

# Transformer based approach for inappropriate content detection in Urdu



By

**Jawad Ahmad**

**Fall-2019-MS-IT 321047**

Supervisor

**Dr Rabia Irfan**

**Department of Computing**

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Information Technology (MS IT)

In

School of Electrical Engineering & Computer Science (SEECS) ,

National University of Sciences and Technology (NUST),

Islamabad, Pakistan.

(June 2023)

## THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis entitled "Transformer based approach for inappropriate content detection in Urdu" written by JAWAD AHMAD, (Registration No 00000321047), of SEECS has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: \_\_\_\_\_  \_\_\_\_\_

Name of Advisor: \_\_\_\_\_ Dr. Rabia Irfan \_\_\_\_\_

Date: \_\_\_\_\_ 17-Jul-2023 \_\_\_\_\_

HoD/Associate Dean: \_\_\_\_\_

Date: \_\_\_\_\_

Signature (Dean/Principal): \_\_\_\_\_

Date: \_\_\_\_\_

## Approval

It is certified that the contents and form of the thesis entitled "Transformer based approach for inappropriate content detection in Urdu" submitted by JAWAD AHMAD have been found satisfactory for the requirement of the degree

Advisor : Dr. Rabia Irfan

Signature:  \_\_\_\_\_

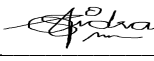
Date: 17-Jul-2023

Committee Member 1:Dr. Nazia Perwaiz

Signature:  \_\_\_\_\_

17-Jul-2023

Committee Member 2:Dr. Sidra Sultana

Signature:  \_\_\_\_\_

Date: 17-Jul-2023

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

# Dedication

This thesis is dedicated to all the deserving children who do not have access to quality education especially young girls.

## Certificate of Originality

I hereby declare that this submission titled "Transformer based approach for inappropriate content detection in Urdu" is my own work. To the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics, which has been acknowledged. I also verified the originality of contents through plagiarism software.

Student Name: JAWAD AHMAD

Student Signature: 

# Acknowledgments

Glory be to Allah (S.W.A), the Creator, the Sustainer of the Universe. Who only has the power to honour whom He please, and to abase whom He please. Verily no one can do anything without His will. From the day, I came to NUST till the day of my departure, He was the only one Who blessed me and opened ways for me, and showed me the path of success. There is nothing which can payback for His bounties throughout my research period to complete it successfully.

**Jawad Ahmad**

# Contents

<b>1</b>	<b>Introduction and Motivation</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Raggedness of Inappropriate Data . . . . .	3
1.3	Self-Identification of Inappropriate Data . . . . .	4
1.4	Problem Statement . . . . .	5
1.5	Research Objectives . . . . .	7
1.6	Thesis Organization . . . . .	7
<b>2</b>	<b>Literature Review</b>	<b>8</b>
2.1	Background . . . . .	8
2.1.1	Characteristics of Urdu Script and the Associated Challenges . . . . .	10
2.1.2	Urdu Text Literature . . . . .	11
2.2	Inappropriate Data Identification Methods . . . . .	14
2.2.1	Machine Learning Approaches . . . . .	15
2.2.2	Deep Learning Approaches . . . . .	16
2.3	Critical Analysis . . . . .	19
<b>3</b>	<b>Background</b>	<b>23</b>
3.1	Deep Learning Models . . . . .	24
3.1.1	Gated Recurrent Unit . . . . .	24
3.1.2	Transformers Encoder . . . . .	25

## CONTENTS

3.1.3	Long Short-Term Memory . . . . .	28
3.1.4	Bidirectional LSTM . . . . .	28
3.2	Word Embedding . . . . .	29
3.2.1	Vector Representaion of Words . . . . .	30
3.2.2	Word Embeddings in Urdu . . . . .	30
<b>4</b>	<b>Design and Methodology</b>	<b>32</b>
4.1	Dataset . . . . .	32
4.1.1	Dataset Collection . . . . .	32
4.1.2	Dataset Annotation and Statistics . . . . .	33
4.2	Dataset Pre-Processing . . . . .	34
4.3	Methodology . . . . .	35
4.4	Experimental Setup . . . . .	37
4.4.1	Sequence Normalization . . . . .	38
4.4.2	Sequential Model Layering . . . . .	38
<b>5</b>	<b>Implementation and Results</b>	<b>40</b>
5.1	Evaluation Metrics . . . . .	40
5.2	Testing . . . . .	41
5.3	Results Comparison . . . . .	42
5.4	Discussions . . . . .	45
<b>6</b>	<b>Conclusion and Future Work</b>	<b>47</b>
6.1	Summary . . . . .	47
6.2	Challenges . . . . .	48
6.3	Limitations . . . . .	49
6.4	Future Work . . . . .	49
6.5	Applications . . . . .	50



# List of Figures

1.1	Language Map of Pakistan . . . . .	2
1.2	Flow Chart of Training and Testing of an NLP Model . . . . .	5
3.1	Gated Recurrent Unit . . . . .	26
3.2	Transformer Encoder . . . . .	27
3.3	Long Short-Term Memory . . . . .	29
3.4	Bi-directional LSTM . . . . .	30
4.1	Dataset Distribution. . . . .	34
4.2	Methodology . . . . .	36
5.1	Results Comparison Without Using Word to Vector Layer. . . . .	42
5.2	Results Comparison Using Word to Vector Layer. . . . .	43
5.3	Accuracy Comparison Graph. . . . .	44
5.4	Recall Comparison Graph. . . . .	44
5.5	Precision Comparison Graph. . . . .	44
5.6	F1-Score Comparison Graph. . . . .	45
5.7	Loss Comparison Graph. . . . .	45
5.8	Accuracy Graph (Transformer Encoder) . . . . .	46
5.9	Accuracy Graph (Transformer Encoder with Word to Vector Layer) . . . . .	46

# List of Tables

2.1	Words in URDU language . . . . .	9
2.2	Loan Words from Other Languages . . . . .	11
2.3	Root Words . . . . .	11
2.4	Short Vowels . . . . .	12
2.5	Literature Summary . . . . .	21
2.6	Literature Summary Continued . . . . .	22

# Abstract

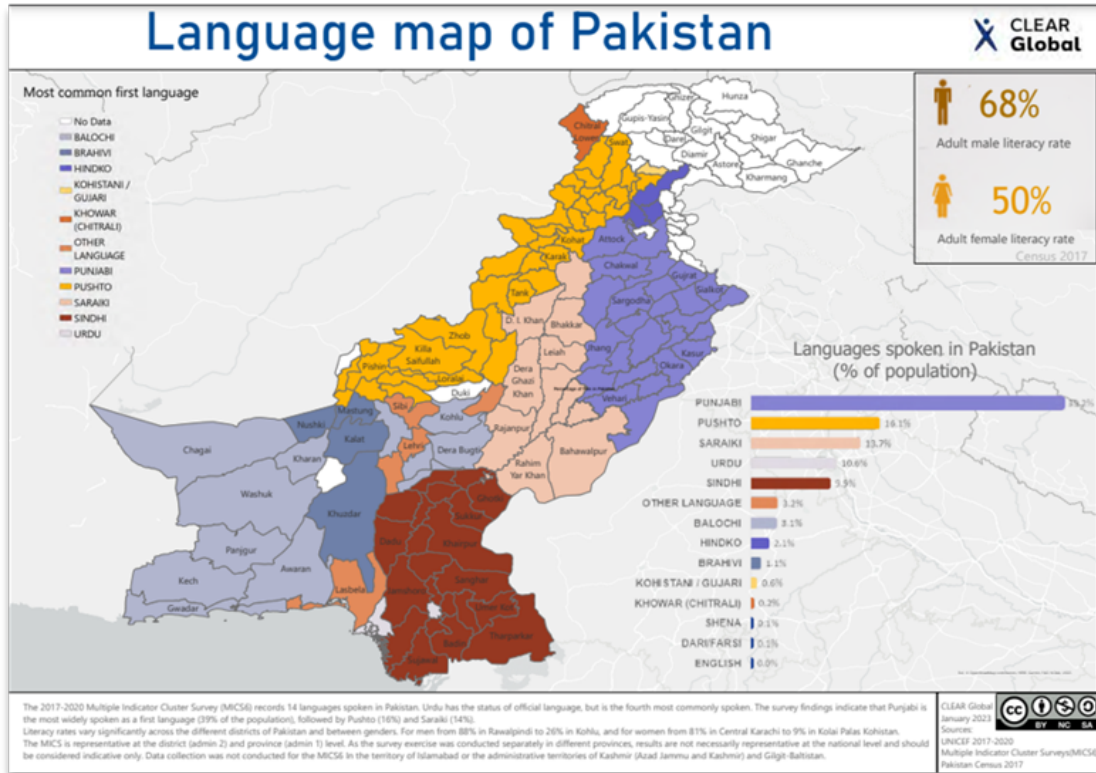
The popularity of social networking sites and online forums has increased the spread of harmful and improper content. While many research have looked into this issue in various languages, there is a big void in the literature when it comes to employing deep learning techniques in native Urdu language. This research is a continuation of Attention based Bidirectional GRU hybrid model for inappropriate content detection in Urdu Language. To improve the areas where other models have limitation (e.g. parallelization, Long-range dependency, sequential computing, positional encoding, scalability and Network) our research suggests an Attention-based Bidirectional Transformer Encoder model for recognizing objectionable content in local Urdu language to fill this gap. The effectiveness of our suggested approach is compared to the above-mentioned research, taking into account evaluation criteria, dataset size, and the word embedding layer's influence. Pre-trained Urdu Word2Vec embeddings are used for our tests. The outcomes show that our transformer-based bidirectional approach improves to 85 percentage. Our tests demonstrate the efficiency-improving power of the attention layer while also emphasizing the inadequacy of pre-trained Word2Vec embeddings for the detection of unsuitable content in Urdu datasets.

# Introduction and Motivation

The exponential growth of social media users has significantly influenced communication technology and transformed the Internet [41]. Users find it more convenient to express their thoughts and opinions in their local languages [34] [51]. Taking into account URDU is widely spoken language bridging between different cultures and areas of Pakitan as shown in 1.1. However, the widespread use of social media also exposes billions of users to cyber-crimes such as bullying, threats, and scams. Additionally, the unrestricted posting of controversial content without proper checks and balances can lead to provocation, social unrest, public outrage, manipulation of public opinion, and chaos [25]. To tackle these issues, many social media platforms rely on manual techniques, where users report problematic content and then it was flagged based on user ratings further investigated by reviewers. However, this approach has limitations as it depends on the speed and expertise of the reviewers and frequency of flagged contents [25]. To address these challenges and counter the misuse of social media, it is crucial to automatically detect, categorize, and filter controversial content before it is posted online. By implementing intelligent algorithms social media platforms can proactively identify objectionable content, ensuring a safer and more responsible online environment [51].

