
| “Raajeffannoo (Exclamation Mark (!))” | “utilized to conclude commands and exclamatory phrases.” |
| “Qoodduu(Comma(.))” | “It is employed to break up a list in a sentence or to break up a series of elements.” |
| “Tuq-lamee (Colon (:)) “ | “Used in addition to various traditional applications, etc., to introduce and divide lists, clauses, and quotations.” |

2.8 Related Works on Hate Speech Detection

To gain a clear understanding of the general approach, methodology, and findings of previous studies, This section offers a comprehensive overview of key works in the field of social media hate speech detection.

The study proposes detecting hate speech in Amharic text using deep learning techniques. Data from Facebook and Twitter was categorized into hate, offensive, and neutral classes. Word embedding and various models were trained, with the BILSTM model achieving the highest accuracy of 88.89% and f1-score of 89% for both original and augmented datasets. However, the study limited its use to machine learning algorithms for Amharic language[25].

The study proposes a deep learning approach for detecting hate speech in Amharic text using data from Facebook and Twitter. The model uses word embedding and features from Keras and Word2Vec embedding. Five models were trained using various deep learning techniques, with the BILSTM model with word2vec showing better performance with an accuracy of 88.89% and an f1-score of 89% for the original dataset. The authors limit their approach to RNNs, BiRNNs, and CNN models, and do not use cross-validation to minimize overfitting[25]. The study used Naive Bayes, Logistic Regression, and Support Vector Machines to create a model for identifying hate speech and offensive language on Twitter. The model made use of tweet-containing, publicly accessible datasets from Crowdfunder. With an accuracy of 95.6%, logistic regression performed better than other models in the ideal range of n-grams for L2 normalization of TF-IDF[34].

The study presented by F. Del Vigna [35] created models to use a deep learning technique to identify hate

speech in the Indonesian language. They contrasted the accuracy of both textual and auditory features. With an F1-score of 87.98%, the best model utilizing textual features outperformed lexical and acoustic features. The study highlights the importance of textual features in hate speech detection. S. G. Tesfaye and K. K. Tune [8] utilized gated recurrent units with word n-grams for feature extraction and long short-term memory to build recurrent neural network models for automated hate speech detection on Facebook. After 100 epochs, they were able to classify posts as free or hate speech with an accuracy of 97.9% thanks to their division of the dataset into train, validation, and test sets. The authors did, however, note that their model was only tested on a single dataset and recommended experimenting with additional deep learning models, such as bidirectional recurrent neural networks and attention mechanisms. A. Cimino [35] proposes an Italian online hate campaign using data from public Facebook pages. Two classifier algorithms, SVM and LSTM, are used to analyze the Italian language. The results show a high F1-score of 80% for binary classification and 79% for ternary classification. The authors suggest further deep-learning models for improved classification performance.

I. Aljarah et al [36] suggest a method that makes use of machine learning and natural language processing to identify hate speech on Twitter. A dataset of tweets about sports, racism, terrorism, journalism, sports orientation, and Islam is used in the study. According to the findings, Random Forest (RF) with profile-related features and TF-IDF achieved 91.3% accuracy. The study recommends further work for a more generalized dataset and effective detection models. Using a dataset from the Arabic region, the study presents a deep learning method for automatic cyber hate speech detection on Twitter. For feature extraction, word embedding techniques and a CNN/LSTM network hybrid were employed. The method classified tweets as normal or hateful with high recall, accuracy, precision, and F1 measures of 66.564%, 79.768%, 65.094%, and 71.688%. For improved outcomes, the study suggests using high-performance deep learning techniques and a more standardized dataset[37].

S. S. Aluru et al [38] explore deep learning-based multilingual hate speech detection, analyzing datasets from 16 publicly available sources in nine(9) languages. Four models were applied: MUSE + CNN-GRU, Translation + BERT, LASER + LR, and mBERT. The results indicated that different languages performed differently, with low-resource languages performing best for LASER+LR and high-resource languages for BERT, respectively. However BERT excelled in scenarios involving large amounts of data.

A. Ababa [2] developed a framework for detecting hate speech in the Afaan Oromo language that makes use of feature extraction and machine learning techniques. The framework achieved an accuracy of 96% using the support vector machine algorithm. Future improvements include expanding the dataset, adding hate speech categories like racism, sexism, politics, religious hate, and socio-economy, and detecting other forms of hate speech content on social media. Whereas, G. O. Ganfure [15] compares five techniques using deep

learning models to identify hate speech in the Afaan Oromo language, resulting in a model classifying hate, neutral, offensive, and hate & offensive content. It recommends investigating classifier ensembles and meta-learning tasks to address content misclassification issues. The I. Journal and O. F. Science [39] utilized machine learning techniques to detect hate speech text on Afaan Oromo social media, with the linear support vector classifier achieving the highest f1-score value of 64%.

Mossie and Wang [16] conducted a study on hate speech detection in the Amharic language using a dataset of 6120 Facebook posts. They classified the speech as "hate" and "not hate" using word2vec and TF-IDF feature extraction. They used machine learning algorithms naïve Bayes and random forest to detect these features, achieving high accuracy rates. The M. O. Ibromim and I. Budi [40] studied hate speech in Indonesian on social media using various machine learning algorithms, including word n-gram and RFDT, and achieved a 93.5% F-measure, with the best performance achieved with the combination of these models.

Davidson et al.[41] Davidson et al. conducted a study on automatic hate speech detection using 33,458 English tweets. They classified hate speech into hate, offensive, and neither categories using a hate speech lexicon. The study employed bigram, unigram, trigram features, TF-IDF, and part-of-speech sentiment lexicon for social media. The results showed high accuracy in detecting hate speech, with a precision of 0.91. D. Benikova et al [42] developed a deep-learning-based hate speech text classification system for Twitter using a dataset of four categories: racism, sexism, both (racism and sexism), and NHS. The model, based on word2vec embedding, achieved a 78.3% F-score, outperforming other models.

The F. Del Vigna, A. Cimino, and F. D. Orletta [35] investigate an Italian online hate campaign using comments on a public Italian Facebook page. Two classifier algorithms, SVM and LSTM, were designed and implemented for the Italian language. Word embedding lexicons, sentiment polarity, and morpho-syntactical features were employed by the researchers to classify the hate campaign. After two experiments, 70% of annotators agreed on the data's class, with SVM and LSTM achieving high F-scores for binary classification and ternary classification, respectively.

Another Florio et al.[43] study on Italian tweets, TWITA highlights the impact of training and test data time difference on models. They also developed Hurltlex, a hate word lexicon for identifying hate speech.[44]. T. M. Ababu and M. M. Woldeyohannis [7] developed a model to detect and classify hate speech using machine learning algorithms from classical, ensemble, and deep learning. The model achieved 0.82% accuracy for eight classification classes, while the deep learning algorithm achieved 0.84% accuracy. However, the study did not identify harassment or apply techniques to handle overfitting during training. Therefore, this research was focused on detecting hate speech and harassment identification based on

protected characteristics such as race, gender (sexism), religion, color, disability, and nationality. This research has used hyperparameter techniques to overcome the overfitting during the training process and to improve the performance of the model.

Table 2-3: Summary of binary class hate speech detection using machine learning approaches.

| Challenges | Solution | Feature extraction | ML and accuracy |
|--|---|---|---|
| Criminal activities online posing using the Amharic language [16]. | Detection of HS in the Amharic language. | Word2vec and TF-IDF. | NB and RF are 79.83% and 65.34% respectively. |
| Detecting an abusive language in the Indonesian language on social media[40]. | Detection of HS in the Indonesian language. | Bag of word(BOW), word n-gram, and character n-gram. | NB, SVM, Bayesian LR(BLR), and (RFDT) 93.5 %. |
| The problem of hate speech detection in online user comments[45]. | Detection of HS with comment embeddings. | Paragraph2vec & BOW with TF & TF-IDF. | Logistic regression obtains 0.80 AUC. |
| Lack of a system that detects cyber conflicts between people on Twitter [46]. | Detection of HS on Twitter. | Unigrams_with sentimental, semantic features, and pattern features. | J48graft, SVM, and RF: Accuracy 87.4 % for binary and 78.4 % for ternary. |
| The manual way of classifying hateful content on Twitter is costly and not scalable[47]. | Ensemble method for HS detection in Indonesian Twitter. | BOW, and TF-IDF weighting. | NB, KNN, maximum entropy, RF, SVM, and two ensemble methods: hard and soft vote, F1 measure 79.8%. (SVM, NB, and RF). |
| Lack of hateful detecting tools in tweets[48] | Detection of HS in Kenyan tweets. | Sentiment analysis & N-gram feature. | NB: P-0.58, R-0.62, A-0.67. |

Table 2-4: Summary of Multi-class hate speech detection using Machine learning Approaches.

| Challenges | Solution | Feature Extraction | ML and Accuracy |
|---|---|---|---|
| The separation of hate speech from other instances of offensive language[49]. | Automated HS and offensive_language detection. | Uni-gram, Bi-gram, and trigram feature with TF-IDF. | Logistic regression with l-1 regularization:90 %. |
| Lack of racism detection tools[50]. | Racism detection in Dutch social media. | Word2vec, Dictionary-based. | SVM; F1: 0.46. |
| Differentiating hate speech and offensive language[34]. | Detecting HS and offensive language on Twitter. | N-gram and TF-IDF. | LR, NB and SVM 95.6 %. |
| Lack of hate speech detection system in English tweets [51]. | Detecting HS in social media. | Word skip-gram, and surface n-gram. | SVM:0.78. |

Table 2-5: Summary of Binary and Multi-class Hate Speech Detection using Deep Learning Approaches.

| Challenges | Solution | Feature Extraction | DL and Accuracy |
|---|--|---|--|
| Lack of deep learning-based Twitter hate-speech text classification system[52]. | Convolutional_neural_network (CNN) to classify HS. | Word2vec, Random vector, character n-grams, and word2vec+character n-grams. | CNN with word2vec:0.78 F-score multi-classification. |
| Lack of automatically detect hate speech systems on social media[53]. | HS detection using NLP. | Word2vec with 300 dimensions. | CNN accuracy of 91 %, and a loss of 36%. |
| The complexity of the natural language constructs makes this task very challenging[54]. | HS detection using deep learning. | Random embeddings & glove embeddings. | CNN-LTSM & Fast Text's best accuracy is 93 % F1-score CNN + Random & Glove Embeddings. |

Table 2-6: Afaan Oromoo Hate Speech Detection Related Work.

| Objective | Work or classification done | Accuracy | Drawback |
|--|---|----------------------------------|--|
| Hate speech detection framework from social media for afaan Oromo language[2]. | Hate speech, and neutral speech. | SVM: 96%. | It detects whether the text is hate speech or neutral and it does not apply any hyperparameter tuning techniques. |
| Comparative hate speech detection for afaan Oromo language[15]. | Hate, neutral, offensive, and hate&offensive. | CNN and BiLSTM F1-score of 87%. | It detects whether the text is Hate, neutral, offensive, and hate&offensive and it does not apply any hyperparameter tuning techniques. |
| Detection of hate speech text in afaan oromo social media using machine learning approaches[39]. | Hate, and normal. | LSVM F1-Score: 64 %. | It detects whether the text is Hate and normal and it does not apply any hyperparameter tuning techniques. |
| Detection of hate speech and classification of the hate speech for the afaan Oromo language[7]. | Race, religion, gender, and offensive class. | SVM: 0.82 %, and BiLSTM: 0.84 %. | The researcher does not consider other areas such as disability, color, and nationality and it does not apply any hyperparameter tuning and overfitting handling techniques. |

3 CHAPTER THREE

RESEARCH METHOD

This chapter covers a research methodology that aims to identify and detect hate speech and harassment on social media platforms using protected characteristics specific to the Afaan Oromo language. We go over the methods of gathering data and the preprocessing steps taken to create datasets on harassment and hate speech. Additionally, the actual process for creating and refining models as well as the research's model evaluation technique are described. The following section provides a brief description of the processes.

3.1 Data Collection

Textual data with protected characteristics serves as the basis for Afaan Oromo hate speech text detection and harassment identification on social media. On the Facebook and Twitter networks, the posts and comments were gathered from well-known public pages. Facebook's privacy policies forbid access to the public content of private pages, so private pages weren't taken into consideration. ID, link, source, posts, comment, context, label, and category are among the features included in the gathered dataset. The posts and comments together formed the basis of the context column. There were two methods used to gather the data. Using Facebook pagers, posts and comments from Twitter or Facebook were first gathered. The following procedures are followed after installing the face pager software: obtain the public page and generate the ID, make a new database; add notes, fetch data, and export the data in Excel file format. A procedure was then established to gather information from the community. Afaan Oromo language speakers were specifically chosen using a Google Form. We took this course of action because the Facebook posts and comments did not contain any hate speech based on nationality, color, or disability. As a result, as can be seen in Appendix 2,2.5. We developed a new procedure and methodology to record hate speech based on nationality, color, and disability using Google Forms.

A methodological framework, Google Forms, was used and adopted[55]. Structure: Google Forms suggested: (a) Use unbiased, clear language: To prevent respondents from being misled or confused, ensure that survey questions are succinct and clear. (b) Randomize the order of the survey's questions to minimize response bias and order effects. Following that, the following steps were conducted:

Step 1: Hire or find a person who has skills and professions in the Afaan Oromo language to collect, organize, annotate, and analyze the dataset, and check or review whether the annotated dataset is accurate or not.

Step 2: Give a description of the dataset and its classes to the expert.

Step 3: The expert has written the text from sources such as books and their corresponding classes.

Step 4: Finally, combine the dataset collected from the expert with the context columns.

Data preparation

The data was cleaned and filtered, and the posts and comments were annotated for the purpose of training models. The ensuing assignments were completed:

- Removing non-textual posts and comments. Removing irrelevant characters.
- Removing null, blank values, and extra whitespace. Combining the data into a single file.
- Removing duplication to ensure the uniqueness of each text in a dataset.

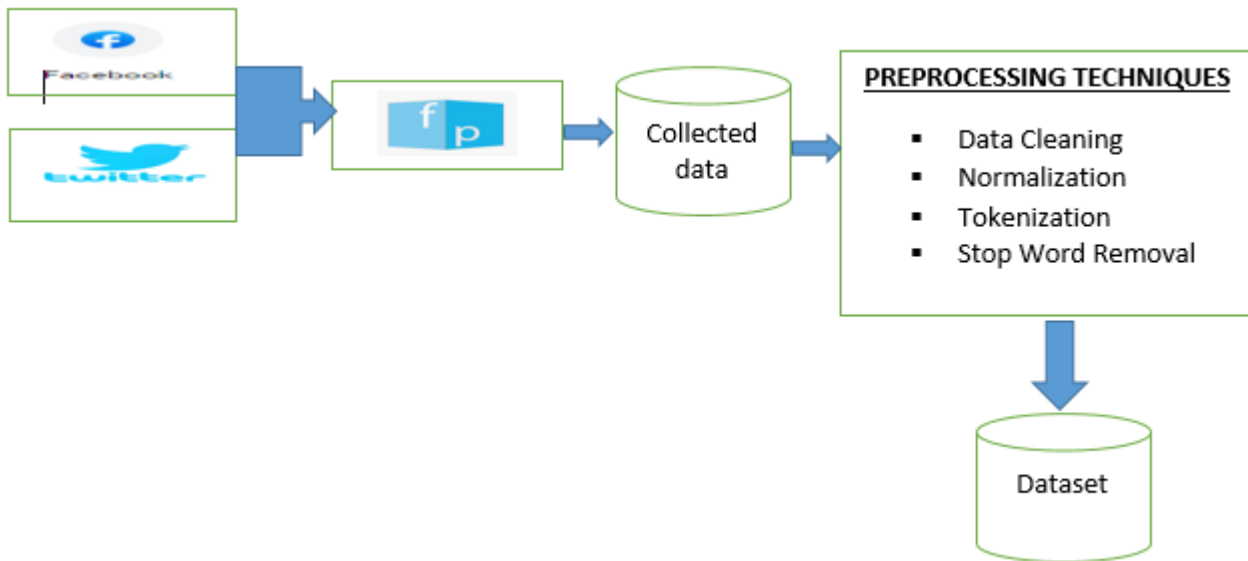


Figure 3-1: Dataset preparation procedure

Table 3-1: Selected Pages

| No | Page name | Page categories |
|-----|-----------------------------------|---|
| 1. | Oromia Media Network | Media /News Company |
| 2. | Ortodoksii Page | Religious Organization |
| 3. | Oromia Broadcasting Service – OBS | Broadcasting and Media Production Company |
| 4. | Tvislaama | Broadcasting and Media Production Company |
| 5. | OBN Afaan Oromoo | Media /News Company |
| 6. | FBC Afaan Oromoo | Media /News Company |
| 7. | VOA Afaan Oromoo | Broadcasting and Media Production Company |
| 8. | Ahmedin Jebel Official | Personal Blog |
| 9. | BBC News Afaan oromoo | Media /News Company |
| 10. | Mana Lubummaa | Religious Organization |

3.1.1 Dataset Annotation Guide line

Thus, when annotating a dataset, we created a general guideline in our study to prevent ambiguity. From the standpoints of law, free speech, and literature review, we define and categorize hate speech as indicated below. Six thematic categories—race, religion, sexism, color, disability, and nationality—are used to compile the dataset. There is hate speech in every place.