## 1.1 Motivation

In recent years, several incidents have occurred in Pakistan that have prompted the government to take precautionary measures to prevent uncontrollable consequences. These incidents include the defamation of political parties and their leaders, targeted



**Figure 1.1:** Language Map of Pakistan

[translatorswithoutborders.org/language-data-for-pakistan](http://translatorswithoutborders.org/language-data-for-pakistan)

harassment of famous media figures, bullying and hurting the sentiments of religious minorities, harassment of women expressing their views, and exchanges of derogatory remarks between people from India and Pakistan due to lingering bitterness from the independence war. These events highlight the challenges faced by the nation in dealing with hate speech online and emphasize the immediate need for an automated filtering system [25]. Manual identification of hate speech content is inefficient due to the large number of online users and the overwhelming volume of internet content. While significant advancements have been made in Natural Language Processing (NLP), most of the research has primarily focused on resource-rich languages like English. Machine Learning (ML) techniques have been the preferred choice for researchers in various NLP tasks such as text translation, classification, and sentiment analysis due to their impressive results and performance [43]. In recent years, Deep Learning (DL) algorithms have also been incorporated for detecting inappropriate content from user comments on social networking sites in different languages, including Turkish and Arabic. Urdu, the national language of Pakistan, is a resource-scarce language with a complex morphological

structure, unique characters, and limited linguistic resources. It is well-known that hate speech content can vary across languages, which further contributes to the limited research in this area due to the scarcity of language resources and small, labeled/unlabeled datasets available for Urdu. Addressing the issue of hate speech in Urdu requires specific attention and dedicated research efforts to develop effective automated filtering systems that can understand the nuances of the language and accurately identify inappropriate content. This necessitates the collaboration of researchers, policymakers, and technology experts to overcome the challenges posed by resource scarcity and contribute to creating a safer online environment in Urdu-speaking communities.

## 1.2 Raggedness of Inappropriate Data

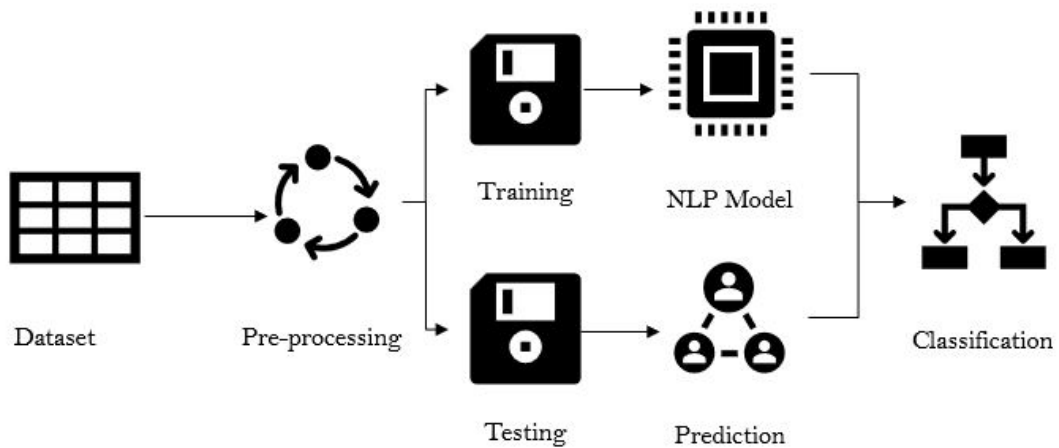
The rapid growth of digital communication platforms has brought people from diverse backgrounds closer together, enabling the exchange of ideas, information, and entertainment on a global scale. However, this surge in digital interaction has also given rise to a concerning issue - the prevalence of inappropriate content. While this issue is prevalent across various languages, it is crucial to address inappropriate content specifically in Urdu, as it is a widely spoken language with limited resources and a dearth of pre-trained models. In this motivation, we highlight the importance of tackling inappropriate content in Urdu and shed light on the significance of a novel transformer-based approach to overcome these challenges. Urdu, a language spoken by millions of people worldwide, holds immense cultural and historical significance [32]. It serves as a means of communication, literature, and artistic expression for a vast community. Unfortunately, the presence of inappropriate content in Urdu poses serious threats to the social fabric, ethics, and values associated with this language. By addressing this issue, we can protect the linguistic heritage, foster a safe digital environment, and promote healthy online interactions. Despite the rising need to combat inappropriate content in Urdu, there is a notable scarcity of work and resources dedicated to this task. Unlike some other widely spoken languages [35].

### 1.3 Self-Identification of Inappropriate Data

With the advancements in the field of data science, language understanding has made significant progress. Researchers have conducted numerous studies using Machine Learning (ML) and Deep Learning (DL) algorithms to detect inappropriate content, particularly in resource-rich languages. Approaches ranging from basic ML models to simple recurrent neural networks and more recently advanced models with transformers have been explored [10]. Transfer learning has emerged as a breakthrough in this field, where pretrained models are fine tuned for various other tasks. This not only reduces the development cycle time but also produces state-of-the-art results. In the basic classification workflow, the process starts with data collection related to the problem domain. The data is thoroughly inspected and cleaned through pre-processing and fixing methods. The cleaned dataset is then loaded into an appropriate DL framework. If any issues arise during data loading, the cleaning stage is revisited otherwise, the data is explored using conventional statistical approaches. This cycle continues to assess and test the performance of DL models [43]. In addition to model building, visualizations are developed to aid in communicating and analyzing the results, as well as generating insights about the data. The outcomes are then released and shared with the community. The stages of the classification workflow are illustrated in 1.2.

In supervised ML classification, there are two major steps: feature extraction and classification. Various techniques are employed to extract features from the data, such as n-gram feature extraction. Other popular techniques include pattern matching and lexicon-based approaches. In DL supervised classification, features are learned by neural networks, and word vector word embedding models like Word2Vec are used for improved text representation. Deep Learning models have shown significant performance improvements compared to conventional ML models, especially when large amounts of data are available. While the official language of Pakistan is English, Urdu is recognized as the national language and widely spoken in many Indian provinces as well. Urdu has a unique writing script and a sophisticated lexicon structure. Its morphological composition, starting from the right and moving to the left, sets Urdu apart from other languages. However, the lack of widespread use of the Urdu script necessitates the availability of a standard dataset or corpus for conducting Urdu NLP tasks [10]. The identification of inappropriate content in Urdu is as significant as in any other

language, as it enables non-Urdu speakers to comprehend the fundamental thoughts, emotions, and perspectives of Urdu writers. Many native Urdu speakers use the Urdu script on platforms like Twitter, Facebook, and YouTube to express their feelings and ideas. Analyzing text written in Urdu is crucial for understanding the thoughts and emotions of native Urdu speakers. Existing literature in Urdu language studies mostly focuses on various aspects of natural language processing, such as sentiment analysis, news classification, and gender identification. There are limited studies that explore DL algorithms for inappropriate content detection in Urdu, with most of the emphasis on ML techniques.



**Figure 1.2:** Flow Chart of Training and Testing of an NLP Model

## 1.4 Problem Statement

Detecting inappropriate content in Urdu is a highly challenging task due to the absence of transformer-based solutions and limited reference work in this area [40]. Inappropriate content encompasses material that violates societal norms, ethical standards, or legal regulations, including hate speech, offensive language, explicit imagery, and other objectionable forms of content [47]. Transformers, such as BERT and GPT, have proven to be powerful tools in natural language processing tasks, including content moderation. However, the availability of transformer-based solutions specifically designed for inappropriate content detection in Urdu is lacking. Transformers are known for their ability to capture contextual information, understand linguistic nuances, and detect patterns



in text. The absence of transformer-based solutions tailored for Urdu hinders the development of advanced detection algorithms, significantly impacting the effectiveness of inappropriate content detection in this language. Additionally, the limited reference work and research dedicated to inappropriate content detection in Urdu further compounds the problem. While content moderation research has gained significant traction in recent years, it has predominantly focused on widely spoken languages, leaving Urdu with minimal attention. The scarcity of comprehensive studies, methodologies, and frameworks specifically tailored to Urdu content moderation restricts the progress of this field and leaves content moderators with insufficient guidance and resources to effectively detect and manage inappropriate content in Urdu. The lack of transformer-based solutions for inappropriate content detection in Urdu can be attributed to several factors. Firstly, the development of such models relies heavily on large-scale annotated datasets for training and evaluation purposes. Unfortunately, the availability of annotated datasets specifically curated for inappropriate content in Urdu is scarce [11]. The absence of labeled data impedes the training of accurate and robust transformer models that can understand the contextual nuances of the Urdu language and accurately detect inappropriate content. Furthermore, the scarcity of reference work and research in Urdu content moderation inhibits the exchange of knowledge and best practices among researchers and practitioners in the field. Without a strong research foundation, it becomes challenging to develop and refine effective algorithms, techniques, and tools for detecting inappropriate content in Urdu. The lack of reference work hampers the progress of the field and limits the ability to leverage existing knowledge and expertise to address the unique challenges of inappropriate content detection in the Urdu language. In conclusion, detecting inappropriate content in Urdu is a formidable challenge due to the absence of transformer-based solutions and limited reference work in this domain. The lack of transformer models specifically designed for Urdu content analysis hinders the development of accurate detection algorithms [11]. Additionally, the limited research and reference work impede the exchange of knowledge and restricts the advancement of effective content moderation techniques in Urdu. Bridging these gaps and dedicating resources to address the specific challenges of inappropriate content detection in Urdu is vital to creating safer digital spaces and preserving the integrity of online communities in the Urdu-speaking world.

## 1.5 Research Objectives

This research makes a valuable contribution by combining the progress made in the "Attention-based Bidirectional GRU for Inappropriate Content Detection in Urdu" study with the ongoing pursuit of improving results and proposing a novel solution. Despite the growing availability of publicly shared annotated Urdu corpora, locating domain-specific data remains a formidable challenge. The predominant utilization of ML baseline models in exploring existing datasets highlights the untapped potential for further investigation and advancement. This research endeavor aims to bridge these gaps and contribute significantly to the field with these key points.

- To improve the results of inappropriate text analysis in Urdu, design a transformer based solution.
- To investigate the effects of word embedding.
- To draw comparative analysis with baseline DL models based on suitable evaluation metrics.

## 1.6 Thesis Organization

- Chapter 2 Literature Review, represents a comprehensive background, encompassing previous studies that have tackled the subject of inappropriate content detection, the origins of text classification, and a brief exploration of ML/DL learning methods utilized within this domain.
- Chapter 3 Background will delve into the discussion of baseline (DNL), elucidating their evolution, framework, and applications, thereby enhancing the understanding of these models.
- The proposed model will be presented in Chapter 4 Design and Methodology, along with an in-depth analysis of the results and subsequent discussion in Chapter 5 Results and Discussions.
- Chapter 5 Conclusion and Future Work will encompass the conclusion drawn from the study, along with highlighting potential avenues for future research.

# Literature Review

This chapter offers comprehensive insights into research conducted on the identification of inappropriate language. It provides a concise overview of the historical background and progression of content identification techniques. Furthermore, it delves into both traditional and neural network methodologies employed in this field. Lastly, a thorough critical analysis of offensive content detection in the Urdu language will be presented.

## 2.1 Background

Social media has emerged as a global platform for individuals to express their opinions, but it has also become a breeding ground for online attacks and the spread of inappropriate content, fueled by the anonymity provided by the platform. While several studies have focused on evaluating and addressing inappropriate content in various languages, the existing standards for filtering online information and combating bigotry fall short when it comes to the Urdu language [37]. This study aims to analyze offensive and upsetting content in Urdu by proposing a novel Transformer based Bidirectional Encoder model specifically designed for identifying inappropriate content in Urdu Unicode text. Quantitative experiments were conducted using two datasets. Baseline deep learning models were used to compare the performance of the proposed model. Evaluation metrics such as Precision, Recall, F1-score, and Accuracy, as well as dataset size and the impact of the word embedding layer, were considered. Pre-trained Urdu Word2Vec embeddings were employed for this study. The study revealed that the attention layer significantly improved the model's efficiency. The attention layer enhanced the model's

ability to focus on important aspects of a sequence and handle long and variable-length sentences effectively. While previous studies have explored ML and DL algorithms for detecting inappropriate content in various languages, limited research has been conducted for Urdu, primarily due to its resource scarcity, complex morphological structure, and limited linguistic resources. This study highlights the need for advanced DL techniques and analysis specific to the Urdu language to filter and identify inappropriate content, ultimately improving communication quality and understanding for Urdu speakers on social media platforms. The identification of inappropriate language online has emerged as a prominent application of natural language processing (NLP) [30]. Detecting inappropriate speech in social media posts presents a considerable challenge due to the prevalence of informal language used by individuals on a daily basis. The interpretation of a sentence’s context can vary among individuals, as everyone has their own perspectives. Certain words may be perceived as humorous by some while being regarded as hateful by others [18], making it difficult to draw a clear distinction. Significant research has been conducted in the area of inappropriate language detection, primarily focusing on English and various other languages. In the case of the Urdu language, Roman Urdu is commonly utilized in social media posts. Roman Urdu represents Urdu words using the English alphabet, simplifying data acquisition compared to the native Urdu script. Consequently, most studies on hate speech identification in Urdu concentrate on Roman Urdu. This underscores the importance of identifying and eliminating inappropriate speech in Roman Urdu to safeguard individuals from online abuse. Previous studies have explored machine learning (ML) baseline models and a few neural networks (NN) to address the problem of offensive language detection in the Urdu script. ML models often struggle to perform well on large datasets, typically learning only specific textual features. On the other hand, deep learning (DL) models have demonstrated significantly enhanced efficiency as the dataset size increases [36].

ح	چ	ج	ث	ٹ	ت	پ	ب	ا	آ
ش	س	ڑ	ز	ڑ	ر	ذ	ڈ	د	خ
گی	کی	ق	ف	غ	ع	ظ	ط	ض	ص
			ے	ی	ہ	و	ن	م	ل

**Table 2.1:** Words in URDU language

### 2.1.1 Characteristics of Urdu Script and the Associated Challenges

The Urdu script possesses distinct language characteristics that introduce specific challenges due to its intricate nature. Let's explore the unique aspects of Urdu script and the associated difficulties it presents. Urdu is renowned for its intricate morphological structure, involving the use of diverse affixes, suffixes, and prefixes. Basic Words in URDU are shown in 2.1. This complexity adds hurdles to tasks such as word segmentation, part-of speech tagging, and morphological analysis. Traditionally, Urdu is written in the Arabic script, which presents difficulties in terms of character recognition and text processing. The Arabic script comprises characters with multiple forms based on their position within a word, demanding specialized handling for accurate analysis and processing [13]. Urdu script incorporates ligatures, which are combinations of multiple characters joined together to form a single character. These ligatures pose challenges in tasks like text normalization and segmentation, requiring specific treatment to ensure precise analysis and processing. The Urdu script encompasses characters absent in other languages, such as additional dots to distinguish similar letters. These distinct characters can complicate tasks like information retrieval, search algorithms, and text indexing. Urdu is considered a resource-scarce language, with limited linguistic resources available for tasks like machine translation, sentiment analysis, and named entity recognition. The scarcity of labeled data and language-specific tools makes it challenging to develop robust language models and systems for Urdu [38]. Acquiring domain-specific and annotated datasets in Urdu remains a challenging endeavor. Most available datasets are either small or focus solely on Roman Urdu, limiting research opportunities and impeding the development of robust models. Addressing these language characteristics and challenges demands dedicated research and development efforts to enhance language resources, build effective language processing tools, and create domain-specific datasets for Urdu. Leveraging advancements in natural language processing techniques, such as deep learning and transfer learning, holds promise in overcoming these challenges and bolstering the capabilities of Urdu language processing systems. Urdu possesses several distinctive qualities, including complex morphological characteristics, that pose challenges in data processing tasks such as stop words removal, stemming, and text normalization [37]. Urdu vocabulary incorporates words borrowed from Persian, English, Turkish, Arabic, and Sanskrit 2.2. These words are blended with Urdu's own vocabulary, forming a comprehensive Urdu dictionary

Word	Origin
أول	Arabic
آسان	Persian
کلاس	English

**Table 2.2:** Loan Words from Other Languages

Urdu has its own distinct alphabet set and employs the Nastaliq writing style, which is intricate and visually complex. Urdu has many words that share a single root but have different meanings 2.3.

علم	معلم	معظمہ
-----	------	-------

**Table 2.3:** Root Words

Urdu is highly context sensitive, and some words cannot be written without spaces between them. Reading these words together completes their meaning. However, the space between words does not define the word boundary, creating challenges in the word segmentation process. For example, the word "Curtain" in Urdu can mean پردے (Forgiveness) depending on the context. Lack of capitalization concept: Unlike English, Urdu does not have the concept of word capitalization. There is no indication of the beginning of a sentence through a capitalized letter. For example, the sentence "I have two cars" has no indication of the start of the sentence in Urdu: ہیں کاریں دو پاس میرے Hence, proper nouns and the start of a sentence are unidentifiable in Urdu. In Urdu, case markers are considered parts of speech and play a crucial role in defining sentence construction. The absence of these markers can lead to ambiguities in grammar. For example بوتل and پانی میری کی both mean "My water bottle," but the word order is different, resulting in different interpretations. Diatric marks in Urdu alter the meaning of words written with the same letters. Additionally, the pronunciation of words with similar alphabets can also vary. For example, کل with a diatric mark ُ means "all," while کل with a diatric mark َ means "yesterday." 2.4.

### 2.1.2 Urdu Text Litratue

Studies in Urdu text classification have gained significant attention in recent years as researchers aim to develop effective methods for analyzing and categorizing Urdu lan-

Pronunciation	Symbol
zabar	ˆ
zer	ˆ
pe <sup>h</sup>	ˆ
tashad	ˆ
jazam	ˆ
khari zabar	ˆ
do zabr	ˆ

Table 2.4: Short Vowels

guage content. Text classification involves automatically assigning predefined labels or categories to textual data, enabling efficient organization and retrieval of information. However, the task of Urdu text classification poses specific challenges due to the complex nature of the language, limited linguistic resources, and scarcity of annotated datasets. Researchers have explored various approaches, including traditional machine learning algorithms, deep learning models, and hybrid methods, to tackle these challenges. These studies often involve preprocessing steps like tokenization, stemming, and stop-word removal, as well as feature extraction techniques such as n-grams, TF-IDF, and word embeddings. Evaluation metrics like accuracy, precision, recall, and F1-score are used to assess the performance of classification models. The results of these studies contribute to advancing Urdu language processing and support applications such as sentiment analysis, topic modeling, and information retrieval in Urdu text. The ongoing research in Urdu text classification demonstrates the importance of developing tailored approaches for the language to improve the accuracy and effectiveness of automated text analysis in Urdu. In the realm of Urdu text classification, several noteworthy research studies have been conducted using both machine learning (ML) and deep learning (DL) techniques. Many studies have explored ML approaches such as Support Vector Machines (SVM), Naive Bayes (NB), K-Nearest Neighbors (K-NN), and Decision Trees. The classification of news articles, social media posts, and news headlines has been a common focus in these studies. [51] propose a hybrid model for the detection of inappropriate content in Urdu text, combining the power of GRU (Gated Recurrent Unit) and deep learning techniques. By leveraging the sequential nature of GRU and its ability to capture temporal dependencies, along with the representation learning capabilities of deep learning,

[51] aim to achieve a robust and accurate system for detecting inappropriate content in Urdu. The hybrid model will undergo rigorous evaluation and analysis to assess its performance, taking into account metrics such as accuracy, precision, recall, and F1-score. This research contributes to the field of natural language processing by addressing the challenges of inappropriate content detection in Urdu, offering insights into the effectiveness of the GRU hybrid model for this specific task. One study [9] delved into the use of SVM for classifying Urdu news headlines. The authors employed techniques such as normalization, stemming, and stop words elimination to preprocess the documents. They also calculated the inverse document frequency and term frequency of words from the selected corpus. Another study [8] investigated feature selection methods including Chi Statistics, Information Gain, Gain Ratio, Symmetrical Uncertain, and oneR. K-NN, Decision Trees, and NB classifiers were applied to two Urdu datasets. The analysis revealed that the Information Gain approach improved the performance of SVM and K-NN classifiers. In a separate analysis conducted by [49] on the Roman-Urdu dataset, three classifiers were examined, namely NB, Decision Tree, and K-NN. It was found that NB performed better than the other two classifiers based on evaluation metrics such as accuracy, recall, and fmeasure. The research [15] focused on the characteristics of classifying the origin of news in Urdu text. SVM, K-NN, and Decision Tree classifiers were compared using a large Urdu dataset containing 16,678 documents, predominantly news pieces from the Urdu publication "The Daily Roshni." TF-IDF weighting scheme was employed for feature selection, and the results indicated that SVM outperformed the other two classifiers in terms of accuracy. In another study [46], the authors concentrated on the multi-class classification of a Pakistani News dataset. Various ML algorithms, including Logistic Regression, SVM, NB, and Random Forest, were applied for both single-class and multi-class classification. The comparative analysis demonstrated that SVM performed best for binary classification, while Logistic Regression excelled in multi-class classification. [33] provided a benchmark Urdu dataset and evaluated various ML and DL techniques in their study. They also explored the impact of transfer learning for Urdu language using the Bidirectional Encoder Representations via Transformers (BERT) approach. The findings from their extensive comparative research indicated that a combination of feature selection, such as Normalized Difference Measure, with ML and DL classifiers yielded significantly improved efficiency. Lastly, in a study that compared ML and DL models [34], three common DL models were explored,



and results were compared with ML models using various preprocessing methods. The study concluded that Convolutional Neural Networks (CNN) outperformed other DL and ML models. Overall, these research studies contribute valuable insights into the application of ML and DL techniques for Urdu text classification, showcasing the effectiveness of different approaches and highlighting the superior performance of certain classifiers and models in specific scenarios.

## 2.2 Inappropriate Data Identification Methods

The identification of inappropriate content in Urdu has been the focus of several research studies aimed at developing effective methods to detect and filter such content. Inappropriate content can include hate speech, offensive language, and derogatory remarks. The challenge lies in the unique characteristics of the Urdu language, including its complex morphology, script adaptation, and scarcity of language resources. Researchers have employed various techniques to address this issue. Traditional machine learning algorithms, such as support vector machines and Naive Bayes classifiers, have been used for binary classification tasks. These approaches rely on features like n-grams, lexical patterns, and syntactic structures to detect inappropriate content [47]. Moreover, deep learning models have gained popularity due to their ability to capture complex linguistic patterns. Convolutional neural networks and recurrent neural networks, such as long short-term memory and gated recurrent unit, have been applied to classify inappropriate content in Urdu. These models leverage word embeddings [12], such as Word2Vec or FastText, to represent the semantic meaning of words. Additionally, researchers have explored ensemble methods and hybrid models that combine multiple classifiers or techniques to improve the accuracy and robustness of the identification process. These approaches integrate the strengths of different models, such as combining SVMs with CNNs or incorporating linguistic features with deep learning architectures [48]. Evaluation of the identification methods is typically done using standard metrics such as precision, recall, F1-score, and accuracy. Researchers also utilize annotated datasets specific to Urdu to train and evaluate their models, although the availability of such datasets can be limited. The ongoing research in inappropriate content identification in Urdu aims to enhance the performance and adaptability of existing methods [40]. It also emphasizes the importance of addressing the language-specific challenges in Urdu and improving

the quality of content moderation on platforms that utilize the Urdu language. In the past decade, there has been a lack of attention from NLP researchers towards the identification of offensive language. However, the advancements in NLP methods for various everyday tasks have shown promise in addressing the challenges associated with detecting hate speech on social networks. The difficulty lies in distinguishing violent language from other objectionable content or even harmless material that may share vocabulary overlap [42]. It is not uncommon to come across derogatory or vulgar language used in sarcastic or humorous contexts. The NLP community has been focused on internet platforms like Twitter, YouTube, Instagram, Facebook, and online blogs in their efforts to identify harmful language [24],[17],[16].