Speeches about race, religion, sexism, color, disability, and nationality:

The sentence (speech) is classified as hate speech in the categories of race, religion, sexism, color, disability, and nationality if at least one of the five criteria (1–5) is met. On the other hand, if the sentence discusses any of the categories but falls short of any of the following criteria, it is classified as not hate speech in the categories of race, religion, sexism, color, disability, and nationality[7].

1. If the post or comment, through discrimination on the basis of race, religion, sexism, color, disability, and nationality, incites hatred or violence.
2. If the post or comments encourage hatred and discrimination against people based on their race, religion, sexism, color, disability, and nationality.

3. If the post or comments encourage violence or injury action must be taken against anyone discriminating based on race, religion, sexism, colour, disability, and nationality.
4. If a comment or post imitates some of the pointless objects, machinery, or animals, and some discourage psychological morals by discriminating based on race, religion, sexism, colour, disability, and nationality.
5. If the comment or post offends someone by discriminating against them based on their race, religion, sexism, colour, disability, and nationality.

Lastly, the class dataset is flagged as containing hate speech under the categories of (nationality, disability, color, race, religion, and sexism). In the event that the aforementioned requirements are met, the following categories—race, religion, sexism, color, disability, and nationality was classified as non-hate speech[7].

3.1.2 Dataset description

The final dataset contains a total of seven features or attributes, such as ID, link, source, posts, comment, context, and label/category, where the context was based on the combination of posts and comments. However, only two variables, such as context and category, were used for model training, as shown in Table 3-2.

Table 3-2: Dataset Description

| No | Context | Category |
|----|---|----------------------|
| 1. | Uummaanni Amaaraa Nafxanyaadha The Amhara people are Nafxanya. | Hate in Race |
| 2. | Oromoon Gadaan Bula. The Oromo are ruled by the Gada. | Not hate in race |
| 3. | Obbo Shimallis Hordofoota Amantaa islaamatiin Baga Geessan jedhan. Mr. Shimallis said Congratulations to the followers of Islam. | Not hate in religion |
| 4. | Amantin Kiristana amanta sobaatti. The Christian religion is a false religion. | Hate in Religion |

| | | |
|-----|---|-------------------------|
| 5. | Dhiira furdaan jibba. I hate fat men. . | Hate in sexism |
| 6. | Shammarrri jimmaa bareeddudha. The girls of Jimma are beautiful. | Not hate in sexism |
| 7. | Uummanni adiin wallaaladha. The white people are ignorant. | Hate in color |
| 8. | Uummanni adiin diina uummata gurraacha miti. The white people are not the enemy of the black people. | Not hate in color |
| 9. | Gurbaa miilla yookiin miila cabaa san hin jaaladhu. I don't like that guy with a broken leg or foot. | Hate in Disability |
| 10. | Namoonni Qaamaan midhaman dandeetti beekumsa qabu. People with disabilities have knowledge and skills. | Not hate in disability |
| 11. | Lammileen Itoophiyaa fiigichaan tokkoffadha. The Ethiopians are famous runners | Not hate in nationality |
| 12. | Lammileen itoophiyaan jaldeessan tokko. The citizens of Ethiopia are one of the wolves. | Hate in Nationality |

3.2 Research Design

This study uses an experimental research design to achieve the goal of the thesis. A study conducted by a scientific research methodology is called experimental research. The purpose of experimental research is to establish a relationship between two variables, referred to as the dependent and independent variables[56]. Thus, in this study, hate speech detection and harassment identification on social media for the Afaan Oromo language were carried out.

To do this, the process flow illustrated in Figure 3-2 was utilized, which comprises the subsequent two primary steps: To better understand the problem, the first step is to identify its domain by reviewing some different previously performed activities. The thesis's general and particular goals are then established. The first stage of the thesis design dealt with data preparation, and the second stage dealt with model creation.

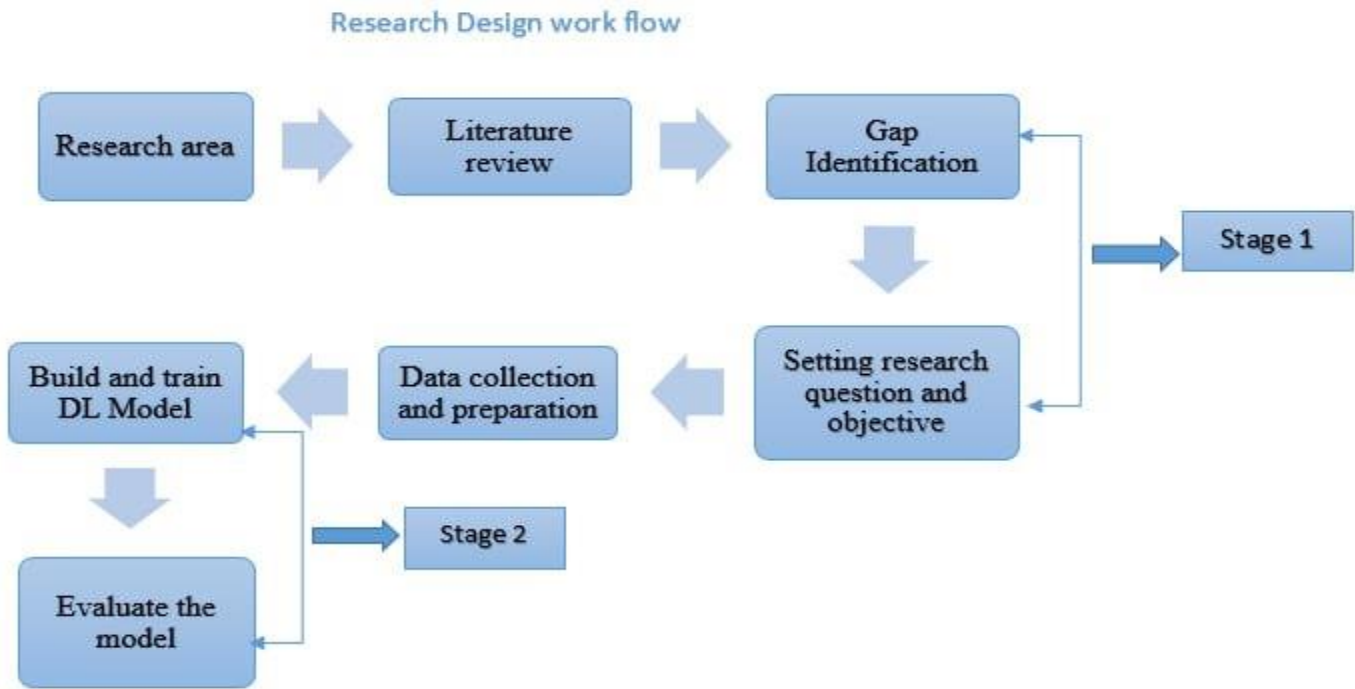


Figure 3-2: Research Design workflow

3.3 Data Preprocessing

Data preprocessing is crucial for detecting harassment and hate speech on social media in the Afaan Oromo language. It improves the quality of the dataset and prevents noise and unorganized forms from being used in the model. Techniques used include removing unrelated items, punctuation, stop words, normalization, and tokenization before developing the deep learning model.

Data cleaning- Data cleaning is the process of removing noise from the dataset—for example, extraneous symbols, white spaces, punctuation, and other elements—to improve the dataset's interpretability for the model.

Stop words- Not every word in the documents is equally significant when it comes to developing a system that uses the protected features of the Afaan Oromo language model to identify and detect hate speech and harassment on social media. Certain words have no bearing on the user's query and are not utilized to determine the relevant tag. To shorten the model's learning time, those words must be removed from the dataset. The stop phrases in Afaan Oromo were prepared for this work by gathering them from various earlier studies [57]. These are the most common terms in all languages, along with articles, particles, conjunctions, pronouns, and prepositions. Afaan Oromo stop words that are frequently used are fi/and, kana/this, Akka/like, kanaaf/so, siif/for, irra/on, sun/that, isa/he, kee/yours, etc.

Normalization- Words written differently by users may have different meanings for the machine. Text normalization, which combines related terms written in various formats into a single, unique word, can solve this issue. It aids the models in identifying the same word when written differently. Lowercasing is the most popular and straightforward normalization technique. The terms "Fayyaa," "FayyAA," and "FaYYaa" are represented differently in the model. Thus, each pattern in the dataset is transformed into a small letter by applying lowercase. Furthermore, terms that are written in different formats but have comparable meanings are normalized using a regular expression. In the Afaan Oromo language, for example, the terms haga/hanga, erga/ega, baayee/baayy'ee, osoo/otoo, mini/miti, etc. have all been normalized into a single word.

Tokenization- Tokenization is the most basic action taken on text data when working with a textual dataset. The text needs to be tokenized into tokens using spaces during the text preprocessing step. As an example, the definition of "Addaa beekamoon naanoo oromiyaa kessa beekaman maal fa'a?" is "What are the most well-known specialties in the state of Oromia?" The tokenization of this sentence is 'Addaa', 'beekamoon', 'naanoo', 'oromiyaa', 'kessa', 'beekaman', 'maal', 'fa'a'?. The deep learning models do not work directly with text data, so the fully cleaned data needs to be transformed into numerical data using techniques like word embedding or text feature extraction.

3.4 Feature Extraction Methods

Taking the list of words and turning them into vectors is called text feature extraction. It is also employed in the development of the high-performance model for feature dimension reduction. The following section discusses the various feature extraction techniques.

3.4.1 Bag of words

This approach only takes into account the document's lexicon of recognized words and their presence within it. It doesn't take the word order or structure into account. With a larger vocabulary, documents are represented more vectorially. Due to its limitations, this feature extraction technique was not used in this study. This method's drawbacks include its sparse representations, disregard for word order, and disregard for context[57].

3.4.2 TF-IDF (Term Frequency-Inverse Document Frequency)

The well-known method known as TF-IDF still performs comparably to other cutting-edge methods[58]. By taking into account the frequency of the words and how often they appear in all documents, TF-IDF resolves the issue with BoW methods. Additionally, since this method disregards word order and semantic

relationships, it is not applied in this study. Compound words are not recognized by it as a single word. These can have an impact on the learning model's accuracy, and we lose out on additional information such as word order, semantics, and context surrounding nearby words in every text.

3.4.3 Word Embedding

The dispersal representation of a word with a vector in many natural language processing tasks, such as chatbots and automatic machine translation, text classification is called word embedding. There are word embedding techniques like word2vec and GloVe[59]. Words with semantic relationships are inserted into the surrounding vector space by the Word2Vec algorithm. The word2vec model uses two different types of methods: the CBoW and skip-gram methods. The skip-gram method, which is effective for small datasets, predicts the context neighbor words as output given the target word. In contrast to the skip-gram method, the CBoW approach predicts the target word using the provided neighbor window size words.

3.5 Model Selection Techniques

Several deep learning algorithms such as CNN, LSTM, BiLSTM, GRU, and BERT are used in advanced (modern) methods of detecting and identifying hate speech and harassment on social media based on protected characteristics for the Afaan Oromo language models. Some problems are impossible to solve with perfect models, so before conducting experiments, it is impossible to determine which model will work best. Based on protected characteristics, this study established some criteria for selecting suitable algorithms for the development of the Afaan Oromo language model, including the detection and identification of hate speech and harassment on social media.

The first prerequisite is to determine the type of problem that needs to be solved—is it one of regression, prediction, or classification? The suggested detection and identification of hate speech and harassment on social media is predicated on the protected characteristics of the Afaan Oromo language model, posing a multi-class classification challenge. The dataset was produced as a multi-text Excel file. Each tag has several patterns that are entered by users; the class that each pattern belongs to is the output. Consequently, the study addresses multi-class classification issues. Deep learning models outperform machine learning models in multi-class classification tasks[60].

Machine learning types form the basis of the second criterion. Based on the methods used to train the algorithms, machine learning can be divided into three categories: supervised, unsupervised, and reinforcement learning. For this study, the supervised deep learning model was chosen since the target value is known and the dataset has been labeled. To develop detection and identification of harassment and hate

speech on social media based on protected characteristics for the Afaan Oromo language using a prepared dataset, it is possible to search for a suitable model that makes the most accurate classification.

The third criterion is the model that has been used the most frequently and has performed the best in earlier studies for handling NLP tasks. Many deep learning algorithms performed well on natural language processing tasks, especially text classification. These algorithms include CNN, DNN, and models based on RNNs[61][62].

The final requirement is to use a prepared dataset to conduct experiments between the suggested deep learning models to determine which model performs better for this investigation. As far as we are aware, one model cannot be declared superior without first testing several others. We can work with the best model to achieve additional performance gains once it is identified.

Multiple models, such as CNN, LSTM, BiLSTM, GRU, and BERT, were trained and evaluated using cross-validation, loss, and accuracy evaluation metrics to determine which model was better for the proposed hate speech detection and identification of harassment problems based on this evaluation. The BERT model was recommended to detect hate speech and identify harassment on social media for the Afaan Oromo language.

3.6 Deep Learning Algorithms for the Harassment Identification

Deep learning models are utilized in a variety of NLP applications such as sentiment analysis, question answering, etc. [63]. Advanced detection and identification of harassment and hate speech technology uses the concepts of understanding natural language and applying deep learning algorithms. DL-based approaches to harassment and hate speech detection are distinct from pattern recognition techniques, which rely on data-driven feature learning. We then select the RNNs-based deep learning models and some other models RNN-based models with the CNN model for this work, which is explained in the next section.

3.6.1 Recurrent Neural Network (RNN)

RNN, as opposed to ANN, is formed of fully interconnected neurons. The recurrent neural network is an extension of the general feed-forward network that considers both the current input and past output while generating the output. It uses the concept of processing sequential data for tasks that involve sequential inputs. The models interpret the sentence as a sequence of words, which makes the algorithms better for capturing word dependencies. Recurrent neural network models have been successful in QA[64]. The algorithm has two input values, the current input values, and values from the recent previous layer, as shown in Fig.3-2.

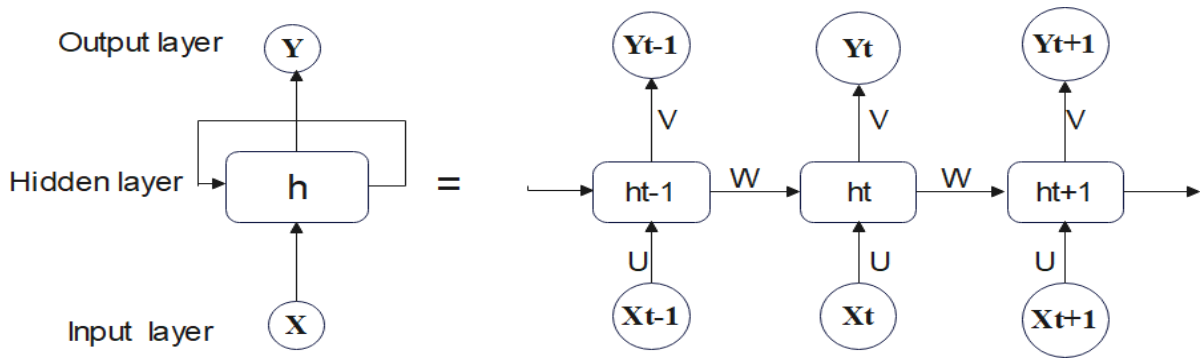


Figure 3-3: The Architecture of the RNN model

To compute the hidden and output layers at timestamp t , it uses the equation Eq.3.1:

$$h_t = \delta(U * X_t + W * h_{t-1} + b_h) \quad \text{Eq. 3-1}$$

$$y_t = \text{softmax}(V * h_t + b_y) \quad \text{Eq. 3-2}$$

Where $h(t)$ and y_t are hidden layers and the output value time stamp t respectively, δ is the activation function, V is the weight vector of the output layer, The word vector's input value is X , the hidden layers weight vector is U , and the bias is b . Despite the RNN being designed to work with sequential data, it has shown shortcomings in recalling input for long sequences. Practically, an RNN model presents limited achievement in long dependencies. Also, it has gradient exploding and vanishing problems. As a result, among various variants of RNNs, the LSTM is the best variant invented to handle the shortcomings of standard RNNs[65].

3.6.2 Long Short-Term Memory (LSTM)

Recurrent neural networks called Long Short-Term Memory (LSTM) Networks were created to solve the vanishing gradient issue with conventional RNNs. Their ability to maintain information over longer sequences and capture long-term dependencies makes them appropriate for tasks such as text generation, text classification, and language modeling. LSTMs are useful for capturing contextual information in text because they can retain significant information over time.

$$F_t = \delta(X_t * U_f + h_{t-1} * W_f + b_f)$$

$$I_t = \delta(X_t * U_i + h_{t-1} * W_i + b_i)$$

$$C_t = \tanh(X_t * U_c + h_{t-1} * W_c)$$

$$N_t = F_t * C_{t-1} + I_t * C_t$$

Where X_t is the current timestamps input, U_f is the input values weight, h_{t-1} is the previous timestamp of the hidden state, b is the bias, W is the hidden states weight matrix, C_t is a vector of new candidate values, and N_t is new information.