### 2.2.1 Machine Learning Approaches

Numerous studies have focused on utilizing machine learning (ML) techniques to identify inappropriate language in different languages. Commonly employed ML algorithms for this task include Support Vector Machines, Logistic Regression, K-Nearest Neighbors, and Decision Trees. Various feature selection techniques, such as lexicon-based approaches and chisquare, have been explored by multiple studies for detecting abusive or offensive language. Character n-gram and word n-gram extraction methods have also been utilized in several studies. For instance, researchers in [26], [22], and [39] employed character n-gram feature extraction methods, while some studies like [26], [17], [21] [31], and [29] explored word n-grams and their variants. In [26], the authors proposed using SVM with a linear kernel to detect inappropriate language in a Bengali dataset. Their approach considered both Unicode Bengali characters and Unicode emoticons as acceptable input. Another study [50] utilized ML classifiers such as SVM, Logistic Regression, Decision Trees, and Random Forest, along with modular cleaning of a Twitter dataset and the development of a tokenizer. In the context of Twitter, threatening comments were categorized using Naive Bayes (NB) in [20]. The International Workshop [21] and [31] focused on tasks related to identifying abusive language in social media platforms, including the detection of objectionable content, automatic classification of offensive categories, and identification of offensive targets. For these tasks, ML classifiers such as NB, Logistic Regression, SVM, and Random Forest were employed. In [29], researchers employed n-gram feature extraction techniques up to 8-gram to evaluate their impact on the aforementioned tasks. They trained three automated systems for three subtasks.

The first subtask involved using a linear SVM with uni-gram and bi-gram models, the second subtask utilized a linear SVM model combined with n-grams up to 4-gram, and the third subtask employed a Decision Trees model with uni-gram to 8-gram feature selection. The study concluded that implementing n-grams was efficient, resulting in high accuracy while being straightforward to employ. For Urdu, which lacks publicly available datasets due to being a resource-poor language, researchers in [34] created an annotated dataset of offensive language in Urdu script and made it publicly available. They thoroughly experimented with multiple ML algorithms on both Urdu and Roman Urdu script datasets, providing a detailed comparative analysis using the n-gram feature selection method. The study revealed that regression-based ML models achieved the best accuracy, although they required more time to create. In a study by [35], multi-class classification of Roman Urdu content extracted from YouTube comments was presented. N-gram and TF-IDF techniques were employed for feature selection, and L1 and L2 normalization approaches were applied for data normalization. To balance classes with unequal instances, SMOTE was used. Logistic Regression, NB, SVM, and SGD classifiers were compared, and the results showed that SVM combined with n-gram feature selection, L2 normalization, and TF-IDF feature values outperformed other models. Additionally, the researchers developed a web interface called YT Monitor, which scrapes user comments from a given keyword or URL link and classifies them into respective hate content categories.

### 2.2.2 Deep Learning Approaches

Deep learning approaches have been increasingly utilized for inappropriate content detection in Urdu, leveraging their ability to capture complex linguistic patterns and semantic representations. These methods aim to automatically identify and classify offensive language, hate speech, and other forms of inappropriate content in Urdu text. Several deep learning architectures have been employed for this purpose: CNNs have been applied to learn local and global features from Urdu text. They utilize convolutional layers to extract meaningful patterns and features at different scales. These features are then passed through fully connected layers for classification. CNNs have shown effectiveness in capturing discriminatory features in text and achieving competitive performance in Urdu content classification tasks. RNNs, particularly Long Short-Term Memory and Gated Recurrent Unit models, have been widely used for sequence modeling and classi-

fication in various languages, including Urdu. RNNs are capable of capturing contextual information and long-range dependencies in text. They process sequences of words, modeling the relationships between words and capturing the context necessary for detecting inappropriate content. Transformer models, such as the popular BERT (Bidirectional Encoder Representations from Transformers) [47], have also been applied to inappropriate content detection in Urdu [40]. These models utilize self-attention mechanisms to capture the contextual information of words in a text, allowing them to understand the relationships between words and their surroundings. Transformers have demonstrated impressive performance in various natural language processing tasks and have shown promise in Urdu content classification as well. Hybrid models combine multiple deep learning architectures or incorporate linguistic features alongside neural networks [7]. For instance, a combination of CNN and LSTM layers can capture both local and global features while modeling the sequential nature of text. Additionally, linguistic features like part-of-speech tags, sentiment analysis scores, or lexicon-based features can be integrated with deep learning models to enhance their performance in identifying inappropriate content [1]. Evaluation of these deep learning approaches is typically conducted using appropriate metrics such as precision, recall, F1-score, and accuracy. The models are trained on labeled datasets specifically annotated for inappropriate content in Urdu. However, the scarcity of such datasets remains a challenge, necessitating further efforts to collect and curate larger and more diverse datasets for robust model training and evaluation. The ongoing research in deep learning approaches for inappropriate content detection in Urdu aims to improve the accuracy, efficiency, and adaptability of models in order to effectively combat the spread of inappropriate and harmful content in the Urdu language space [1]. NLP poses challenges in identifying inappropriate language due to characteristics such as spelling and grammar errors, contextual ambiguity, polysemy, and semantic variations. To address this, a hybrid deep learning model combining Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory was proposed in [14] for automatically identifying inappropriate language. The study focused on two application scenarios: search engine query completion recommendations and user chats in messenger applications. In [23], DL models were employed to automatically recognize Facebook postings from users in need of emergency help due to domestic violence situations. Data from Facebook posts related to domestic violence was collected to develop a multi-class identification model. Multiple DL classifiers and word embeddings

were experimented with, and a Gated Recurrent Unit classifier along with word embeddings achieved the best accuracy. Authors in [19] used an ensemble method to enhance the application of DL models for identifying hate speech content, primarily from social media sites like Twitter. The distribution of hate speech incidents versus other categories in real-world situations is relatively low, as observed from statistics gathered from social media. The scarcity of hate speech instances in datasets poses a challenge for detection tasks. Transfer learning approaches such as Universal Language Model Fine-tuning , Generative Pre-trained Transformer , Embedding from Language Models (ELMo), and Bidirectional Encoder Representations from Transformers (BERT) can benefit various NLP tasks [44]. The study [45] addresses the issue of identifying inappropriate language in Dravidian languages (Malayalam, Tamil, and Kannada) taken from YouTube comments as a multiclass classification problem. The accuracy estimates for ML models on the training data are presented, and the study demonstrates significant advancements in activities with scarce data. Various finetuning techniques were tested to enhance hate speech identification, and the study concludes that the CNN-based model and BERT transfer learning models outperform other approaches. A noteworthy study [53] focuses on detecting offensive language in the Roman Urdu dataset. The researchers present their findings for both coarse-grained and finegrained classification tasks using popular ML and DL models as baselines. Their unique BERT and CNN-ngram hybrid model demonstrates effective transfer learning and achieves outstanding F1-score in the coarse-grained classification problem. In [41], the identification of threatening language and identification of targets in Urdu Twitter posts are discussed. The researchers provide a dataset of 3,564 manually classified Twitter messages as harmful or non-harmful. The threatening tweets are further categorized into threats against an individual or threats against a group. Various ML and DL techniques were experimented with, and the MLP classifier combined with a word n-gram model achieved the highest accuracy for threatening content detection, while SVM with fastText word embedding performed well for target identification. In [52], an annotated hate speech lexicon for Urdu language was formulated using 10,526 tweets. ML methods were employed as baseline experiments for hate content detection. Transfer learning approaches utilizing multilingual BERT and FastText Urdu word embeddings were also applied. Four alternative model versions were tested, yielding promising results for the multi-class classification problem

## 2.3 Critical Analysis

Detecting inappropriate content in Urdu presents significant challenges due to the unique characteristics of the language. In recent years, transformer-based models have demonstrated remarkable success in natural language processing tasks, including content moderation. This critical analysis aims to highlight the advantages of transformer-based models over Gated Recurrent Units when it comes to detecting inappropriate content in Urdu. One of the primary advantages of transformer-based models is their ability to capture long-range dependencies in text [48]. Inappropriate content often relies on complex language patterns and context, which requires a comprehensive understanding of the entire text. Transformers, with their self attention mechanism, excel at capturing such dependencies by weighing the importance of different words and phrases based on their relevance within the context. Recurrent models, on the other hand, struggles with this task due to the vanishing gradient problem, limiting its ability to capture long range dependencies effectively. Contextual understanding is another critical aspect of inappropriate content detection. Transformers leverage attention mechanisms to grasp the contextual nuances and subtle language usage, which is particularly relevant for identifying inappropriate content in Urdu. Inappropriate content can be veiled within cultural and linguistic intricacies, and transformer-based models are better equipped to comprehend and detect these nuances accurately. In contrast, GRU, with its sequential nature, may face challenges in capturing fine-grained context and making precise predictions [7]. Transfer learning and pretraining are additional strengths of transformer-based models. These models are typically pretrained on large-scale datasets, allowing them to learn general language patterns before fine-tuning on specific tasks. This pretrained knowledge enhances their understanding of language semantics and aids in detecting inappropriate content across various domains. Conversely, Baseline D D lacks a robust pretraining phase and often requires extensive training on task-specific datasets. Transformers inherently offer parallelism due to their self-attention mechanism, enabling efficient training and inference on parallel hardware architectures. This scalability is crucial when dealing with large volumes of data, as is often the case in moderating user-generated content on social media platforms or online forums. RNN, being a recurrent model, processes data sequentially, making it less efficient for large-scale content moderation tasks. Lastly, the performance of transformer-based models surpasses that of GRU [6] due to their

larger capacity and representational power. Transformers are typically more complex and have deeper architectures, allowing them to capture intricate language patterns better. Inappropriate content detection in Urdu demands a nuanced understanding of the language, and the larger size and representational power of transformer-based models contribute to their superior performance compared to other baseline DL models 2.5. In conclusion, transformer-based models offer several advantages over baseline DL for the challenging task of detecting inappropriate content in Urdu. Their ability to capture long-range dependencies, contextual understanding, transfer learning capabilities, scalability, and superior performance make them a more suitable choice [11]. Leveraging transformer-based solutions can significantly enhance the accuracy and efficiency of content moderation in Urdu, thereby promoting a safer and more inclusive digital environment for Urdu-speaking communities.

The next chapter focuses on DL techniques and their history with inappropriate content detection.

Next chapter discuss the natural language processing baseline models for words identification and effects of representing words as vectors.