Finally, the output o_t is calculated as $O_t = (X_t * U_o + h_t * w_o + b_o)$ and then gets the final hidden state of timestamp t as $H_t = O_t * \tanh(N_t)$. This shows how the three gates of the LSTM algorithm work together to produce h hidden state at time t . Fig.3-4, depicts the general architecture of the LSTM model.

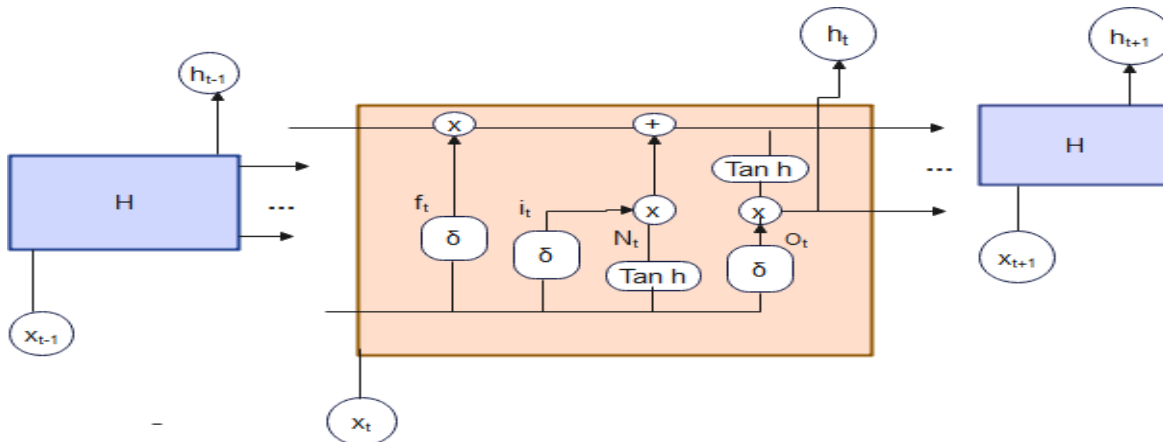


Figure 3-4: The LSTM Model's architecture.

3.6.3 Bidirectional Long Short-Term Memory (BiLSTM)

An extension of LSTMs, bidirectional LSTMs process input sequences simultaneously in both forward and backward directions. They improve performance in tasks like machine translation, sentiment analysis, and text classification by capturing a more thorough understanding of the text. Both local and global dependencies are successfully captured in the text by them.

3.6.4 Gated recurrent unit (GRU)

Recurrent neural networks called Gated Recurrent Units (GRUs) solve the vanishing gradient issue and

outperform LSTMs in computation. GRUs are computationally less expensive with fewer parameters, but they still perform well on sequential data modeling tasks, which makes them appropriate for machine translation, text classification, and language modeling [66].

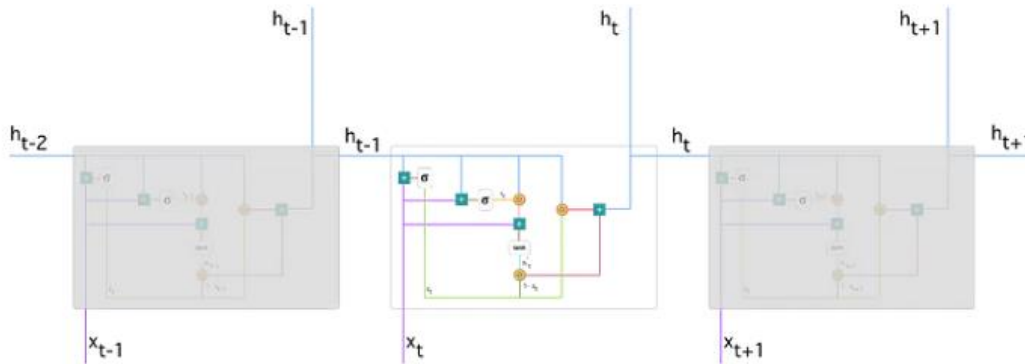


Figure 3-5: The GRU Model's Architecture

3.6.5 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are robust models for data processing that can recognize hierarchies and local patterns in the data. They are excellent at text classification because they can identify and extract important features from sequential data, which makes them perfect for tasks like text categorization, sentiment analysis, and document classification. They can use truncation or padding to handle inputs of varying lengths[67]. It's called ConvNets, and it requires numerical data to work with text; for this, the word embedding method is utilized. As shown in Fig.3-6, the word vector matrix, the one-dimension convolutional layer, MaxPooling, and the fully connected layer are required to work with text data when using the CNN model for text processing. MaxPooling carries out the network to hold only the maximum value in a feature vector which is the most useful and local feature.

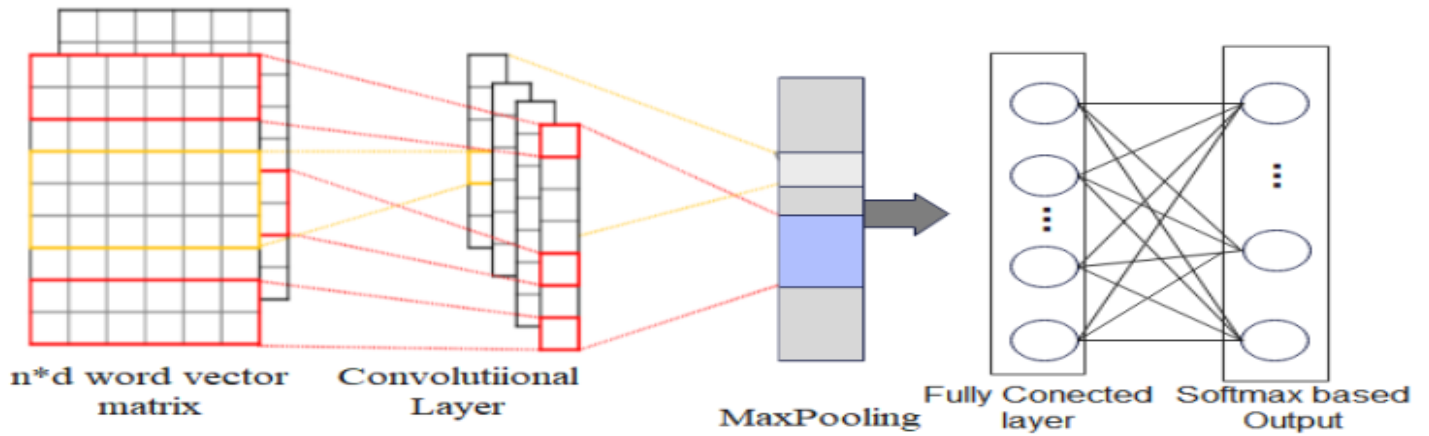


Figure 3-6: The Architecture of CNN Model

3.7 BERT Pre-trained Model

Google AI's BERT Natural Language Processing Model demonstrated exceptional accuracy on 11 NLP and NLU tasks, such as GLUE and the Stanford Question Answering Dataset. BERT can be adjusted to particular NLP tasks because it was pre-trained using text from Wikipedia and Book Corpus. It gets around the problem of not having enough training data by applying a large unlabeled text corpus and tailoring it to particular tasks. Transformers enable BERT to be bi-directional, meaning it can read text in both directions at the same time[68].

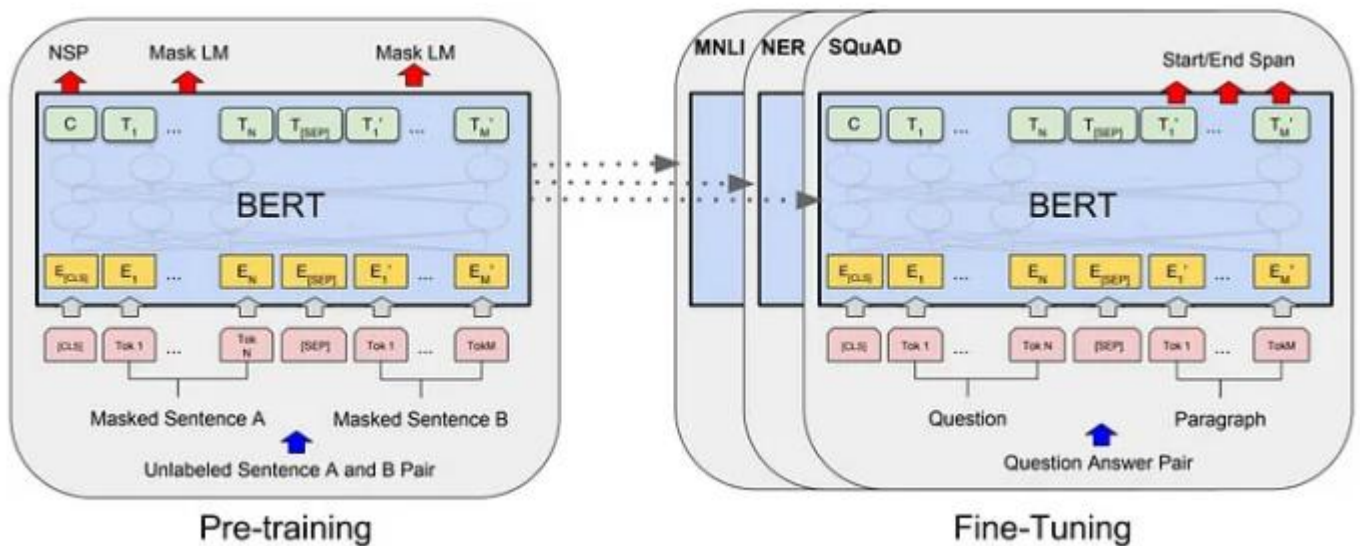


Figure 3-7: Bert Pre-trained model and fine-tuned architecture

3.8 Advantages of deep learning over classical machine learning

SVM, decision trees, and random forests are a few examples of machine learning models that provide interpretability, computational efficiency, and efficacy even with small datasets. However deep learning models—such as CNNs, RNNs, and BERT—are more adept at picking up on subtle hate speech patterns. These models are perfect for tasks involving the detection of hate speech because they can automatically extract features from raw text data. They are adept at processing vast volumes of data, drawing precise generalizations, and spotting intricate patterns. Transfer learning is made possible by pre-trained models, such as BERT, which help them acquire rich language representations[69].

3.9 Model Evaluation Techniques

Model performance evaluation is a quantified representation of the learning process. This work evaluated the proposed model performance using several model evaluation techniques such as cross-validation and evaluation metrics like accuracy and loss. A model needs to perform similarly in the unseen dataset; simply choosing the model with the highest accuracy on the training dataset is insufficient to declare it to be good. As a result, the performance of the suggested model is assessed using the evaluation methodologies listed below.

3.9.1 Cross-Validation

Cross-validation is a method of measuring the model performance by separating the dataset into k equal partitions and used for training and testing iteratively. It is utilized to prevent model overfitting, in a case when the amount of data is limited, and allows one to evaluate the model's capacity for generalization.

➤ K-fold Cross-Validation

K-fold cross-validation is a machine-learning technique used for model evaluation and selection. A dataset is divided into k folds, a subset is used to train the model, and the remaining fold is used to evaluate it. This procedure is carried out k times., resulting in a more reliable estimate of the model's performance. K-fold cross-validation is also used for hyperparameter tuning, allowing data scientists to select the best combination of hyperparameters for optimal results. It also aids in model selection, ensuring every data point is used for training and evaluation, especially in limited datasets. It also reduces variance in performance estimates[70][71]. Because it shuffles the data into folds at random.

➤ Stratified K-fold Cross-Validation

K-fold stratified cross-validation is an extension of k-fold cross-validation, dividing data into folds while maintaining class distribution. It is useful in dealing with imbalanced datasets with uneven class distributions. The goal is to guarantee that a comparable percentage of samples from each class are present in each fold or target variable category, resulting in more reliable performance estimation. K-fold stratified cross-validation is commonly used in imbalanced datasets, classification tasks, and model evaluation and comparison. It provides a more accurate estimate of model performance, mitigates potential bias, and ensures a more reliable evaluation of the model's ability to generalize to different class categories[72][73]. Thus, the stratified K-fold cross-validation ensures that each fold represents the complete data.

3.9.2 Evaluation Metrics

Accuracy: is a measure that frequently describes the model's performance in all classes. In every industry, evaluating the effectiveness of machine learning models is a crucial metric. It indicates the proportion of our test data that is appropriately classified. This metric is sufficient when the significance of each class is equal. The accuracy of the model can be determined by dividing the total number of classifications by the number of correctly classified instances. Eq 3-3 provides the mathematical formula that is used to compute it.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{Eq. 3-3}$$

Where TP is a true positive that is accurately estimated as factual, FP is a false positive that is a negative value predicted as positive, TN is a true negative that is correctly predicted as negative, and FN is a false negative that is a positive value predicted as negative.

Loss: - is an assessment metric that calculates the model's training and validation losses. While the validation loss assesses the training loss and the model's performance on the validation dataset gauges the model's fit on the training dataset. The sum of the errors for each instance in the training set and validation set is used to calculate the training and validation losses, respectively. In this work, this metric was used to inform us about whether further tuning of the model is necessary or not. As a result, we use this metric to visualize the model's fit on the training and validation datasets, respectively, on a graph.

A transformer layer, a self-attention layer, and a hidden state layer are some of the parts of the Bert model. An integral component of the transformer architecture is the transformer layer, which is made up of a feed-forward neural network layer and a self-attention layer. The outputs are subjected to non-linear transformations by the feed-forward layer, while the self-attention layer calculates attention weights among words. Transformers mimic complex relationships and hierarchical structures in input sequences by employing multiple-layer

stacking. The hidden state layer is the intermediate layer that contains contextualized representations for tasks that come after, like sequence labeling or classification. The hidden state layers in BERT are the most helpful of those elements because they provide a deeply contextualized representation of the input tokens. However, fine-tuning procedures like removing or altering the particular hidden state layer of the model and assessing the impact on performance, using different attention mechanisms, or changing the input representations were carried out to ascertain which specific architecture, component, feature, layer, or attribute of a BERT model is significantly contributing to achieving high or low performance. The components that significantly contribute to the model's performance are identified by contrasting the outcomes of the modified models with the original model. By doing this, the result shown in Section 4.2 was obtained.

3.10 Natural language processing

Natural language processing scales other language-related tasks and facilitates human-to-computer communication in the vernacular. Computers can now read text, hear speech, analyze it, gauge sentiment, and pick out the key details thanks to natural language processing (NLP). One popular Python library for working with natural language is called NLTK. Features like word count, tagging, tokenization, stemming, and lemmatization are all included in this library. Natural language processing and closely related fields like machine learning, artificial intelligence, information retrieval, and linguistics are supported by NLTK. This study worked with and preprocessed Afaan Oromo hate speech and harassment identification datasets using the NLTK library.

3.11 Model Overfit Handling Techniques

Model overfitting, which occurs when a built model is only well-optimized on the training set and fails to generalize well for unknown data, is the most prevalent issue with neural networks. In the other scenarios, either an inadequate amount of data or a complex architecture leads to model overfitting. Deep learning models that overfit can be addressed in a variety of ways. Deep learning models frequently use dropout, data augmentation, early stopping, and cross-validation—the two most popular regularization techniques—to address overfitting. Because the suggested models are prone to overfitting, the cross-validation method and the L2 regularization technique have been used in this study. The L2 regularization approach is used because it shortens training times, decreases generalization error loss, and increases squared magnitude as a penalty for forcing the weights to a small value—but not zero—the L2 regularization technique is used[74].

3.12 Hyperparameter techniques

The three layers of neural network-based models must have the appropriate hyperparameters adjusted when

working with them. The basis for differences in accuracy between models is variations in hyper-parameters within the same neural network. To increase the neural network model's performance, more care must be taken in determining the ideal hyperparameter. It is a laborious and experimental task to tune the neural network with the appropriate hyperparameters, such as the number of iterations (Epochs), learning rate, optimization algorithms, neurons in each layer, batch size, etc. To determine the ideal hyperparameter for the neural network, two hyperparameter tuning algorithms are available: randomized search and grid search[75].

3.13 Ethics of the Research

Speech that denigrates, harasses, threatens, or incites hatred towards an individual or group based on a trait like race, ethnicity, religion, gender, or sexual orientation is considered hate speech and harassment. When interacting with people on social media, always use appropriate language. We would be wise to refrain from disseminating information about specific racial or religious groups. Uploading graphic images, such as those from traffic accidents or other violent incidents, may raise ethical concerns. Instead, share only pertinent information and keep your distance from other people.