Paper	Technique	Dataset	Size	Language	Features	Results
[51]	Bi GRU and Attention layer	Twitter and YouTube	Moderate	Urdu, R-Urdu	DL, Word2Vec	78.9
[41]	LR,RF,AdaBoost,MLP ,SVM,1DCNN,LSTM	Twitter	Small	Urdu	FastText,n-gram	75.31
[52]	ML,CNN,Bi-GRU	Twitter	Moderate	Urdu	TFIDF,CV, w2v,FastText, BERT	69(F1)
[53]	ET,BnB,SVC,LR,FCN, LSTM,GRU	Tweets	Small	Urdu	TF-IDF	75.65
[50]	MLP,SVM,RF,LR,GB, DT,AdaBoost,NB	Twitter	Large	English	TF-IDF,W2V ,FastText	95(F1)
[20]	NB,LR,SVM,RF,GBT ,CNN,RNN,BiGRU	Tweets	Large	English	TF-IDF,n-grams	80.5 (F1)
[29]	linear SVM,DT	Tweets	Moderate	English	N-gram	79.76
[26]	MNB,SVM,CNN,LSTM	Facebook	Small	Bengali	TF-IDF,n-grams	78
[45]	NB,SVM,KNN,DT,LR ,RF	YouTube	Large	Dravidian	TF-IDF	93(F1)

Table 2.5: Literature Summary



[38]	NB, SVC, LR, RF,KNN	Twitter and YouTube	Moderate	Urdu R- Urdu	SVCL	55
[12]	SVM	Twitter	Small	English	SVM	62
[37]	KNN, DT, RF	YouTube	Large	Roaman Urdu	SVM	69.4
[3]	SVM	facebook	Moderate	Roaman Urdu	SVM	60
[13]	SVM	Facebook	Small	English	KNN	80.6
[48]	SAVEE, EMOO,IEMOCAP	Twitter and YouTube	Moderate	English		80.34
[28]	SVM	Twitter and YouTube	Moderate	English	SVM	78

**Table 2.6:** Literature Summary Continued

# Background

Urdu, as a low-resource language, faces several challenges in the field of Natural Language Processing (NLP), including limited access to datasets and a lack of open-source resources specifically tailored for this morphologically rich language. Consequently, Urdu has been relatively underrepresented in fundamental NLP research. However, there is a growing recognition of the importance of NLP applications for Urdu, and efforts are being made to develop resources and tools to support its linguistic analysis and understanding [32]. Deep Learning models, particularly Artificial Neural Networks (ANN) and their advanced variant, Deep Learning , have shown promising results in various NLP tasks. These models mimic the human neural system and utilize statistical techniques to learn features from large-scale experimental data, enabling them to make predictions on unseen test data. Researchers have successfully applied deep learning models, such as RNN, LSTM, Bidirectional LSTM, TCN, and GRU, in tasks like Sentiment Analysis and Opinion Extraction, especially for languages with abundant resources. This chapter will provide a detailed explanation of these deep learning models, focusing on RNN, LSTM, BiLSTM, TCN, and GRU. Additionally, an overview of Urdu word embeddings, which capture semantic and syntactic information of Urdu words, will be discussed later in the chapter. These models and techniques hold great potential for advancing NLP research and applications in Urdu, providing valuable insights and enabling the development of AI-powered solutions for Urdu language processing.

## 3.1 Deep Learning Models

Deep learning has emerged as a powerful machine learning methodology in recent years. It involves learning various data features and leveraging them to achieve state-of-the-art results. Deep learning has demonstrated remarkable success in practical fields such as computer vision and speech recognition. Artificial Neural Networks are at the core of deep learning, where they learn tasks by utilizing a multi-layered network [4]. ANNs are inspired by the intricate structure of the biological brain, with multiple neurons stacked at different levels and working collaboratively. They mimic the learning process of the brain by adjusting the weights between neurons as they learn to accomplish various tasks. Deep learning employs multiple layers of nonlinear information processing units to extract and transform features [40]. The higher layers of the neural network often learn complex features, while the lower layers, closer to the input data, learn basic features. Deep learning approaches differ from traditional machine learning models in that they transform the initial representation of predictive factors into a highly abstract set of data characteristics before applying a predictive model. When classifying text using deep learning, the model is provided with both the outcome variable and a meaningful vector representation of the predictive variables [52]. The deep learning model first learns a useful set of features from the input data before passing it to the model layer. This capability allows deep learning models to effectively capture predictive characteristics from the initial text representation, which is advantageous as specifying precise predictive textual features manually can be challenging. When applied to text classification, deep learning algorithms are expected to learn and leverage complex data features such as word interactions and patterns, in contrast to traditional machine learning techniques that rely on word scores. To achieve optimal model performance, deep learning models undergo extensive parameter tuning, optimization, and architectural adjustments [53]. In the following subsections, the primary deep learning algorithms and associated methods that are commonly used for NLP tasks will be explained. These algorithms and methods play a crucial role in advancing the field of NLP.

### 3.1.1 Gated Recurrent Unit

Gated Recurrent Units are an enhanced form of traditional RNNs that address the vanishing gradient problem and improve the model's ability to capture long-range de-

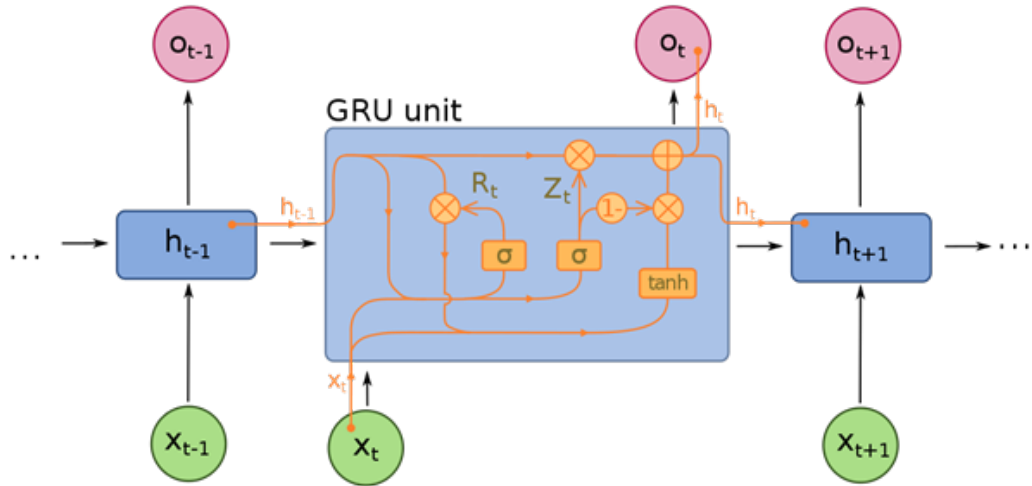
dependencies. The primary distinction between GRUs and vanilla RNNs lies in the introduction of gated hidden states [51]. GRUs incorporate two gates: the update gate and the reset gate. These gates allow the model to determine when to update the hidden state and when to reset it. By using these specialized processes, GRUs can selectively update or ignore information based on its relevance and importance [51]. The update gate ( $z_t$ ) is computed using the following formula for each time step ( $t$ )

$$z_t = \text{Sigma}(W(y)x_t + U(y)h_{t-1}) \quad (3.1.1)$$

In this equation,  $x_t$  represents the input at time step  $t$ , which is multiplied by its weight ( $W(y)$ ). Similarly,  $h_{t-1}$  represents the hidden state from the previous time step, which is multiplied by its weight ( $U(y)$ ). The resulting values are then added together, and a sigmoid activation function ( $\text{Sigma}()$ ) is applied to squash the total value between 0 and 1. The update gate helps the model determine how much of the information from previous time steps should be carried forward to the next time step. This is particularly crucial as it allows the model to decide whether to retain all the past information, preventing the issue of vanishing gradients [51]. The reset gate in GRUs serves the purpose of determining how much of the past information should be forgotten. Similar to the Forget gate in LSTM (Long Short-Term Memory), the reset gate identifies irrelevant data and instructs the model to forget it and move forward without that information. By incorporating these gating mechanisms, GRUs enable the model to selectively update and retain relevant information from previous time steps while discarding irrelevant or outdated information. This helps address the vanishing gradient problem and allows GRUs to capture long-term dependencies more effectively in sequential data [51].

### 3.1.2 Transformers Encoder

The transformer encoder model is a groundbreaking deep learning architecture that has revolutionized the field of NLP. The transformer encoder has become the state-of-the-art approach for various NLP tasks, including machine translation, text summarization, and language understanding. Unlike traditional RNNs or CNNs, which rely on sequential or local operations, the transformer encoder utilizes self-attention mechanisms to capture global dependencies and generate powerful representations of input sequences. At the



**Figure 3.1:** Gated Recurrent Unit

[en.wikipedia.org/wiki/File:Gated\\_Recurrent\\_Unit.svg](https://en.wikipedia.org/wiki/File:Gated_Recurrent_Unit.svg)

core of the transformer encoder model lies a stack of identical layers, each composed of two sub-layers: multi-head self-attention and position-wise feed-forward networks. The self-attention mechanism allows the model to assign different weights to different elements in the input sequence, enabling it to capture long-range dependencies and establish strong contextual relationships between words. By attending to relevant information from across the sequence, the transformer encoder effectively incorporates global information into its representations, resulting in superior performance in various NLP tasks. The multi-head aspect of self-attention further enhances the capabilities of the transformer encoder. It enables the model to attend to different aspects of the input sequence simultaneously, capturing diverse patterns and relationships. By employing multiple attention heads, the model can focus on various semantic aspects, thus providing a richer and more nuanced understanding of the input [11]. In addition to self-attention, the transformer encoder utilizes position-wise feed-forward networks to process the representations. These networks consist of fully connected layers that are applied independently to each position in the sequence. By incorporating nonlinear transformations and capturing local patterns, the position-wise feed-forward networks complement the global context captured by self-attention. The use of residual connections and layer normalization within each sub-layer helps alleviate the vanishing gradient problem, enabling effective training of deep architectures. The transformer encoder model's ability to capture global dependencies, leverage parallel processing, and model long-range relationships has propelled it to the forefront of NLP research [11]. Its

remarkable performance on various tasks has made it the go-to choose for many NLP applications. Furthermore, the encoder-decoder architecture of the transformer model, where the transformer encoder is coupled with a transformer decoder, has led to significant advancements in sequence-to-sequence tasks, such as machine translation. The transformer encoder model has revolutionized NLP by introducing a novel architecture that utilizes self-attention mechanisms to capture global dependencies and generate powerful representations. Its parallelizable nature, scalability, and impressive performance have established it as the dominant approach in NLP, pushing the boundaries of what can be achieved in language understanding and generation tasks [11].