3.14 The Proposed Architecture

The proposed detection and identification of harassment and hate speech on social media based on protected characteristics for the Afaan Oromo language is intended to categorise Afaan Oromo's social media posts and remarks in twelve (12) classes as hate in race, not hate in race, hate in religion, not hate in religion, hate in sexism, not hate in sexism, hate in color, not hate in color, hate in disability, not hate in disability, hate in nationality, and not hate in nationality. The suggested architecture includes preprocessing, feature representation, model building, and model evaluation components, as seen in Figure 3–8. Preprocessing techniques are applied to the hate speech and harassment dataset, which is written in the Afaan Oromo language. Tokenization, normalization, and cleaning are examples of preprocessing techniques used before representing the dataset as a feature vector.

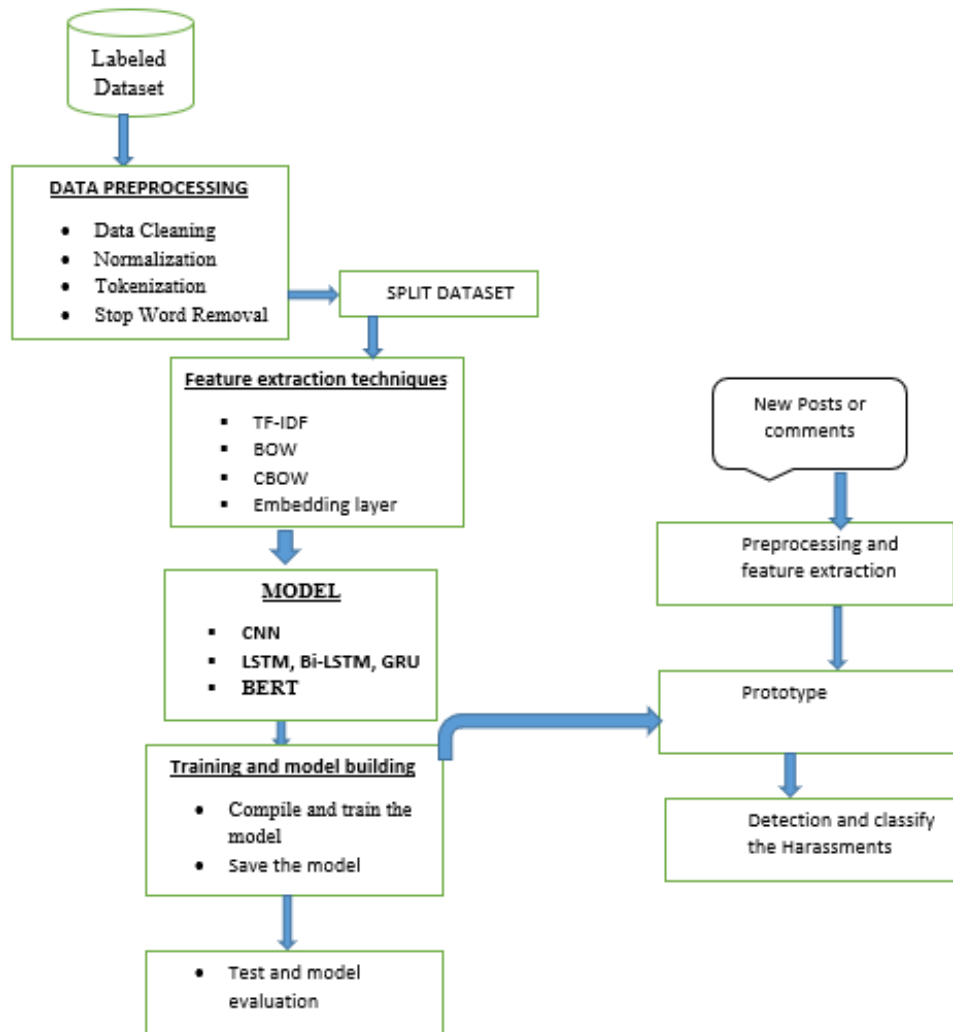


Figure 3-8: The General architecture of Hate Speech Detection and harassment identification Model

3.15 Proposed Deep Learning Model

To propose the deep learning model for this investigation, the study set different criteria, which are detailed in section 3.7. Based on those criteria, instead of using the standard RNN, one variant of recurrent neural networks called CNN, LSTM, BiLSTM, and GRU is proposed. The recurrent neural networks and even LSTM have no access to future information to learn sequence correlations. And then, the model will lose some extra information when processing sequential data. It has also proven that the RNN model has gradient exploding and vanishing problems. To overcome these drawbacks, the study proposed four deep learning models which are described in section 3.7, with their explicit capability and limitations in NLP tasks. The study experimented with the model using a newly collected dataset and evaluated them under stratified cross-validation to identify the most suitable model for designing hate speech detection and harassment identification, as shown in Fig.3-9.

3.16 BERT Pre-Trained Language Model

A language model based on contextual representations and trained on massive amounts of data is called Bidirectional Encoder Representations from Transformers (BERT). BERT is made up of layers for the model (such as Named Entity Recognition, Question Answering, and Classification) and feature extraction layers, which include word embedding[76].

When compared to other language models, BERT, the most recent model, yields state-of-the-art results for a variety of NLP tasks. In the word embedding training process, BERT is different from other word embedding models in that it generates a bidirectional representation of words that can be learned in both left and right directions. Word representation produced by word embedding techniques such as Word2Vec and GloVe is static and does not adapt to changes in context because they look just in one direction, which can be either right to left or left to right. BERT is distinct from earlier language models (such as ELMo, which stands for Embeddings from Language Models) [76] in that it is capable of left- and right-handed manipulation of the context in all layers. It combines both the left and right contexts through cooperative conditioning, as opposed to superficial combining techniques like concatenating.

The 2,500 million-word English Wikipedia and the 800 million-word Books Corpus are used to train BERT (devlin2018bert). The BERT, a previously trained language model that had been adjusted for our goal, was used in the second experiment. Fine-tuning is the process of training an application-specific subset of a pre-trained model that was initially trained on a sizable generic text. The input text is encoded by BERT using its embedding vectors. For the sequence classification model, we employed BERT, which consists of a classification neural network layer. The input sentence is converted to tokens in the first stages of the BERT model. The segment, position, and token embeddings are combined to form the token embedding vector. To identify the beginning position of the classification task—that is, the starting position of the fully connected layer to the last encoder layer and ultimately to the softmax layer—BERT uses the shorthand [CLS] for classification. This unique token is inserted at the beginning of the sentence tokens.

Different versions of BERT were released, each with unique properties depending on the language (Chinese, English, Multilingual, etc.) and alphabet (BERT-Base and BERT-Large, for example). With 12 Transformer layers and 12 self-attention heads per layer, the BERT-Base model has 768 hidden states in total. There are a total of 1024 hidden layers, 16 self-attention heads, and 24 transformer layers in the BERT-Large model. The model parameters for training testing are batch size = 16,8, number of train epochs = 3.0, and learning rate = $2e-5$, which are the values suggested by the literature for sequence classification tasks[68].

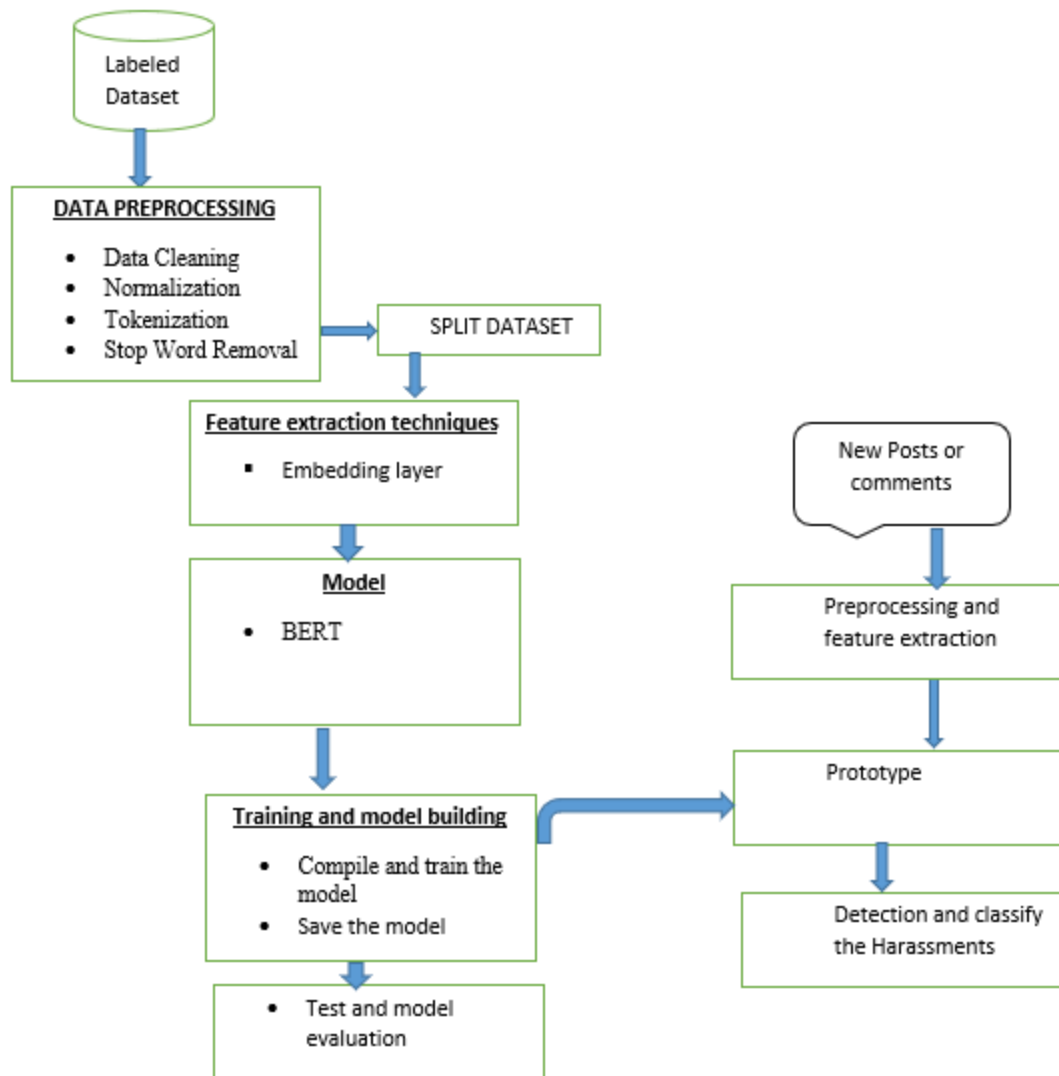


Figure 3-9: The proposed architecture for hate speech detection and harassment identification model.

To prepare the gathered dataset for the suggested model, it generally underwent many stages. The dataset was gathered and formatted to satisfy the research objective, and different data preprocessing methods were used to clean the data. Subsequently, as demonstrated in Appendix 2,2.2, text data is represented with vectors using word representation techniques. The study uses the word2vec model to represent text data as vectors. The data preparation and preprocessing steps for the development of harassment identification and hate speech detection are shown in Fig. 3-10.

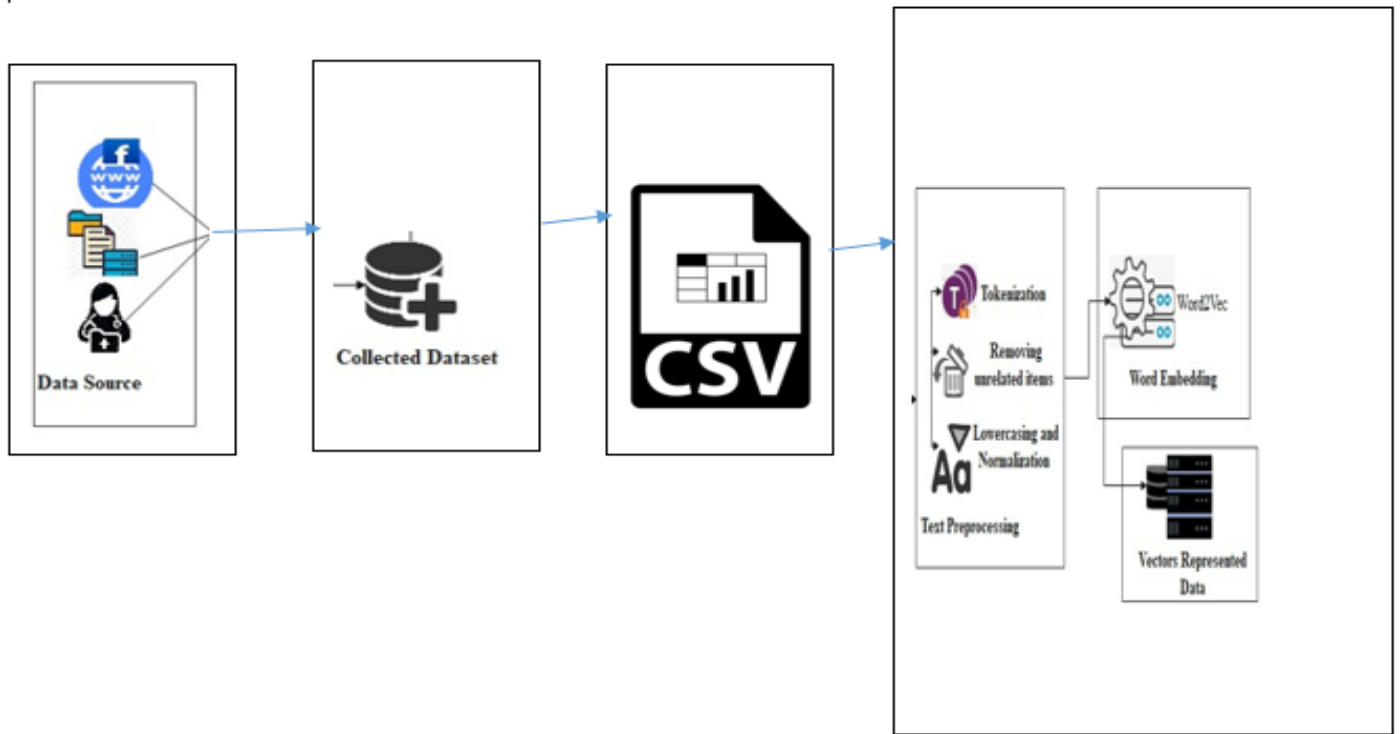


Figure 3-10: Proposed Data Preparation, Preprocessing, and Word Embedding Techniques

Following preprocessing, the distinct extracted words acquired from tokenization are represented as a feature vector using word embedding. For model development, the BERT-pre-trained model was employed. Five-fold cross-validation was used to assess the models in order to determine which detection model was the best. Based on the evaluation results, the model with the best performance is chosen to classify the data into the following categories: hate in race, not hate in race, hate in religion, not hate in religion, hate in sexism, not hate in sexism, hate in color, not hate in color, hate in disability, not hate in disability, hate in nationality, and not hate in nationality. Lastly, a prototype for the detection and identification of hate speech and harassment on social media based on protected characteristics for the Afaan Oromo language model is developed using the best-performing model that was chosen that can take new Afaan Oromo texts as input and classify the input as hate in race, not hate in race, hate in religion, not hate in religion, hate in sexism, not hate in sexism, hate in color, not hate in color, hate in disability, not hate in disability, hate in nationality, and not hate in nationality.

3.17 Proposed Feature Representation

3.17.1 Word Embeddings

The dataset is transformed into feature vectors by the suggested feature representation component. Since text is not understandable by computers, to transform text data into a numerical format that can be understood by machines, feature representation techniques are applied. The features were represented by word embeddings.

Word embeddings are word representations learned unsupervisedly whose relative similarity correlates with semantic similarity [77]. To benefit from word embedding's ability to simulate semantic similarity between words, we used it. Using unlabeled posts and comments from the dataset, we trained word2vec for the suggested detection and identification of hate speech and harassment on social media based on protected characteristics for the Afaan Oromo language. Continuous bag-of-words or skip-gram can be used to train Word2vec. Our word2vec model for this study is derived from the skip-gram model, wherein the neural network determines the context words given the target word.

3.18 Saving the Model for Future Use

It is not necessary to retrain the model for use in the future once the optimal model has been identified. Since the model answers users' questions promptly, it can be saved and used. The sample code that follows demonstrates how to load and store Python objects—such as dictionaries and lists—into a file for later use. Appendix: 2, 2.3 contains an example of code for loading and saving the model.

3.19 Tools

3.19.1 Data preparation and preprocessing tools

3.19.1.1 Facepager 4.3.3

Using web scraping and APIs, Face Pager is an open-source program that collects data from websites and social media platforms, including Facebook, Twitter, YouTube, and others. Using social media data to build the dataset facilitates the process of gathering data. The collected data can be extracted into a CSV file and stored in a shared database like SQLite. Face pager was used because it streamlines the procedure for gathering data and makes data extraction simple as a CSV file.

3.19.1.2 Scikit-learn 0.21.3

A free Python machine-learning library is called Scikit-learn. Sklearn (Skit-learn) is the most dependable and efficient Python machine-learning library. It provides a range of efficient techniques for statistical modeling and machine learning, such as dimensionality reduction, clustering, and classification, through a standardized Python interface. This library was mostly written in Python and is based on NumPy, SciPy, and Matplotlib.