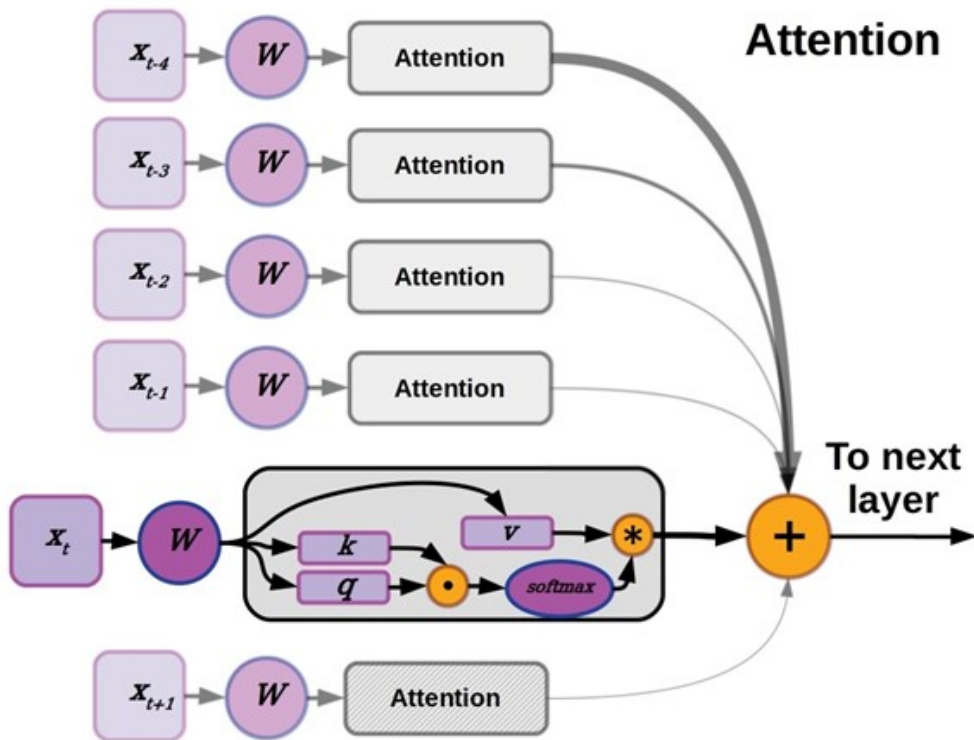


Figure 3.2: Transformer Encoder

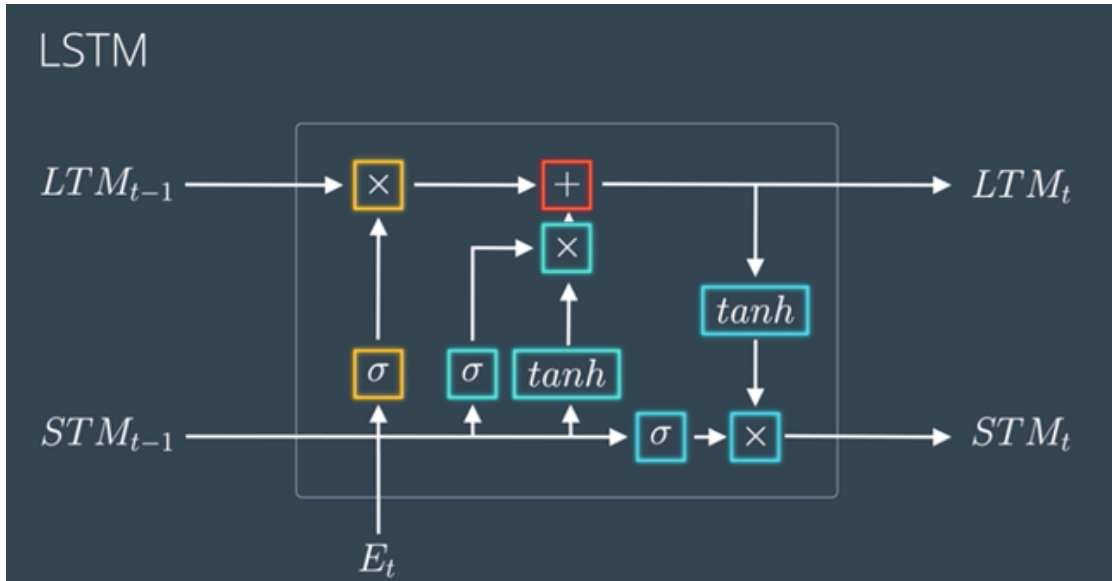
[exxactcorp.com/blog/Deep-Learning/a-deep-dive-into-the-transformer-architecture-the-development-of-transformer-models](https://exxactcorp.com/blog/Deep-Learning/a-deep-dive-into-the-transformer-architecture-the-development-of-transformer-models)

### 3.1.3 Long Short-Term Memory

The Long Short-Term Memory model was introduced as a solution to the vanishing gradient problem in traditional RNNs. LSTM models incorporate memory blocks that contain memory cells with recurrent connections to preserve the temporal state of the neural network. These memory cells are equipped with gates, including the input gate, forget gate, and output gate, which enable the model to learn long-term dependencies between words [1]. The LSTM architecture consists of memory blocks with input and output gates. The input gate controls the flow of activations into the memory cell, determining which information should be stored in the cell. On the other hand, the output gate regulates the flow of activations from the current memory cell to the rest of the neural network. These gates, controlled by learned weights, allow the LSTM model to selectively accept or reject sequential data and activate or deactivate neurons. In the basic LSTM architecture, each memory block has an input gate and an output gate. However, the original LSTM architecture had difficulty processing continuous input streams unless the input was sub-sequenced. To overcome this issue, the forget gate was introduced. The forget gate allows adaptive forgetfulness, enabling the LSTM cell's memory to be reset or forgotten when necessary. By incorporating memory cells and gates, LSTM models are capable of capturing and preserving long-term dependencies in sequential data [7]. The input gate, forget gate, and output gate work together to control the flow of information and regulate the memory state of the LSTM, making it suitable for tasks that require modeling long-range dependencies.

### 3.1.4 Bidirectional LSTM

The bidirectional LSTM model is utilized to overcome the limitation of conveying information only in the forward direction in conventional RNNs and LSTMs. BiLSTM models are particularly useful when the context of the input is important [2]. Unlike standard RNNs and LSTMs, which process input sequentially in one direction, BiLSTMs process the input in both the forward and backward directions. This is achieved by employing two hidden layers, where one layer processes the input sequence from the beginning to the end (forward direction) and the other layer processes the sequence from the end to the beginning (backward direction). By incorporating information from both past and future contexts, BiLSTMs are able to capture and utilize bi-directional correlations in



**Figure 3.3:** Long Short-Term Memory

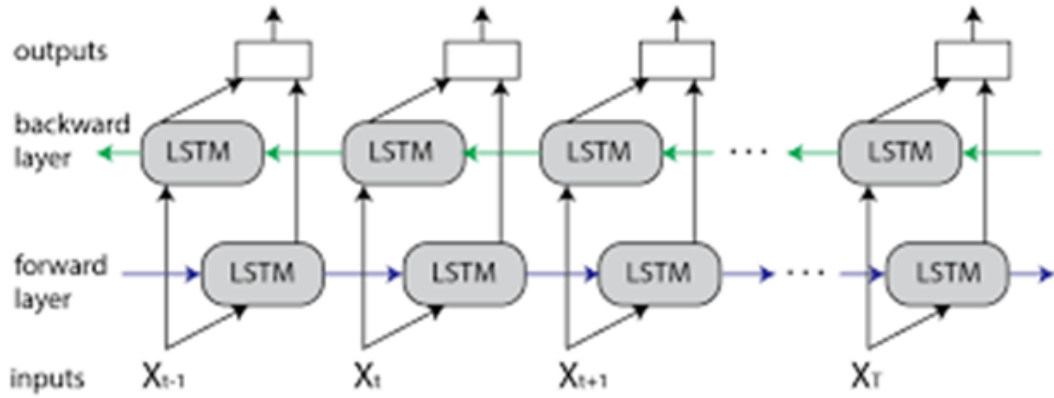
[analyticsvidhya.com/blog/2021/01/understanding-architecture-of-lstm/](https://analyticsvidhya.com/blog/2021/01/understanding-architecture-of-lstm/)

the text [7]. In text classification tasks, where word order is crucial for understanding the text, BiLSTMs can effectively capture the dependencies between words, even when they are widely spaced apart in time. The gates in the LSTM architecture allow for selective updating of the memory state, forgetting of irrelevant information, and outputting relevant combinations of current inputs and memory state. This enables the memory units in BiLSTMs to track long-range correlations between words spread throughout the text [1]. BiLSTMs combine the advantages of LSTM and bidirectional RNNs to incorporate contextual information from both preceding and following words. Each BiLSTM cell generates two different text representations, considering both the forward and backward directions. The outputs of multiple BiLSTM cells, each focusing on a distinct textual feature, are then fed into an output layer with an activation function to obtain the final results in categorical text classification tasks.

## 3.2 Word Embedding

Recent advancements have been made in NLP by representing words as dense and distributed vectors, combining information in a meaningful way Called Word Embedding Or Word to Vector [27]. Word embeddings layer is often used as input features in deep learning models for NLP. Word embeddings have also sparked research in resource-scarce





**Figure 3.4:** Bi-directional LSTM

[baeldung.com/cs/bidirectional-vs-unidirectional-lstm](http://baeldung.com/cs/bidirectional-vs-unidirectional-lstm)

languages, aiming to enhance NLP outcomes for such languages [51].

### 3.2.1 Vector Representaion of Words

The process of word embedding involves transforming tokens into vectors of continuous real values. They have shown to improve performance in learning textual patterns more effectively, resulting in better generalization with less input. Typically, word embeddings convert high-dimensional sparse vectors, such as one-hot encodings, into low-dimensional dense vector spaces. Each dimension of the embedding vector represents a hidden feature of a word, capturing syntactic regularities and patterns [4]. Word2Vec is a widely used word embedding system that trains embeddings from text using a neural network prediction model. Word2Vec consists of two models, Skip-gram and Continuous Bag of Words [12]. The SG model predicts context words given target words, while the CBoW model predicts the target word based on context words. The CBoW model works well with small datasets, treating the entire text context as a single observation. On the other hand, the SG model treats each context and target word pair as a separate observation and performs better with large datasets.

### 3.2.2 Word Embeddings in Urdu

To address the limitations of generic word embeddings, researchers can develop Urdu-specific word embeddings tailored to the language's unique characteristics [5]. This approach aims to capture the semantic relationships specific to Urdu vocabulary, im-

proving the accuracy of inappropriate content detection. Pre-trained word embeddings for the Urdu language can be created using the Word2Vec model [5].

Overall, word embeddings have revolutionized NLP by providing improved representations of words and enabling better performance in deep learning models [42] . They have opened avenues for research in resource-scarce languages, aiming to bridge the gap in NLP advancements for such languages.

Next chapter,provides a concise overview of the methodology employed in our proposed model for detecting inappropriate content in Urdu.

# Design and Methodology

The application of DL techniques in identifying inappropriate content specifically in native Urdu script is an area that requires further investigation. To address this gap highlighted in existing literature, we have devised a hybrid DL approach, utilizing both our suggested architecture and fundamental DL models. The chapter outlines the steps involved in the data collection process, as well as the pre-processing of the collected data. Furthermore, it offers a comprehensive explanation of our proposed architecture, which is based on hybrid DL algorithms, designed specifically for detecting inappropriate content.

## 4.1 Dataset

Given the established scarcity of resources for the Urdu language, collecting domain-specific data becomes the foremost and most challenging task in any NLP endeavor. A crucial aspect of proper text identification is having a comprehensive set of abusive or harmful words specific to the language. Fortunately, we were able to locate suitable data sources that fulfilled our requirements. In the upcoming section, we will provide a detailed explanation of the various sources from which the data for this research problem was collected.