3.19.1.3 Pandas 1.2.3

It is a free, open-source Python library with high-performance tools for data analysis and manipulation. Pandas provides many tools, ranging from parsing file formats to converting a whole data table into a

NumPy matrix array. It is an effective, user-friendly tool for data analysis. We used pandas to read, manipulate, publish, and handle the data frame.

3.19.1.4 Numpy 1.19.5

The primary Python library for scientific computing is called numpy. Apart from providing instruments for utilizing these arrays, it provides an efficient multidimensional array object. The text-to-numeric data conversion for the features, as well as the model testing and training, were handled by NumPy.

3.19.2 Package managers and environments

3.19.2.1 Anaconda Navigator 4.10.0

The Anaconda individual edition includes Anaconda Navigator, a graphical user interface (GUI) for managing packages and environments and running programs without requiring command-line instructions. It simplifies training with various Python and package versions, as well as setting up various configurations.

3.19.2.2 Google Colab

It is a cloud-based Jupyter Notebook environment that is free to use. We may use free GPU with the aid of colab, a free cloud service. It supports a variety of well-known machine-learning libraries that are simple to add to your notebook. Deep learning codes that require a lot of resources and time can be run effectively with its assistance and without the need to explicitly install any packages.

3.19.2.3 Jupyter Notebook 6.0.1

An open-source web program called Jupyter Notebooks enables us to create and share documents with real-time code, equations, visuals, and text. Data processing and cleaning, numerical simulation, statistical modeling, data visualization, and machine learning are some of the uses. We utilized it to implement the model because it has an anaconda navigator integrated into it.

3.19.3 Modeling tools and packages

3.19.3.1 Python 3.10

Python is a dynamically rich, object-oriented, interpreted, high-level programming language. It is an effective and simple-to-learn programming language for creating machine-learning applications to process linguistic data. Python is an excellent programming language choice for NLP jobs for several reasons. It is a great option for applications requiring natural language processing because of its straightforward syntax and

transparent semantics. In addition, Python gives programmers access to a large range of NLP tools and modules that help us with a variety of NLP-related tasks, including document classification, topic modeling, part-of-speech tagging, word vectors, and sentiment analysis.

3.19.3.2 RegEx 2.2.1

You can conduct string matching, removal, and replacement using the functions in this module. RegEx (also known as a RE) provides a list of strings that match it. This software was utilized for text preparation.

3.19.3.3 Genism 3.8.0

Genism (Generate Similar), a Python package for topic modeling, document indexing, and similarity retrieval, has a sizable dataset. It is applied in the limiting word-to-vec model of this study.

3.19.3.4 Matplotlib 3.1.1

Matplotlib is a graphing library for the Python programming language and its NumPy numerical mathematics extension. It provides an object-oriented API to embed charts into programs by using all-purpose GUI toolkits. We utilized it in this study to visualize the data and findings.

3.19.3.5 TensorFlow 2.1.0

A complete open-source platform for machine learning tasks is called TensorFlow. It has a vast, adaptable ecosystem of resources from the community, libraries, and tools that enable researchers to advance the latest developments in deep learning. TensorFlow's adaptable design enables simple compute deployment across a range of platforms (CPUs, GPUs, and TPUs), from desktops to server clusters to mobile and edge devices. It is popular over other deep learning platforms because of its more flexible but also easily understandable syntax. In addition to its versatility, TensorFlow gives the researcher additional network control and insight into the tasks carried out by a particular model.

3.19.3.6 Keras 2.3.1

Keras is one of the Python libraries that utilises TensorFlow to power a high-level neural network. It is a Python-based deep learning experimentation API. Working with Keras is faster and enables us to carry out more tests with less effort thanks to its dependable and straightforward high-level API.

3.20 Hardware Tools

The tools covered in section 3.19 above have been installed on a personal computer with an Intel® Core™ i5-4310M processor, which has a 2.70GHz CPU, 2 cores, 8 gigabytes of physical memory, and a 1TB (1000 gigabyte) hard disc storage capacity. Windows 10 Pro 64-bit is the operating system.

4 CHAPTER FOUR

RESULTS AND DISCUSSION

This chapter discusses the findings of the experiments conducted on the suggested method for identifying and detecting hate speech and harassment on social media platforms based on protected traits for the Afaan Oromo language using deep learning techniques.

The dataset contains a total of 1285 instances and 12 classes. Using this described dataset, the results of all proposed models are summarized as follows:

4.1 Result of Approach-1

To create detection and identification of harassment and hate speech on social media based on protected characteristics for the Afaan Oromo language in this study, the CNN model, and one form of RNNs-based models named LSTM, BiLSTM, and the GRU and model have been presented, and assessed. All of the deep learning models that have been presented are described in section 3.6 along with a comprehensive discussion of their capabilities and limits. The outcomes of each deep learning model proposed were outlined in this paper as follows:

4.1.1 Results of the CNN model

CNN achieved an accuracy of 78.99% under Stratified 5-fold cross-validation. The training accuracy and the validation accuracy are also well suited to unseen datasets. The model's training, validation, and loss curves are shown in Fig.4-1.

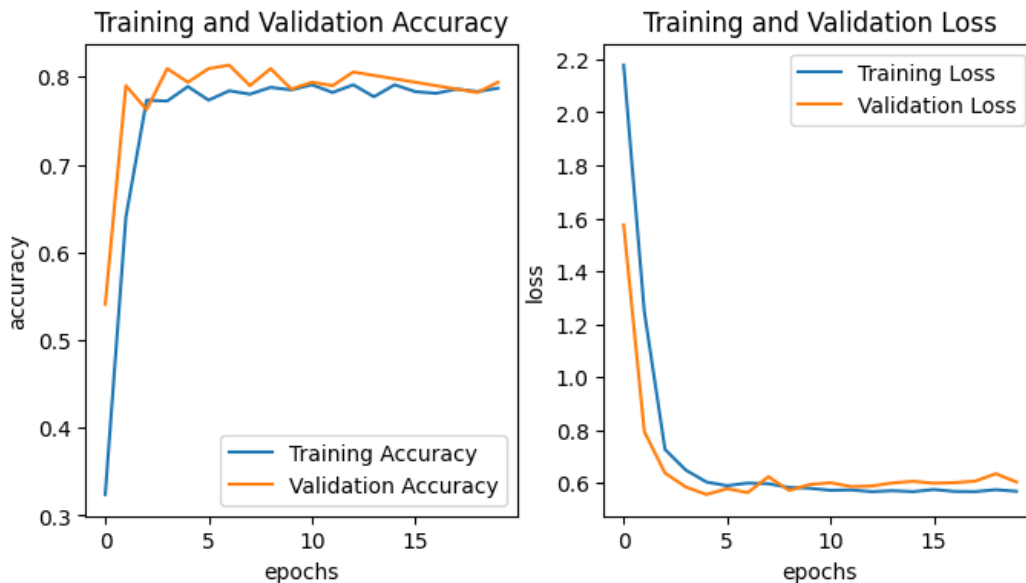


Figure 4-1: CNN Model Accuracy and Loss through Epochs.

The aforementioned depicted graph can be interpreted as follows: it demonstrates that the CNN model has been effectively trained, and its accuracy tends to increase consistently from a small value to a larger value, as evidenced by the ascending blue line, which represents the training accuracy. Furthermore, the orange line represents the validation accuracy, which also demonstrates a good level of generalization on unseen or test data. It is worth noting that the validation accuracy line consistently surpasses the training accuracy line. Conversely, the graph depicting the training and validation losses reveals a decreasing trend in both, with the validation loss line consistently positioned above the training accuracy line. This observation suggests that overfitting is encountered.

4.1.2 Results of the LSTM model

LSTM achieved an accuracy of 99.22% under Stratified 5-fold cross-validation. The training, validation, and loss curves for the model are shown in Fig. 4-2.

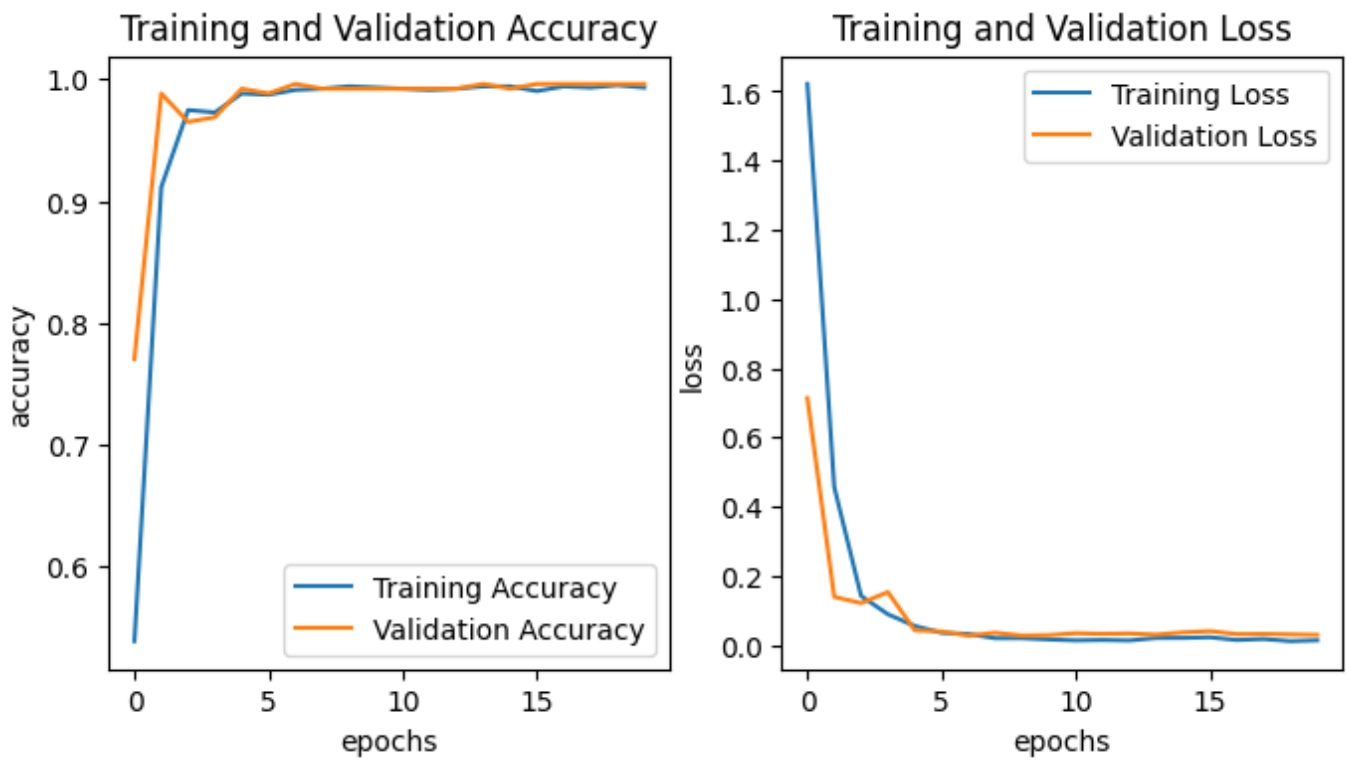


Figure 4-2: LSTM Model Accuracy and Loss through Epochs

The above-illustrated graph Figure 4-2 is interpreted as follows: it shows the LSTM model has trained well, and its accuracy tends to increase from a small number to a large number, as the blue line indicates the training accuracy, and the orange line indicates the validation accuracy is also generalized well in the unseen data or test data, and the validation accuracy is above the training accuracy. However, the training and validation loss graph indicates that the loss in training and validation is decreased, but the validation loss line exists above the training accuracy, which indicates that overfitting is encountered.

4.1.3 Results of the BiLSTM model

BiLSTM achieved an accuracy of 96.50% under Stratified 5-fold cross-validation using this model. The model's training, validation, and loss curves are shown in Fig. 4-3.

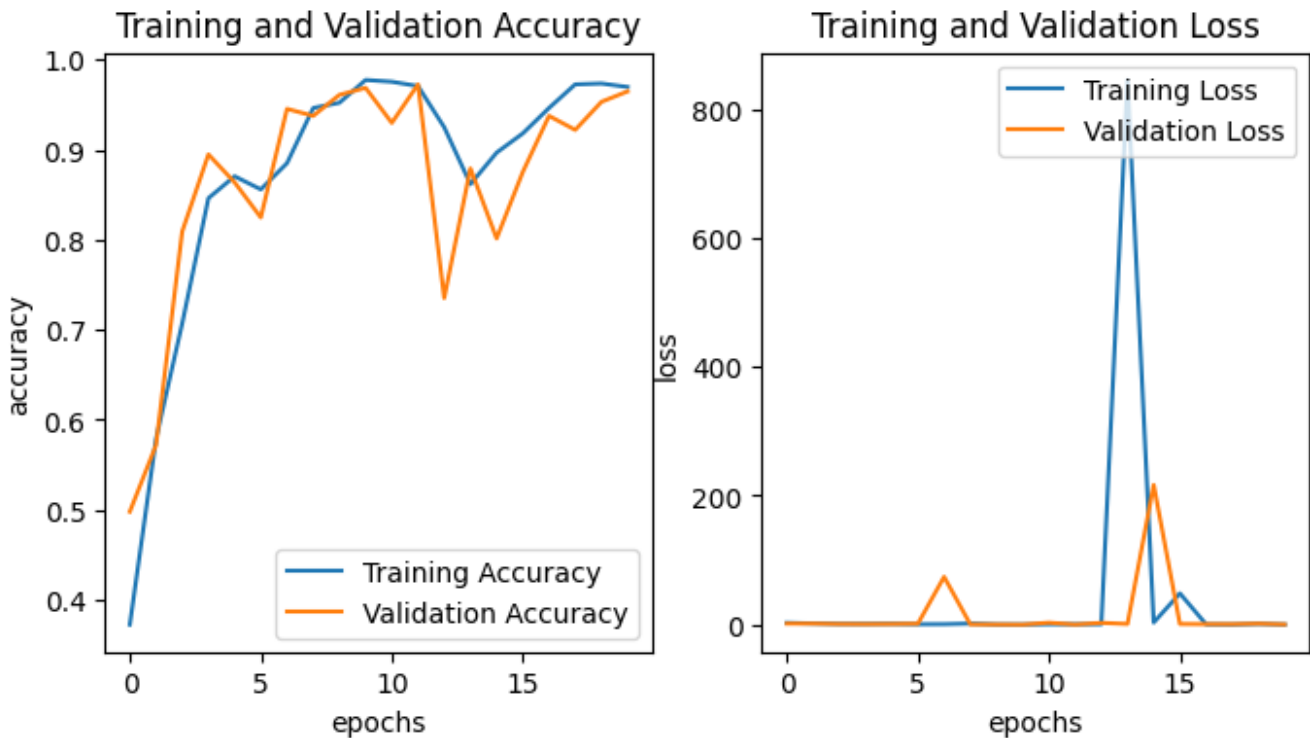


Figure 4-3: BiLSTM Model Accuracy and Loss through Epochs

Figure 4-3's above graph has the following explanation: The training accuracy line is continuously higher than the validation accuracy line at the end of the training process, which could indicate that the model is overfitting. The BiLSTM model validation accuracy rises and falls with the training accuracy line during the training process. Based on the validation accuracy fluctuating with the training accuracy, it appears that the model's

performance on the validation data varies during training. This shows that the model is not consistently performing well when applied to new, untested data.

The training accuracy line is consistently above the validation accuracy line at the end, suggesting that the model performs better on the training data than the validation data.

The model is learning from the training data and broadly generalizing to the validation data when the training loss and validation loss coincide in the early stages of training. This demonstrates that the model is generating precise forecasts and effectively encapsulating the fundamental patterns. However, the validation loss increases with training and reaches a point at which it exceeds the training loss. This divergence between the two losses suggests that the model may be starting to overfit the training set. It is probably less effective on unknown data (higher validation loss) because the training set contains too many specific examples and too much noise.

If, later in training, the validation loss starts to converge and align with the training loss again, the model stabilizes and adjusts its parameters to better generalize to new data. This convergence demonstrates how the model is getting better at finding a balance between recognizing the patterns in the training data and generalizing to new examples.

4.1.4 Results of the GRU model

GRU scored an accuracy of 99.61% Stratified 5-fold cross-validation success rate with this model. The model's training, validation, and loss curves are shown in Fig. 4-4.

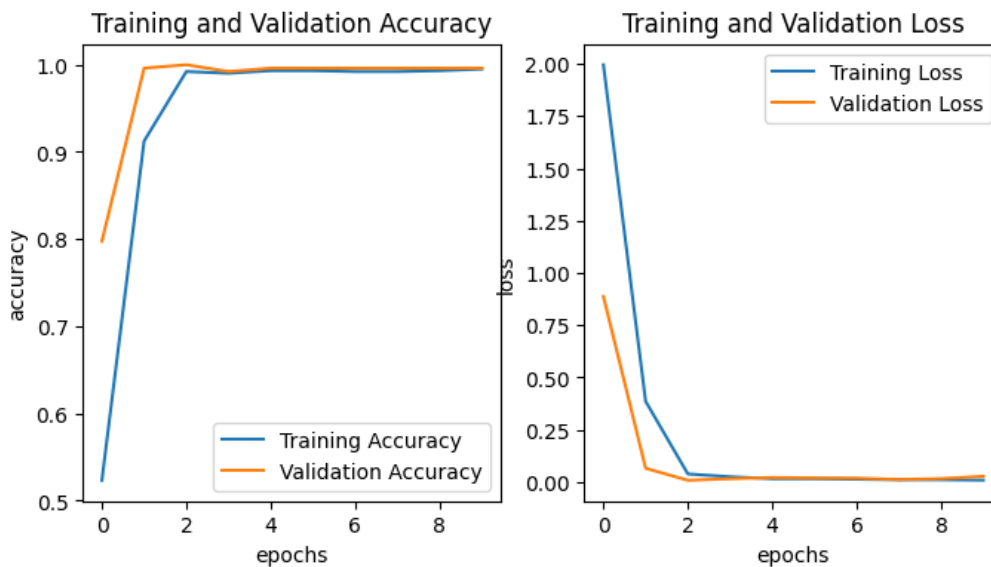


Figure 4-4: GRU Model Accuracy and Loss through Epochs

The GRU model is depicted in the aforementioned illustrated graph in Figure 4-4. The training accuracy starts out small and steadily rises over time. This demonstrates how the model is improving its ability to identify hate speech instances accurately through learning from the training set.

The validation accuracy consistently beats the training accuracy during the training process. This indicates that the model is generalizing well to new data because it performs better on the validation set than on the training set. A higher validation accuracy demonstrates the model's ability to effectively capture the underlying patterns of hate speech and predict new, unseen instances of it.

The training loss gradually decreases from its initial high level. This demonstrates that the model is getting better over time at minimizing the error, or mismatch, between the true labels in the training data and its predictions. The decreasing training loss suggests that the model may be adjusting its parameters to better fit the training data. The training loss and validation loss coincide at specific epochs. This demonstrates that the model is generalizing to the validation set in addition to overfitting to the training set. The model is recognizing the underlying patterns without becoming overly tuned to the training set, as evidenced by the alignment of the two loss curves.

In the end, the validation loss line is just slightly above the training loss line. This suggests that the model performs slightly worse on the validation set than it does on the training set. The tiny difference between the two losses shows that the model is still generalizing well and is not overfitting significantly.

This graph's increasing training accuracy and declining training loss indicate that the GRU model is improving with time. The model's consistently higher validation accuracy and the alignment of the validation and training losses suggest that it is successfully capturing the relevant hate speech patterns and generalizing well to new cases. There is a slight difference between the training and final validation losses.

Table 4-1: Summary result of each model

| | CNN | | LSTM | | BiLSTM | | GRU | |
|--------------|----------|--------|----------|--------|----------|--------|----------|--------|
| Stratified | Accuracy | Loss | Accuracy | Loss | Accuracy | Loss | Accuracy | Loss |
| 5-fold CV | 78.99 % | 0.6286 | 99.22 % | 0.0611 | 96.50 % | 0.4253 | 99.61 % | 0.0275 |

4.2 Result of Approach-2

As the models suggested in Approach-1 have encountered overfitting and small datasets, Approach-2 with BERT-pretrained and CNN (Convolutional Neural Network) models was employed.

4.2.1 Results of BERT Pre-trained Model

The BERT Pre-trained model score an accuracy of 98.83%. The model's training, validation, and loss curves are shown in Fig.4-5

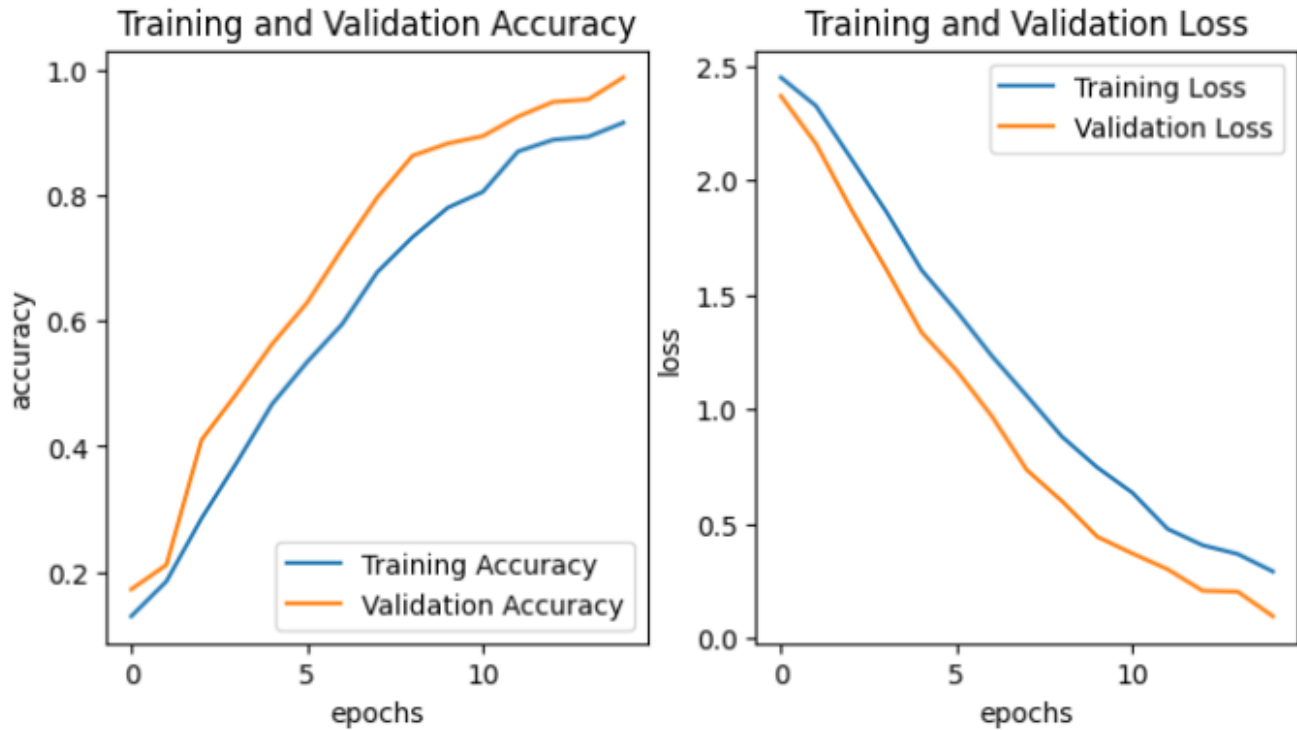


Figure 4-5: BERT Pre-trained model Accuracy and Loss through Epochs.

The graph above in Figure 4-5 can be understood as follows: it displays the BERT pre-trained model, and the training accuracy line is slightly elevated compared to the validation accuracy line. The fact that the model's accuracy during training and validation is rising suggests that it is learning and developing over time.

With a tiny gap, the validation accuracy line is continuously above the training accuracy line. This implies that the model is outperforming the training set on the validation set and generalizing well to new data. A higher validation accuracy demonstrates the model's ability to effectively capture the underlying patterns of hate speech and predict new, unseen instances of it.

The validation loss line and the training loss line are separated by a tiny elevation gap. This implies that the model is experiencing a slightly greater loss or mismatch between its predictions and the true labels on the training data when compared to the validation data. The gap shows that the model is not overfitting to the training set because the validation loss is significantly smaller.

Overall, this graph's growing training and validation accuracy indicates that the BERT pre-trained model is improving with time. The smaller gap between the training and validation accuracy lines and the higher

validation accuracy shows that the model generalizes well to new instances and effectively captures the relevant hate speech patterns. It seems that there isn't a major overfitting issue with the model because the training loss is only slightly higher than the validation loss.

4.2.2 Results of CNN Pre-trained Model

The CNN Pre-trained model score an accuracy of 98.44% . The model's training, validation, and loss curves are shown in Fig.4-6.

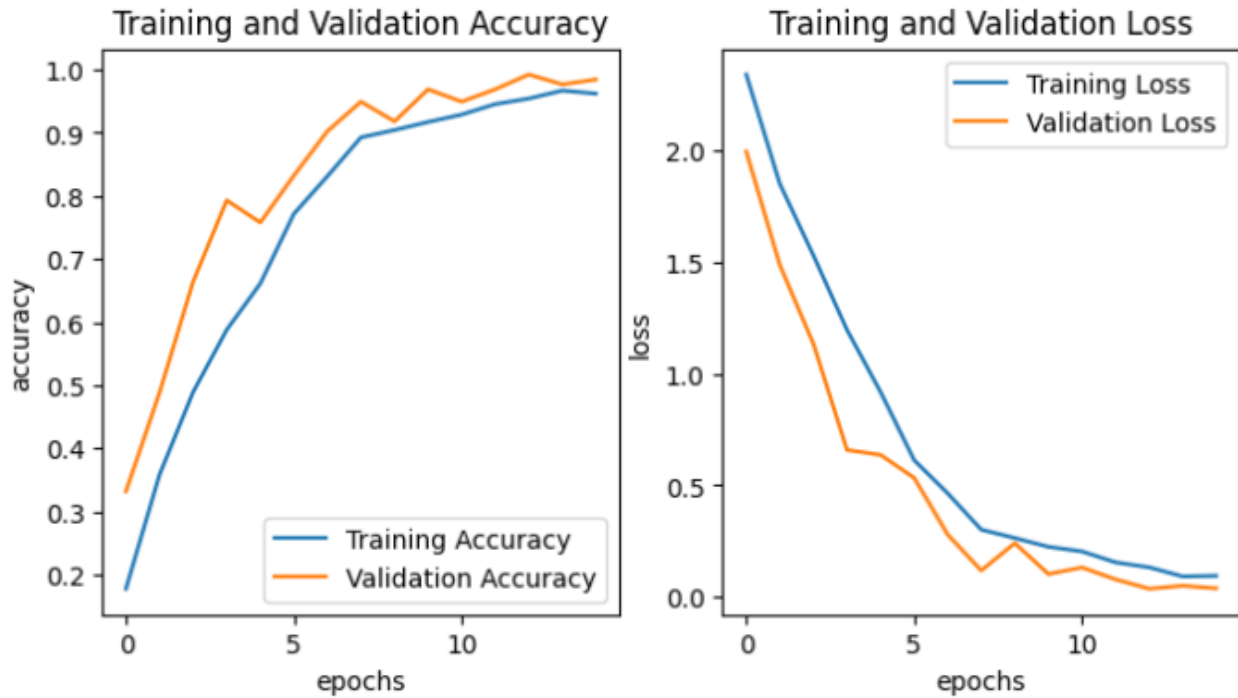


Figure 4-6: CNN with pre-trained model Accuracy and Loss through Epochs

The graph depicted in Figure 4-6 illustrates a CNN-pretrained model, with the training accuracy line being marginally higher than the validation accuracy line. The model appears to be learning and evolving, based on the fact that its accuracy is increasing during training and validation.

The training accuracy line and the validation accuracy line are separated by a narrow gap. This suggests that the model is generalizing well to new data and outperforming the training set on the validation set. A higher validation accuracy demonstrates the model's ability to effectively capture the underlying patterns of hate speech and predict new, unseen instances of it.

The training loss line is positioned above the validation loss line with a small elevation gap. This implies that the model is experiencing a slightly greater loss or mismatch between its predictions and the true labels on the

training data when compared to the validation data. The gap shows that the model is not overfitting to the training set because the validation loss is significantly smaller.

Overall, this graph's increasing training and validation accuracy indicates that the CNN model is improving with time. The smaller gap between the training and validation accuracy lines and the higher validation accuracy shows that the model generalizes well to new instances and effectively captures the relevant hate speech patterns. The model does not seem to be severely overfitted, despite the possibility that there is still room for improvement in terms of minimizing the error or mismatch in the training data, as shown by the slightly higher training loss relative to the validation loss.

4.3 Hyperparameter Tuning

The grid search hyperparameter tuning approach is employed in this work to determine the optimal hyperparameter for the neural network models that are suggested. The results stated in section 4.1 were obtained by applying the hyperparameter tuning experimentation presented in Table 4-2.

Table 4-2: Hyperparameter for all Proposed Models

| <i>Models</i> | <i>Epochs</i> | <i>Batch size</i> | <i>Activation</i> | <i>Optimizers</i> | <i>Learning rate</i> | <i>Filters</i> | <i>Kernel size</i> | <i>Hidden Layers</i> |
|---------------|---------------|-------------------|-------------------|-------------------|----------------------|----------------|--------------------|----------------------|
| <i>CNN</i> | <i>20</i> | <i>64</i> | <i>Relu</i> | <i>Adam</i> | <i>0.001</i> | <i>128</i> | <i>7</i> | <i>2</i> |
| <i>LSTM</i> | <i>20</i> | <i>120</i> | <i>softmax</i> | <i>Adam</i> | <i>0.001</i> | <i>128</i> | <i>7</i> | <i>2</i> |
| <i>BiLSTM</i> | <i>20</i> | <i>120</i> | <i>Relu</i> | <i>Adam</i> | <i>0.001</i> | <i>-</i> | <i>-</i> | <i>2</i> |
| <i>GRU</i> | <i>10</i> | <i>128</i> | <i>Relu</i> | <i>Adam</i> | <i>0.001</i> | <i>128</i> | <i>7</i> | <i>2</i> |

4.4 Discussions

As explained in Section 2.8, some research has been done for Afaan Oromoo for hate speech text detection on social media. A. Ababa [3] classified the text into binary classes using machine learning approaches and achieved high accuracy with an SVM of 96%. Whereas, G. O. Ganfure [15] performed a comparative study using deep learning techniques that classified the text into four classes with CNN and BiLSTM and achieved the same F1-score of 87%. Also, the I.J. and O.F. Science [39] conducted a similar study on afaan oromo hate speech detection using machine learning methods that classified the text into two classes and scored an F1-score of 64% with LSVM.

Lastly, T. M. Ababu and M. M. Woldeyohannis [7] investigate hate speech detection and classification using deep learning algorithms that only consider four semantic areas, classify the text into eight classes, and achieve accuracy with SVM = 0.82% and BiLSTM = 0.84%. However, the researcher does not consider other classes such as color, disability, and nationality, and also, hyperparameter tuning and overfitting handling techniques were not applied. The study, in addition to the findings of [7] hate speech detection and classification for four semantic areas with six semantic areas, was performed with the highest accuracy of the BERT pre-trained model of 98.83 %.

Therefore, a recent study that considered six semantic areas and classified the text into twelve classes was conducted. The major objective of this study was to create methods for detecting and identifying harassment and hate speech on social media based on protected characteristics. The dataset was collected using Facebook and Google Forms. Since the collected datasets are small, a deep learning technique was applied to the prepared hate speech and harassment Afaan Oromo language dataset. Because deep learning models—like CNNs, RNNs, and BERT—are better able to identify subtle patterns in hate speech. These models can automatically extract features from unprocessed text data, which makes them ideal for tasks involving the detection of hate speech. They are skilled at analyzing enormous amounts of data, making accurate generalizations, and identifying complex patterns. Pre-trained models, like BERT, enable transfer learning by assisting them in acquiring rich language representations.

To the best of our knowledge, using deep learning techniques on the provided hate speech and harassment dataset, this study is the first to propose the detection and identification of harassment and hate speech on social media based on protected characteristics of the Afaan Oromo language. We used a hate speech and harassment dataset created from freshly gathered Afaan Oromo texts from Facebook, Twitter, and local society platforms to implement the models. We used text preprocessing, including data cleaning, tokenization, and normalization, before the model's implementation. Characters with the same meaning but various spellings were

standardized. To create the models, we trained word2vec on word embeddings for the feature representation.

The recommended evaluation metrics are used to evaluate the models (see Section 3.9, "Model Evaluation Methods"). Using the supplied hate speech and harassment dataset and the pre-trained models, two models are put into practice and assessed. We put CNN and BERT pre-trained models into practice. To assess how well the CNN and BERT pre-trained models perform in comparison to the suggested models, both models are applied. Using 5-fold cross-validation, both models are evaluated. To address the overfitting issues, techniques such as L2 regularization and cross-validation were employed. To increase the training dataset, a Bert-pretrained model was applied. The BERT pre-trained model with a 5-fold evaluation yields the best model performance. The BERT pre-trained model with word2vec outperformed the other models on our dataset of hate speech and harassment, according to the 5-fold testing trial findings (Figure 4-5). We employed hyperparameter tuning, such as spatial dropout1D, along with a 0.5 dropout rate to enhance performance and reduce the issue of overfitting. Additionally, early stopping is used to help the model stop learning at the most effective epochs during model training. To find the best hyperparameter combination that offers high accuracy and minimal validation error, further hyperparameter tuning is done as shown in (Table 4-2).