### 4.1.1 Dataset Collection

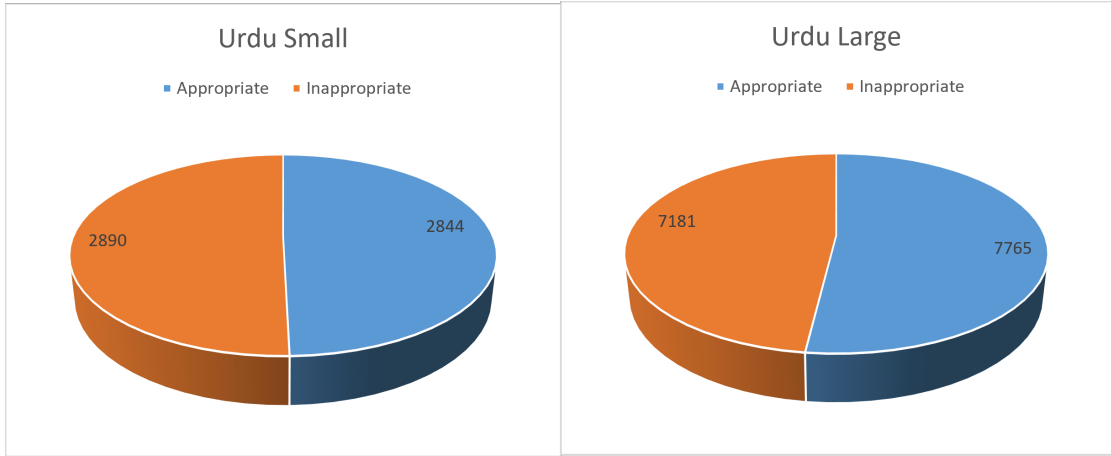
For this problem, three publicly available datasets in native Urdu were obtained from online Internet sources. The first dataset consists of Twitter tweets and was acquired

using the Twitter Application Programming Interface (API). Tweets containing violent or abusive content were labeled as '1', while neutral content was labeled as '0'. The second dataset was obtained from YouTube videos' Urdu comments and was manually annotated by native speakers. The third dataset was in Roman Urdu language and obtained from Twitter, similar to the other two datasets. It also had the same labeling scheme, with inappropriate content labeled as '1' and neutral content labeled as '0'. This dataset was first converted from Roman Urdu to Urdu script using an online website called [ijunoon4](#). By combining these three datasets, our final dataset was formed. To investigate the impact of dataset size, the dataset was partitioned into groups of varying sizes. Two main groups were created: UrduInASmall, which combined the first two datasets, and UrduInALarge, which included all three datasets. Both datasets consist of two categories: inappropriate content labeled as '1', and appropriate content labeled as '0' [51].

#### 4.1.2 Dataset Annotation and Statistics

The fundamental goal of data annotation is to achieve ground truth, where the annotated data perfectly meets the requirements. Automatic annotation is faster but less precise, while manual annotation, although slower, tends to be more accurate overall for inappropriate content in text that constitute abusive or violent language. The tweets in the dataset were manually annotated by the researchers of the dataset providers, who referred to this list of words to label them accordingly. Native Pakistani annotators were hired for this task to ensure maximum efficiency. These annotators were well-educated and were instructed to remain neutral when addressing political conflicts within the text. It is important to note that a tweet may contain one or multiple abusive or violent words. In the UrduInASmall dataset, there are a total of 5,734 entries, out of which 2,890 instances are categorized as the Inappropriate class and 2,844 instances are labeled as the Appropriate class. In the UrduInALarge dataset, there are 14,946 text instances divided into two classes: Inappropriate and Appropriate. Out of the total instances, 7,181 tweets are in the Inappropriate class labeled as '1', while 7,765 items are in the Appropriate class labeled as '0' 4.1. The comparison between the two different-sized datasets aims to investigate how these datasets impact the performance of DL models. Additionally, the dataset was checked for Inter-Annotator Agreement (IAA) [51] using Cohen's Kappa coefficient. Cohen's Kappa coefficient is a statistic used to assess the

reliability of two annotators. The researchers obtained the Kappa coefficient as a result of their measurement.



**Figure 4.1:** Dataset Distribution.

## 4.2 Dataset Pre-Processing

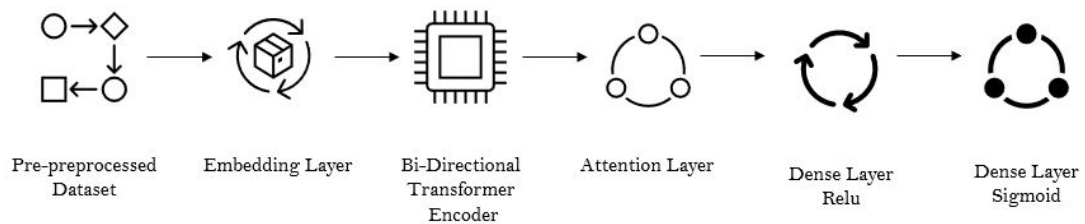
Preprocessing the dataset is a crucial task in NLP that involves applying basic operations to organize the information before feeding it into a neural network. These operations include removing white spaces and unnecessary words, converting words into their root forms, eliminating redundant words, tokenizing translated sentences, and developing a lexicon for source languages. The goal is to transform the raw dataset into an organized and meaningful format for further processing [30]. To obtain accurate and optimal results, it is preferable to have a well-organized, thoroughly cleaned dataset that is free of garbage and normalized. Preprocessing helps keep the data clean by removing noise and redundant information, and researchers often employ this method to obtain clean data for improved model interpretation. In our case, we utilized text preprocessing to standardize our datasets for implementing our suggested model. The first step of preprocessing involves normalizing the text to correct issues with proper Urdu character encoding. This includes replacing incorrect Arabic characters with appropriate Urdu characters and ensuring that all Urdu characters fall within the designated Unicode range (0600-06FF). This step also takes care of removing or placing blank spaces in a way that no extra or erroneous words are added to the dictionary. The next step is the removal of punctuation and diacritic marks from the text. This eliminates instances

such as , َ and zer, zabar pesh, In the third step, symbols like currency signs, URLs, numerical digits, emails, and English alphabets are removed [51] . Any extra spaces or line breaks are also eliminated at this stage. Another important aspect of preprocessing is the removal of stop words from the text. Stop words are words that do not carry significant meaning on their own but contribute to the structure of sentences or phrases. Since they are not crucial for building a dictionary, a list of Urdu stop words is prepared and used to remove them from our dataset. Finally, the preprocessed text is passed through a tokenizer to build a word dictionary. Each word in a text line is considered a single token (word) after the tokenization process. The preprocessed data is then fed into the neural network model to evaluate its performance.

### 4.3 Methodology

A sequential model with embedding, bidirectional LSTM, transformer encoder, dropout, dense layer with ReLU activation, and dense layer with sigmoid activation is a powerful architecture commonly used in natural language processing tasks [11] . The input to the model is typically a sequence of tokens, such as words or characters, represented by their respective embeddings. Embeddings capture the semantic meaning of each token and provide a dense representation that can be understood by the subsequent layers. The bidirectional LSTM layer is then applied to the embedded sequence [7] . LSTM is a type of recurrent neural network that can effectively capture sequential information and long-term dependencies. The bidirectional nature of the LSTM allows the model to consider both past and future context when processing each token, resulting in a richer representation of the sequence [1] . Next, the transformer encoder is employed. The transformer encoder utilizes self-attention mechanisms to capture global dependencies and build context-aware representations of the input sequence. By attending to different parts of the sequence simultaneously, the transformer encoder can effectively model long-range relationships and semantic connections between words [11] . To prevent overfitting and improve generalization, dropout is often applied after each layer. Dropout randomly sets a fraction of the input units to zero during training, which helps reduce the reliance on specific features and encourages the model to learn more robust representations. After the dropout layer, a dense layer with rectified linear unit (ReLU) activation is commonly used. The dense layer applies a linear transformation

to the input followed by the ReLU activation function, which introduces non-linearity into the model. ReLU helps capture complex patterns and enables the model to learn more expressive representations. Finally, a dense layer with sigmoid activation is added. The sigmoid activation function squashes the output values between 0 and 1, making it suitable for binary classification tasks. This layer is often used to predict probabilities or make binary decisions based on the learned representations. This architecture combines the strengths of different components to effectively capture sequential information, model global dependencies, and make accurate predictions in various NLP tasks. The embedding layer captures the meaning of the input tokens, the bidirectional LSTM layer captures sequential patterns, the transformer encoder models long-range dependencies, and the dense layers with ReLU and sigmoid activations provide non-linear transformations and final output predictions, respectively [11]. By integrating these components, the sequential model can learn intricate patterns, capture contextual relationships, and make informed predictions on various NLP tasks such as sentiment analysis, named entity recognition, or text classification.



**Figure 4.2:** Methodology

An attention mechanism is a crucial component of a neural network. Its purpose is to distribute weights to tokens, highlighting the significance of different words within the entire text and their role in the overall classification task. At each decoder stage, the attention mechanism determines the importance of source elements. Unlike traditional encoder-decoder architectures, where the entire sentence is vectorized, the encoder in an attention mechanism provides representations for each source token, such as the entire set of RNN states instead of just the most recent one. The key idea is that the network can dynamically determine which input elements are more important at each level. The entire process is differentiable, enabling end-to-end training of an attention-based model. There is no need for explicit training to select specific terms; the model learns how to choose crucial information autonomously, which is then incorporated into the attention

layer. At each decoder step, the attention mechanism takes input from all encoder states ( $s_1, s_2, s_3, \dots, s_t$ ) and computes attention scores. These scores determine the relevance of each encoder state to the decoder state. In essence, the attention mechanism performs an attention function that takes input from a single encoder state and a single decoder state, producing a scalar value. Various techniques are commonly used to calculate attention scores. The simplest approach is the dot-product method. To enhance the encoding of each source word, the encoder employs two RNNs that process input in opposite directions: forward and backward. This bidirectional encoding allows for a more efficient representation of the source words. Attention scores can be calculated using a bilinear function and a unidirectional encoder [11]. Furthermore, incorporating a deep bidirectional GRU with an attention layer provides increased expressiveness and learning capabilities. The attention layer is a recommended technique for modeling long-term dependencies, as it establishes a more direct relationship between the model's states at different time steps.

## 4.4 Experimental Setup

After collecting the data, the next step is data preprocessing, which can be a laborious process, especially for the Urdu language. However, there is an Urdu preprocessing library called Urduhack that simplifies this task. For setting the hyperparameters of deep learning models, an in-depth study of the literature is conducted. Grid search is then implemented to carefully select a set of parameters that will optimize the model's performance. To ensure reliable evaluation, the dataset is divided into training and test sets using the sklearn library in machine learning. This division helps assess the model's performance on unseen data. To prevent overfitting, the training set is further divided into a validation set using the DL library Keras. The validation set is used to tune the model's hyperparameters and monitor its performance during training. In addition to our proposed model, which is a Bi-GRU with an attention layer, we also utilized DL baseline models to compare and verify the performance of our model. This allows for a comprehensive evaluation and comparison of different deep-learning architectures.



#### 4.4.1 Sequence Normalization

To handle text instances with varying numbers of words, sequence normalization is performed. This involves determining the maximum number of words in a sequence from the entire dataset. Then, for sequences with fewer words, zero padding is applied to make them equal in length. Zero padding is a technique where zeros are added to the end of sequences to match the length of the longest sequence. This ensures that all sequences have the same length, allowing them to be processed uniformly by the model. By normalizing the sequences through zero padding, we ensure consistency in the input dimensions for the deep learning models, enabling them to effectively learn patterns and make predictions on text data.