Finally, the proposed detection and identification of harassment and hate speech on social media based on protected features for the Afaan Oromo remained the best performance, with an accuracy of 98.83% attained by the BERT pre-trained model. The Bert model has components such as a transformer layer, a self-attention layer, and a hidden state layer. A feed-forward neural network layer and a self-attention layer combine to form the transformer layer, which is an essential part of the transformer architecture. The feed-forward layer applies non-linear transformations to the outputs, and the self-attention layer computes attention weights between words. Transformers use multiple-layer stacking to simulate hierarchical structures and intricate relationships in input sequences. Contextualized representations for subsequent tasks, such as classification or sequence labeling, are found in the hidden state layer, which is the intermediate. Among those, the hidden state layers are the main useful components in BERT because they perform a deep contextualized representation of the input tokens.

However, fine-tuning procedures like removing or altering the particular hidden state layer of the model and assessing the impact on performance, using different attention mechanisms, or changing the input representations were carried out to ascertain which specific architecture, component, feature, layer, or attribute of a BERT model is significantly contributing to achieving high or low performance. The components that significantly contribute to the model's performance are identified by contrasting the updated models' performance with that of the original model. By doing this, the result shown in Figure 4-5 was achieved [75]. From comparing the performance of the CNN and BERT pre-trained models, we can see that the BERT pre-trained model outperforms the CNN model. Finally, the experiment supports the hypothesis that the BERT pre-

trained model outperforms other deep learning approaches for the proposed detection and identification of harassment and hate speech on social media based on protected characteristics of the Afaan Oromo language. Therefore, in this research, BERT has the highest accuracy, and it outperforms other algorithms with an accuracy of 98.83%.

5 CHAPTER FIVE

CONCLUSION AND FUTURE WORKS

5.1 Conclusion

This study suggested detecting and identifying hate speech and harassment on social media by using deep learning techniques based on Afaan Oromo's protected linguistic features. Initially, texts written in Afaan Oromo were used to create the hate speech and harassment dataset. The dataset is preprocessed in several ways before being encoded as a feature vector. The proposed models are developed using feature representation based on word embedding. The reason word embedding was chosen is that it can represent words according to their semantic contexts. Additionally, we trained a word2vec using the Afaan Oromo texts. Word2vec is trained using the skip-gram model, which is useful for modeling uncommon words even in the lack of a substantial amount of data. Two models for the experiment were developed using the harassment and hate speech datasets. Together with the pre-trained model, the recommended CNN and BERT pre-trained models are used for model comparison. The 5-fold cross-validation model comparison revealed that the BERT-pretrained model performed better than all other models. The BERT-pre-trained model had a 98.83% accuracy rate. In general, we introduced a deep learning model that can recognize and identify harassment and hate speech in the Afaan Oromo language on social media. Following that the research questions and the specific objective of the study is answered and met. Encouraging a safe and healthy online environment requires addressing hate speech and harassment on social media. To effectively identify and filter out hate speech and harassment on a variety of languages and platforms, more study and cooperation are required. The main contribution of the researcher are as follows : - develop a labeled hate speech dataset for afaan oromoo language from social media such as twitter and facebook, we increase the number of category or classes to twelve classes by considering six thematic areas such as race,religion, sexism, color, disability , nationality , and applying transfer learning techniques using the BERT pretrained model.

5.2 Future Works

To further improve performance, future research can use social media data to create large, unique, pre-trained word embeddings. Based on protected features of the Afaan Oromo language, the study implemented and experimented with recurrent neural networks for the detection and identification of harassment and hate speech on social media. Subsequent research endeavors may evaluate the efficacy of the CapsNet pre-trained model in detecting hate speech and identifying harassment on non-textual data, including audio, video, and images. Additionally, by generating datasets for Ethiopian languages, the model can be assessed using multilingual data.

Reference

- [1] T. Anderson, “Challenges and Opportunities for use of Social Media in Higher Education,” *J. Learn. Dev.*, vol. 6, no. 1, pp. 6–19, 2019, doi: 10.56059/jl4d.v6i1.327.
- [2] A. Ababa, “Department of Computer Science Hate Speech Detection Framework from Social Media Content: The Case of Afaan Oromoo Language Lata Guta kanessaa A Thesis Submitted to the Department of Computer Science in Partial Fulfilment for the Degree of Master of Scie,” 2021.
- [3] P. Burnap and M. L. Williams, “Us and them: identifying cyber hate on Twitter across multiple protected characteristics,” *EPJ Data Sci.*, vol. 5, no. 1, 2016, doi: 10.1140/epjds/s13688-016-0072-6.
- [4] P. Sekyere and B. Asare, “An Examination Of Ethiopia’s Anti -Terrorism Proclamation On Fundamental Human Rights,” *Eur. Sci. Journal, ESJ*, vol. 12, no. 1, p. 351, 2016, doi: 10.19044/esj.2016.v12n1p351.
- [5] B. Liu, “Sentiment analysis and subjectivity,” *Handb. Nat. Lang. Process. Second Ed.*, pp. 627–666, 2010.
- [6] T. Fund, “No Title,” pp. 1–11.
- [7] T. M. Ababu and M. M. Woldeyohannis, “Afaan {O}romo Hate Speech Detection and Classification on Social Media,” *Proc. Thirteen. Lang. Resour. Eval. Conf.*, no. June, pp. 6612–6619, 2022, [Online]. Available: <https://aclanthology.org/2022.lrec-1.712>
- [8] S. G. Tesfaye and K. K. Tune, “Automated Amharic Hate Speech Posts and Comments Detection Model Using Recurrent Neural Network,” *Res. Sq.*, pp. 1–14, 2020, [Online]. Available: https://www.researchsquare.com/article/rs-114533/latest?utm_source=researcher_app&utm_medium=referral&utm_campaign=RESR_MRKT_Researcher_inbound
- [9] THE FEDERAL DEMOCRATIC REPUBLIC OF ETHIOPIA, “Hate Speech and Disinformation Prevention and Suppression Proclamation,” *Fed. Negarit Gaz.*, vol. 39, no. I, pp. 8205–8234, 2015.
- [10] I. Gagliardone, “Mapping and Analysing Hate Speech Online,” *SSRN Electron. J.*, 2015, doi: 10.2139/ssrn.2601792.
- [11] M. M. Hager and L. A. W. Journal, “Harassment and Constitutional Tort: The Other Jurisprudence HARASSMENT AND CONSTITUTIONAL TORT :,” vol. 16, no. 2, 1999.
- [12] S. K. Mohapatra, S. Prasad, D. K. Bebartha, T. K. Das, K. Srinivasan, and Y. C. Hu, “Automatic hate speech detection in english-odia code mixed social media data using machine learning techniques,” *Appl. Sci.*, vol. 11, no. 18, 2021, doi: 10.3390/app11188575.
- [13] M. Corazza *et al.*, “Comparing different supervised approaches to hate speech detection,” *CEUR Workshop Proc.*, vol. 2263, 2018, doi: 10.4000/books.aaccademia.4772.
- [14] 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.
- [15] G. O. Ganfure, “Comparative analysis of deep learning based Afaan Oromo hate speech detection,” *J. Big Data*, vol. 9, no. 1, 2022, doi: 10.1186/s40537-022-00628-w.
- [16] Z. Mossie and J. Wang, “SOCIAL NETWORK HATE SPEECH,” pp. 41–55, 2018.
- [17] P. Analysis, S. Thermal, P. Chicken, E. Incubator, and T. E. Storage, “August 2021 Adama, Ethiopia,” no. August, 2021.
- [18] B. Hepple, “The New Single Equality Act in Britain,” *Equal Rights Rev.*, vol. 5, pp. 11–24, 2010.
- [19] N. Aulia and I. Budi, “Hate speech detection on Indonesian long text documents using machine learning approach,” *ACM Int. Conf. Proceeding Ser.*, pp. 164–169, 2019, doi: 10.1145/3330482.3330491.
- [20] P. Fortuna and Sergio Nunes, “A Survey on Automatic Detection of Hate Speech in TextA Survey on Automatic Detection of Hate Speech in Text,” *ACM Trans. Internet Technol.*, vol. 20, no. 2, 2020.
- [21] B. G. Alhogbi, “Tracking hatred: An international dialogue on hate speech in the social media,” 2017, [Online]. Available: <https://www.unaoc.org/2015/12/hate-speech-part-3/>
- [22] E. Bertoni, “Hate Speech Under the American Convention on Human Rights,” *ILSA J. Int’l Comp. L.*, vol. 12, pp. 569–574, 2005.
- [23] Ministry of Women Children and Youth Affairs, “Combined 4th and 5th Periodic Reports of the Federal Democratic Republic of Ethiopia to the United Nations Committee on the Rights of the Child (2006 – 2011),” vol. 43, no. April, 2012, [Online]. Available: <http://www2.ohchr.org/english/bodies/crc/docs/CRC.C.ETH.4-5.doc>
- [24] W. Akram and R. Kumar, “A Study on Positive and Negative Effects of Social Media on Society,” *Int. J. Comput. Sci. Eng.*, vol. 5, no. 10, pp. 351–354, 2017, doi: 10.26438/ijcse/v5i10.351354.
- [25] B. S. Kemal, “Bilingual Social Media Text Hate Speech Detection For Afaan Oromo and Amharic Languages Using Deep Learning Baharudin,” *Biling. Soc. Media Text Hate Speech Detect. Afaan Oromo Amharic Lang. Using Deep Learn. Baharudin*, vol. 1, pp. 250–281, 2023.
- [26] A. Muhammad, N. Wiratunga, and R. Lothian, “Contextual sentiment analysis for social media genres,” *Knowledge-Based Syst.*, vol. 108, pp. 92–101, 2016, doi: 10.1016/j.knosys.2016.05.032.
- [27] A. Al-Hassan and H. Al-Dossari, “Detection of Hate Speech in Social Networks: a Survey on Multilingual Corpus,” pp. 83–100, 2019, doi: 10.5121/csit.2019.90208.
- [28] A. E. D. Mousa and B. Schuller, “Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis,” *15th Conf. Eur. Chapter Assoc. Comput. Linguist. EACL 2017 - Proc. Conf.*, vol. 2, pp. 1023–1032, 2017, doi: 10.18653/v1/e17-1096.
- [29] “Orcid id: 0000-0001-9987-2797,” vol. 90, no. 380, pp. 0–1.
- [30] Y. Y. AKLILU, “Exploring Neural Word Embeddings,” 2019.

- [31] J. Daba, “Bidirectional English-Afaan Oromo Machine Translation Using Hybrid Approach,” no. November, 2013.
- [32] A. T. S. To, “School of Graduate Studies College of Natural Sciences Department of Computer Science School of Graduate Studies College of Natural Sciences,” no. March, 2013.
- [33] G. Tulu, “Bidirectional Amharic-Afaan Oromo Machine Translation Using Hybrid Approach,” no. March, 2020, [Online]. Available: <http://etd.aau.edu.et/handle/123456789/22101>
- [34] A. Gaydhani, V. Doma, S. Kendre, and L. Bhagwat, “Detecting Hate Speech and Offensive Language on Twitter using Machine Learning : An N-gram and TFIDF based Approach”.
- [35] F. Del Vigna, A. Cimino, and F. D. Orletta, “Hate me , hate me not : Hate speech detection on Facebook Hate me , hate me not : Hate speech detection on Facebook,” no. January, 2017.
- [36] I. Aljarah *et al.*, “Intelligent detection of hate speech in Arabic social network: A machine learning approach,” *J. Inf. Sci.*, vol. 47, no. 4, pp. 483–501, 2021, doi: 10.1177/0165551520917651.
- [37] H. Faris, I. Aljarah, M. Habib, and P. A. Castillo, “Hate speech detection using word embedding and deep learning in the Arabic language context,” *ICPRAM 2020 - Proc. 9th Int. Conf. Pattern Recognit. Appl. Methods*, no. March, pp. 453–460, 2020, doi: 10.5220/0008954004530460.
- [38] S. S. Aluru, B. Mathew, P. Saha, and A. Mukherjee, “Deep Learning Models for Multilingual Hate Speech Detection,” pp. 1–16, 2020, [Online]. Available: <http://arxiv.org/abs/2004.06465>
- [39] I. Journal and O. F. Science, “Detection of Hate Speech Text in Afan Oromo Social Media using Machine Learning Approach,” pp. 2567–2578, 2021.
- [40] M. O. Ibrohim and I. Budi, “ScienceDirect A Dataset Dataset and and Preliminaries Preliminaries Study Study for for Abusive Abusive Language Language Detection Detection in Indonesian Social Media in Indonesian Social Media,” *Procedia Comput. Sci.*, vol. 135, pp. 222–229, 2018, doi: 10.1016/j.procs.2018.08.169.
- [41] H. M. Saleem, K. P. Dillon, S. Benesch, and D. Ruths, “A Web of Hate: Tackling Hateful Speech in Online Social Spaces,” 2017, [Online]. Available: <http://arxiv.org/abs/1709.10159>
- [42] D. Benikova, M. Wojatzki, and T. Zesch, “What does this imply? examining the impact of implicitness on the perception of hate speech,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10713 LNAI, pp. 171–179, 2018, doi: 10.1007/978-3-319-73706-5_14.
- [43] K. Florio, V. Basile, M. Polignano, P. Basile, and V. Patti, “Time of your hate: The challenge of time in hate speech detection on social media,” *Appl. Sci.*, vol. 10, no. 12, 2020, doi: 10.3390/APP10124180.
- [44] E. Bassignana, V. Basile, V. Patti, and D. Informatica, “Hurtlex : A Multilingual Lexicon of Words to Hurt,” 2016.
- [45] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, “Hate Speech Detection with Comment Embeddings,” 2015.

- [46] H. Watanabe, M. Bouazizi, and T. Ohtsuki, "Hate Speech on Twitter : A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection," *IEEE Access*, vol. 6, pp. 13825–13835, 2018, doi: 10.1109/ACCESS.2018.2806394.
- [47] M. A. Fauzi and A. Yuniarti, "Ensemble Method for Indonesian Twitter Hate Speech Detection," no. July, pp. 294–299, 2018, doi: 10.11591/ijeecs.v11.i1.pp294-299.
- [48] K. K. Kiilu, G. Okeyo, R. Rimiru, and K. Ogada, "Using Naïve Bayes Algorithm in detection of Hate Tweets .," no. July, 2018, doi: 10.29322/IJSRP.8.3.2018.p7517.
- [49] T. Davidson, D. Warmesley, M. Macy, and I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language," no. Icwsn, pp. 512–515, 2017.
- [50] L. Hilte, E. Lodewyckx, B. Verhoeven, and W. Daelemans, "A Dictionary-based Approach to Racism Detection in Dutch Social Media," 2005.
- [51] M. Zampieri, "Detecting Hate Speech in Social Media".
- [52] B. Gambäck and U. K. Sikdar, "Using Convolutional Neural Networks to Classify Hate-Speech," no. 7491, pp. 85–90, 2017.
- [53] S. Biere and M. B. Analytics, "Hate Speech Detection Using Natural Language Processing Techniques," 2018.
- [54] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep Learning for Hate Speech Detection in Tweets," no. 2.
- [55] J. Qian, A. Bethke, Y. Liu, E. Belding, and W. Y. Wang, "A Benchmark Dataset for Learning to Intervene in Online Hate Speech," 2017.
- [56] K. L. Knight, "Study/Experimental/Research Design: Much More Than Statistics," vol. 45, no. 1, pp. 98–100, 2010.
- [57] "A Gentle Introduction to the Bag-of-Words Model. In A Gentle Introduction to the Bag-of-Words Model (p. 1).", [Online]. Available: <https://machinelearningmastery.com/gentle-introduction-bag-words-model/>
- [58] A. I. Kadhim, "Term Weighting for Feature Extraction on Twitter: A Comparison between BM25 and TF-IDF," *2019 Int. Conf. Adv. Sci. Eng. ICOASE 2019*, pp. 124–128, 2019, doi: 10.1109/ICOASE.2019.8723825.
- [59] A. Rendel, R. Fernandez, R. Hoory, and B. Ramabhadran, "Using continuous lexical embeddings to improve symbolic-prosody prediction in a text-to-speech front-end," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2016-May, pp. 5655–5659, 2016, doi: 10.1109/ICASSP.2016.7472760.
- [60] Fekadu Eshetu Hunde, "Designing and Developing Bilingual Chatbot for Assisting Ethio-Telecom Customers on Customer Services, Adama Science and Technology University," no. May, 2021, doi:

10.24297/j.cims.2022.12.148.

- [61] M. Nuruzzaman and O. K. Hussain, "A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks," *Proc. - 2018 IEEE 15th Int. Conf. E-bus. Eng. ICEBE 2018*, no. June 2020, pp. 54–61, 2018, doi: 10.1109/ICEBE.2018.00019.
- [62] K. Ramakrishna and M. A. Maha, "A Review on Chatbot Design and Implementation Techniques," *Int. Res. J. Eng. Technol.*, vol. 7, no. 2, pp. 2791–2800, 2020, [Online]. Available: www.irjet.net
- [63] H. Liang, X. Sun, Y. Sun, and Y. Gao, "Text feature extraction based on deep learning: a review," *Eurasip J. Wirel. Commun. Netw.*, vol. 2017, no. 1, pp. 1–12, 2017, doi: 10.1186/s13638-017-0993-1.
- [64] L. Notes, "Westminster Research Are Deep Learning Approaches Suitable for Natural Language Processing? This is an author's accepted manuscript of a paper presented at NLDB 2016: 21st International Conference on Applications of Natural Language to Information Syst," 2016.
- [65] P. Kandpal, K. Jasnani, R. Raut, and S. Bhorge, "Contextual chatbot for healthcare purposes (using deep learning)," *Proc. World Conf. Smart Trends Syst. Secur. Sustain. WS4 2020*, pp. 625–634, 2020, doi: 10.1109/WorldS450073.2020.9210351.
- [66] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," pp. 1–9, 2014, [Online]. Available: <http://arxiv.org/abs/1412.3555>
- [67] C. Yao *et al.*, "SPECIAL SECTION ON BIG DATA ANALYTICS FOR SMART AND CONNECTED HEALTH A Convolutional Neural Network Model for Online Medical Guidance," vol. 4, 2016.
- [68] H. Saleh, A. Alhothali, and K. Moria, "Detection of Hate Speech using BERT and Hate Speech Word Embedding with Deep Model Detection of Hate Speech using BERT and Hate Speech," *Appl. Artif. Intell.*, vol. 37, no. 1, 2023, doi: 10.1080/08839514.2023.2166719.
- [69] F. H. A. Shibly, U. Sharma, and H. M. M. Naleer, *Performance Comparison of Machine Learning and Deep Learning Algorithms in Detecting Online Hate Speech*, vol. 473. Springer Nature Singapore, 2023. doi: 10.1007/978-981-19-2821-5_59.
- [70] C. Schaffer, "Technical Note: Selecting a Classification Method by Cross-Validation," *Mach. Learn.*, vol. 13, no. 1, pp. 135–143, 1993, doi: 10.1023/A:1022639714137.
- [71] G. L. Team, "What are Cross-Validation and its types in Machine learning_ Great Learning?," 2020. <https://www.mygreatlearning.com/blog/cross-validation/>
- [72] V. L. Kouznetsova, J. Li, E. Romm, and I. F. Tsigelny, "Finding distinctions between oral cancer and periodontitis using saliva metabolites and machine learning," *Oral Dis.*, vol. 27, no. 3, pp. 484–493, 2021, doi: 10.1111/odi.13591.
- [73] P. Sharma, "Different Types of Cross-Validations in Machine Learning - Analytics Vidhya.," 2022. <https://www.mygreatlearning.com/blog/cross-validation/>
- [74] Kurtis, "Fighting Overfitting With L1 or L2 Regularization: Which One Is Better?," 2021.

- [75] J. Nabi, “Hyper-parameter Tuning Techniques in Deep Learning _ by Javaid Nabi _ Towards Data Science,” 2019. <https://towardsdatascience.com/hyper-parameter-tuning-techniques-in-deep-learning-4dad592c63c8>
- [76] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, no. M1m, pp. 4171–4186, 2019.

1 APPENDICES

Appendix 1: Sample Code

1.1 Loading the dataset

```
# Loading the dataset
import pandas as pd
data=pd.read_excel('/content/drive/MyDrive/AsmeResearch1/Original
dataset2.xlsx')
data.head()
```

1.1 Text Normalization

```
# Normalization of words
data['Context']= [string.replace("otoo", "osoo") for string in data['Context']]
data['Context']= [string.replace("ugaa", "dhugaa") for string in data['Context']]
data['Context']= [string.replace("dabree", "darbee") for string in data['Context']]
data['Context']= [string.replace("mini", "miti") for string in data['Context']]
data['Context']= [string.replace("ykn", "Yookiin") for string in data['Context']]
data['Context']= [string.replace("Waahee", "Waa'ee") for string in data['Context']]
data['Context']= [string.replace("eennu", "eenyu") for string in data['Context']]
data['Context']= [string.replace("haga", "hanga") for string in data['Context']]
data['Context']= [string.replace("ega", "erga") for string in data['Context']]
data['Context']= [string.replace("jiha", "ji'a") for string in data['Context']]
data['Context']= [string.replace("bahe", "ba'e") for string in data['Context']]
data['Context']= [string.replace("yaahe", "yaa'e") for string in data['Context']]
print(data['Context'])
```

1.2 Sample code of data cleaning

```
# Data Cleaning
def clean_text(text):
    text = re.sub(r'\s+', ' ', text)
    text=re.sub('[. *?]', '',text)
    text=re.sub("\n", "",text)
    text = re.sub(r'[^\\w\\s]', "",text)
    text = re.sub(r'\s+', ' ', text).strip()# Cleaning the whitespaces
# text=re.sub('[%s?]'% re.escape(string.punctuation),"",text)
    text=re.sub("\w*\d\w*", "",text)
    text = re.sub('([@0-9_+])|[^\\w\\s]|#|http\\S+', "", text)
    return text
clean= lambda x:clean_text(x)
cleaned_Data=pd.DataFrame(data.Posts.apply(clean))
cleaned_Data
```

1.3 Sample code of text tokenization

```
# Tokenization
# Here text is Tokenized into individual words
import nltk
from nltk.tokenize import RegexpTokenizer
regexp = RegexpTokenizer("\\w+")
cleaned_Data['cleared_Context']=cleaned_Data['Context'].apply(regexp.tokenize)
# Processed_Data.head(50)
token_data=cleaned_Data['cleared_Context']
token_data
```

1.4 Sample code of word2vec implementation

```
# Word2vec implementation
from gensim.models import Word2Vec
W2V= Word2Vec(array_file,min_count=1,workers=4, vector_size=300, sg=1, window=5)
print(W2V)
    W2V.train(array_file, total_examples=W2V.corpus_count, epochs=10)
W2V1=W2V.wv.save_word2vec_format('/content/drive/MyDrive/AsmeResearch1/Word2vecModel.
bin', binary=True)
trainedModel =
KeyedVectors.load_word2vec_format('/content/drive/MyDrive/AsmeResearch1/Word2vecModel.bin
', binary= True)
voc = list(trainedModel.key_to_index)
voc_size=len(voc)
voc_size
```

1.5 Sample of List of stop words in afaan oromo Language

| | | | | | | |
|---------|----------|-----------|-----------|---------|------------|-----------|
| itti | kanaafuu | natti | akkasumas | sitti | isii | ofirratti |
| narraa | tanaaf | jala | na | kan | irratti | ofirraa |
| akka | immoo | gubbaa | nu | illee | wanta | jira |
| ati | hogguu | hanga | nuti | ala | asii | narratti |
| ammo | alatti | of | akkuma | sun | haa | keessaa |
| kana | siin | jara | nurraa | isaa | irraa | turuuf |
| an | ittuu | henna | sana | Kanaaf | kam | kootu |
| kanaafi | tanaafuu | duuba | silaa | eegasii | akkamii | kanneen |
| ani | amma | kee | kennaa | ishiif | sin | wanti |
| tun | waan | bira | siif | koo | baayeen | miti |
| booddee | warra | ishiirraa | eega | iseen | wal | sirraa |
| keessa | ta'ullee | teenya | yommuu | yoona | si | Miti |
| inni | aan | isatti | jedhaman | kana | jirutti | irraan |
| keessan | ishii | keenya | nan | yookiin | anan | naaf |
| gara | kanan | kami | gad | maalif | akkamiitu | isiin |
| isaan | baayee | akkamii | atoo | moo | takkaaf | Yookaan |
| naf | ni | kami | akkan | isa | akkamittin | Hoo |

1.6 Sample Code for all Proposed Model

1.6.1 CNN Model Sample Code

```
kfold = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
cvscores = []
for train, test in kfold.split(X_value, Y_value):
    Cnnmodel = Sequential()
    Cnnmodel.add(EmbeddingMat)
    Cnnmodel.add(Conv1D(filters=128, kernel_size=7, activation='relu'))
    Cnnmodel.add(MaxPooling1D(pool_size=5))
    Cnnmodel.add(Flatten())
    # Cnnmodel.add(Conv1D(filters=128, kernel_size=5, activation='relu'))
    # Cnnmodel.add(MaxPooling1D(pool_size = 3))
    Cnnmodel.add(Flatten())
    Cnnmodel.add(Dense(300, activation='relu'))
    Cnnmodel.add(Dense(200, activation='relu'))
    Cnnmodel.add(Dense(12, activation='softmax'))
    # sgd = SGD(learning_rate=0.01, decay=1e-6, momentum=0.9, nesterov=True)
    Cnnmodel.compile(loss = 'sparse_categorical_crossentropy',optimizer=Adam(learning_rate=0.001), metrics =
['accuracy'])

    historycnn =
Cnnmodel.fit(X_value[train],Y_value[train],epochs=20,verbose=1,validation_data=(X_value[test],Y_value[test]
), batch_size=64)

    score = Cnnmodel.evaluate(X_value[test],Y_value[test], verbose=1)

    print("%s: %.2f%%" % (Cnnmodel.metrics_names[1], score[1]*100))
    cvscores.append(score[1] * 100)
print("%.2f%% (+/- %.2f%%)" % (np.mean(cvscores), np.std(cvscores)))
```

1.6.2 LSTM model sample code

```
kfold = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
cvscores = []
for train, test in kfold.split(X_value, Y_value):
    lstmmodel=Sequential()
    lstmmodel.add(EmbeddingMat)
    lstmmodel.add(LSTM(1000, return_sequences=True))    lstmmodel.add((LSTM(500,
        return_sequences=False)))    # read about Regulazation techniques use it if you want to use it
    lstmmodel.add(Dense(12, activation='softmax'))
    lstmmodel.compile(loss='sparse_categorical_crossentropy',optimizer='Adam', metrics=['accuracy'])
    lstmhistory=lstmmodel.fit(X_value[train],Y_value[train], epochs=20, batch_size=120,
validation_data=(X_value[test],Y_value[test]),verbose=1)#,callbacks=[es,mc])

    score = lstmmodel.evaluate(X_value[test],Y_value[test], batch_size=64)

    print("%s: %.2f%%" % (lstmmodel.metrics_names[1], score[1]*100))
    cvscores.append(score[1] * 100)
print("%.2f%% (+/- %.2f%%)" % (np.mean(cvscores), np.std(cvscores)))
```

1.6.3 BiLSTM model sample code

```
kfold = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
cvscores = []
for train, test in kfold.split(X_value, Y_value):

    BilstmModel = keras.Sequential()
    BilstmModel.add(EmbeddingMat)
    BilstmModel.add(tf.keras.layers.Bidirectional(LSTM(500, return_sequences=True, activation='relu')))
    BilstmModel.add(tf.keras.layers.Bidirectional(LSTM(256, return_sequences=False, activation='relu')))
    BilstmModel.add(Dense(12, activation='softmax'))
    BilstmModel.compile(loss='sparse_categorical_crossentropy', optimizer=Adam(learning_rate=0.001),
metrics=['accuracy'])

    historybilstm=BilstmModel.fit(X_value[train],Y_value[train],epochs=20,
validation_data=(X_value[test],Y_value[test]), verbose=1, batch_size=120)
    score = BilstmModel.evaluate(X_value[test],Y_value[test], verbose=1)
    print("%s: %.2f%%" % (BilstmModel.metrics_names[1], score[1]*100))
    cvscores.append(score[1] * 100)
print("%.2f%% (+/- %.2f%%)" % (np.mean(cvscores), np.std(cvscores)))
```

1.6.4 GRU model sample code

```
kfold = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
cvscores = []
for train, test in kfold.split(X_value, Y_value):
    BiGruModel = Sequential()
    BiGruModel.add(EmbeddingMat)
    BiGruModel.add(tf.keras.layers.Bidirectional(GRU(464, return_sequences=True, activation='relu')))
    BiGruModel.add(tf.keras.layers.Bidirectional(GRU(256, return_sequences=False, activation='relu')))
    BiGruModel.add(Dense(12, activation='softmax'))
    BiGruModel.compile(loss = 'sparse_categorical_crossentropy',optimizer =
Adam(learning_rate=0.001),metrics = ['accuracy'])

    historyGRU = BiGruModel.fit(X_value[train], Y_value[train],
validation_data=(X_value[test],Y_value[test]),epochs=10, verbose=1, batch_size=128)

    score = BiGruModel.evaluate(X_value[test],Y_value[test], verbose=1)

    print("%s: %.2f%%" % (BiGruModel.metrics_names[1], score[1]*100))
    cvscores.append(score[1] * 100)
print("%.2f%% (+/- %.2f%%)" % (np.mean(cvscores), np.std(cvscores)))
```

2 Appendix 2: BERT Pre-trained Model

2.1 Sample code for BERT Pre-trained Model

```
#Loading the BERT Pretrained model
from transformers import TFBertModel
from transformers import BertTokenizer, TFBertModel
from tensorflow.keras.layers import Input, Conv1D, LSTM, Bidirectional, GRU, Dense
from tensorflow.keras.models import Model
Model = TFBertModel.from_pretrained('bert-base-cased') # bert base model with pretrained weights
#defining input layers,Hidden Layer,output layer for input_ids and attn_masks
input_ids = tf.keras.layers.Input(shape=(256,), name='input_ids', dtype='int32')
attn_masks = tf.keras.layers.Input(shape=(256,), name='attention_mask', dtype='int32')
bert_embds = model.bert(input_ids, attention_mask=attn_masks)[1] # 0 -> activation layer (3D), 1 ->
pooled output layer (2D)
intermediate_layer = tf.keras.layers.Dense(512, activation='relu',
name='intermediate_layer')(bert_embds)
output_layer = tf.keras.layers.Dense(12, activation='softmax', name='output_layer')(intermediate_layer)
# softmax -> calcs probs of classes
BERT Pretrained_model = tf.keras.Model(inputs=[input_ids, attn_masks], outputs=output_layer)
BERT Pretrained_model.summary()
```

2.2 Sample of Embedding Particular words in word2vec Representation

```
# Embedding Particular Words in word2vec
trainedModel.get_vector('mootummaa')
array([ 0.0004829 ,  0.29656544,  0.08796004,  0.08934715,  0.028344 ,
        -0.17975666,  0.14381617,  0.3833357 ,  0.06106214, -0.11175472,
         0.07954981, -0.1090292 ,  0.00814906,  0.02248891, -0.09678948,
        -0.09610216,  0.05452825, -0.05410339, -0.00972384, -0.08686047,
        -0.15099576,  0.05096466,  0.15640458,  0.07704843,  0.18327172,
         0.05063627, -0.19326888,  0.00609517, -0.0922816 , -0.18784584,
         0.02987543, -0.09461341, -0.06613402, -0.04707714,  0.02773323,
         0.16031499,  0.04563878, -0.10416348,  0.01241101, -0.18511006,
        -0.03453348,  0.0300621 ,  0.00338035, -0.09594163,  0.02126907,
         0.2878082 ,  0.00980495,  0.21472831, -0.08018392,  0.2427959 ,
         0.02645852,  0.00041745, -0.17954448,  0.04872487, -0.07843781,
         0.15962835,  0.07698739,  0.05274283,  0.03715935, -0.02425543,
        -0.03444716,  0.006368 , -0.03310902,  0.02868694,  0.03233947,
         0.03940826, -0.03451903,  0.075711 , -0.14495288, -0.08051153,
        -0.02806479,  0.11911911,  0.08719892, -0.13038512, -0.05622246,
         0.13334054, -0.19523624,  0.02530321, -0.06743927,  0.174003 ,
        -0.04924 , -0.15221287,  0.04982901,  0.3092744 ,  0.07712586,
```


2.3 Sample code for saving the Model

```
BERT Pretrained_model.save('/content/drive/MyDrive/Transfer learning/BERT_model')
```

2.4 Sample code for Hate speech Detection and Harassment Classification

```
# Prediction
def prepare_data(input_text, tokenizer):
    token = tokenizer.encode_plus(
        input_text,
        max_length=256,
        truncation=True,
        padding='max_length',
        add_special_tokens=True,
        return_tensors='tf'
    )
    return {
        'input_ids': tf.cast(token.input_ids, tf.float64),
        'attention_mask': tf.cast(token.attention_mask, tf.float64)
    }
def make_prediction(model, processed_data, classes=[]):
    probs = BERT_model.predict(processed_data)[0]
    return classes[np.argmax(probs)]
```

2.5 Sample form data collection using Google forms

Detection and Identification of Harassment on social media for Afaan Oromoo language using Deep learning

Gaafii siniif dhiyaaten seera ser-luga Afaan Oromootin Guutachuun naaf Deegara

asmeadane049@gmail.com [Switch account](#)

Not shared

* Indicates required question

1. Himoota /jechoota Arabso fi miti Arabso tahaan lamummaan Naaf Tarressa ? *

Your answer

2. Himoota /jechoota Arabso fi miti Arabso tahaan Bifaan Naaf Tarressa ? *

Your answer

3. Himoota /jechoota Arabso fi miti Arabso tahaan Qamaa midhamtootaf Naaf Tarressa ? *

Your answer

Submit [Clear form](#)