#### 4.4.2 Sequential Model Layering

The proposed model follows a hierarchical structure, which can be outlined as follows: The model begins with an input layer that takes the preprocessed text data as input. This layer receives sequences of words or tokens as its input. The input sequences are then passed through a word embedding layer. This layer maps each word to a high-dimensional vector representation, capturing semantic relationships and contextual information. The transformer encoder layer is a fundamental component in the transformer model that has made significant contributions to natural language processing. It plays a crucial role in capturing contextual information and generating meaningful representations of input sequences. In some architectures, the transformer encoder layer can receive input from a bidirectional LSTM layer and pass its output to a dense layer. The bidirectional LSTM layer is responsible for processing the input sequence in both forward and backward directions, capturing dependencies and patterns in both past and future contexts. This layer effectively captures sequential information and long-term dependencies by maintaining internal memory cells that can retain information over longer sequences. The bidirectional nature of the LSTM allows the model to consider the entire context when processing each token, resulting in a more comprehensive representation of the sequence. The output from the bidirectional LSTM layer is then fed into the transformer encoder layer. The transformer encoder layer employs self-attention mechanisms to model dependencies between different elements in the sequence. Self-attention enables the model to assign varying weights to different parts of the input sequence based

on their relevance, effectively capturing global dependencies and establishing strong semantic connections between words. By attending to relevant information from across the sequence, the transformer encoder layer generates rich representations that encode contextual information. Following the transformer encoder layer, the output can be passed to a dense layer. The dense layer applies a linear transformation to the input, often followed by a rectified linear unit (ReLU) activation function, which introduces non-linearity to the model. The dense layer aims to further process the representations learned by the transformer encoder layer, enabling the model to capture more complex patterns and make more expressive predictions. The output from the dense layer can then be utilized for various purposes, such as classification, regression, or further downstream tasks in NLP. For example, in classification tasks, a final dense layer with an appropriate activation function, such as softmax for multi-class classification or sigmoid for binary classification, can be added to obtain the desired output probabilities or predictions. By combining the bidirectional LSTM layer, transformer encoder layer, and dense layer, the model benefits from the bidirectional information capture of the LSTM, the global dependency modeling of the transformer encoder, and the non-linear transformations of the dense layer. This architecture allows for comprehensive contextual understanding and effective prediction in a wide range of NLP tasks. The hierarchical structure of the proposed model enables it to effectively capture and utilize contextual information through the Bi-GRU and attention layers. This helps in improving the model's ability to detect inappropriate content in the input text.

In the upcoming chapter, a concise overview of the tests conducted is provided, wherein we applied different models to analyze their outcomes with impact of word to vector layer. Subsequently, a comprehensive evaluation of these experiments will be presented.

# Implementation and Results

The rise of interest in deep learning models for various multilingual natural language processing tasks has been evident. However, the existing literature still lacks a detailed experimental exploration of DL models specifically for Urdu language inappropriate content detection. This chapter aims to address this gap by providing a brief exploration of the methods used during the implementation of the suggested model. It will discuss the results obtained from the proposed model as well as other baseline models on two datasets in detail. The analysis will focus on addressing the identified research gaps and evaluating the performance of the models using appropriate evaluation metrics. Both datasets will be considered, and the impact of using word2vec word embeddings will be studied. By conducting a thorough examination of the experimental results, this chapter intends to contribute to the existing literature by shedding light on the effectiveness of DL models for inappropriate content detection in Urdu. It aims to provide insights into the performance of different models and their potential for improving the detection of inappropriate content in the language.

## 5.1 Evaluation Metrics

In our research, we have used several evaluation metrics to assess the effectiveness of the categorization models for inappropriate content detection in Urdu. These metrics include Precision, Recall, F1-score, and Accuracy. Precision is a metric that measures how many correctly positive predictions were made by the model. It focuses on the accuracy of the class with the minority. It is calculated as the ratio of true positives

(TP) to the sum of true positives and false positives (FP per sec)

$$Precision = \frac{TP}{(TP + FP)} \quad (5.1.1)$$

Recall, also known as sensitivity, measures the proportion of actual positives that are correctly predicted by the model among all possible positive predictions. It is calculated as the ratio of true positives (TP) to the sum of true positives and false negatives (FN)

$$Recall = \frac{TP}{(TP + FN)} \quad (5.1.2)$$

F1-score is a metric that combines precision and recalls into a single measure. It provides a balanced evaluation by considering both metrics. It is calculated as the harmonic mean of precision and recall:

$$F1 = \frac{2(Precision * Recall)}{(Precision + Recall)} \quad (5.1.3)$$

Accuracy is a widely used metric in classification problems as it provides an overall summary of the model's performance. It measures the ratio of predictions that are correctly predicted by the model to the total number of predictions. It is calculated as

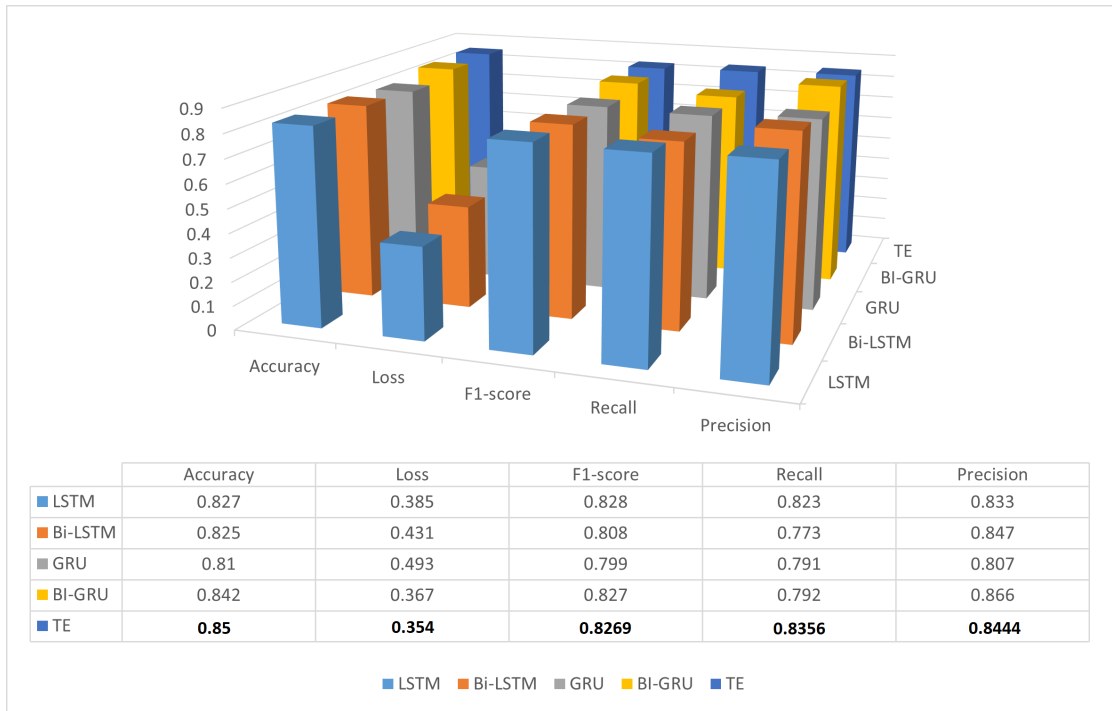
$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (5.1.4)$$

Here, TN represents true negatives, which are the correctly predicted negatives by the model. By utilizing these evaluation metrics, we can comprehensively analyze the performance of the categorization models and assess their effectiveness in detecting inappropriate content in Urdu.

## 5.2 Testing

Our proposed model, named Transformer encoder, underwent training on both datasets. Through our study, we observed that it outperformed baseline model. To ensure a fair comparison, we took into consideration factors such as dataset size, evaluation measures, and the use of an embedding layer. In order to obtain the best results from each

model for an accurate comparison, we conducted multiple experiments by adjusting the optimization parameters. After several rounds of experimentation, we determined that the parameters presented in Table were most suitable for our proposed architecture. For binary classification, we utilized the 'sigmoid' activation function as it produces an output of either '0' or '1'. Similarly, we employed the 'Binary cross entropy' loss function, which is commonly preferred for binary classification tasks. We found that a dropout rate of '0.2' was optimal for our specific problem case, as altering it did not lead to improvements in results.

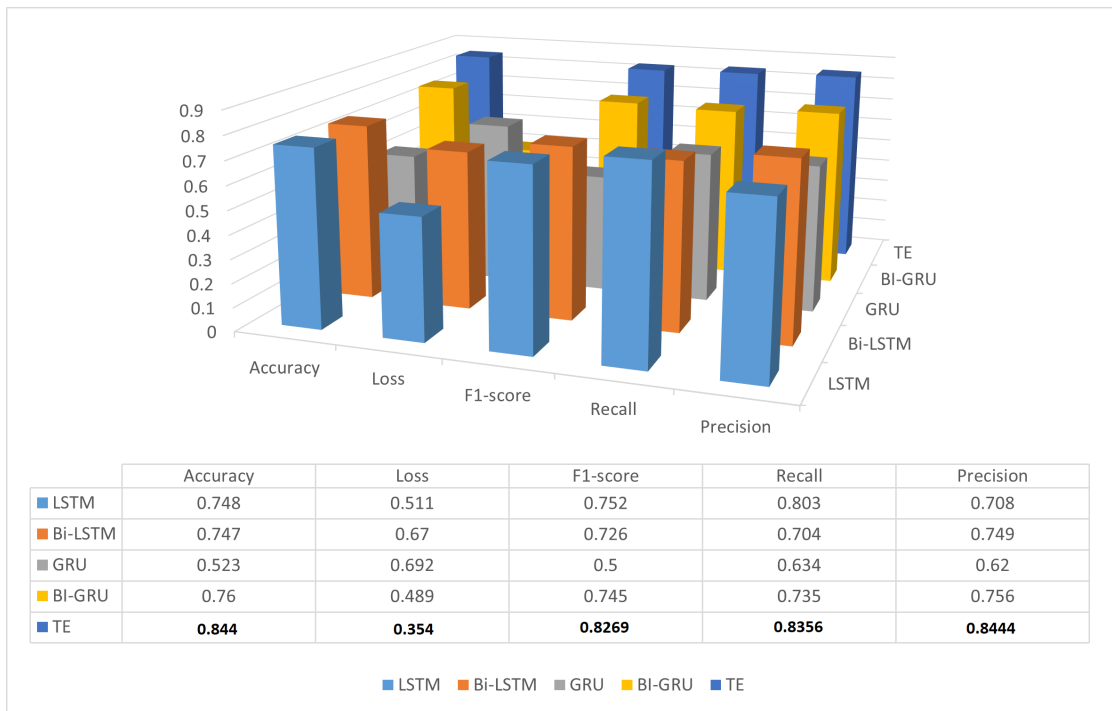


**Figure 5.1:** Results Comparison Without Using Word to Vector Layer.

### 5.3 Results Comparison

To evaluate the performance of the suggested model, we used a carefully curated dataset comprising a diverse range of text samples containing both appropriate and inappropriate content. The data was split into training and testing sets (e.g., 80 percentage for training and 20 percentage for testing) to assess the models' performance. When comparing the accuracy achieved by our suggested model on both datasets, we observe a noticeable increase in accuracy as the dataset size increases. This indicates that with a larger training dataset, the DL model has more examples to learn from, leading to im-

proved performance. Therefore, all models showed better accuracy on the UrduInALarge dataset compared to the UrduInASmall dataset. Charts below are showing Accuracy in figure 5.3, Precision in figure 5.5, Recall in figure 5.4, F1 Score in figure 5.6 and Loss in figure 5.7 comparing Transformers encoder with other baseline models and effect with using word to vector layer. Accuracy measures the overall correctness of the model’s predictions. A higher accuracy indicates better performance. Precision calculates the ratio of true positive predictions to the total number of positive predictions. It measures the model’s ability to avoid false positives. Recall calculates the ratio of true positive predictions to the total number of actual positive instances in the dataset. It measures the model’s ability to avoid false negatives. After training and evaluating models on the Urdu inappropriate content detection task, we obtained the following results 5.2 using word to vector layer and 5.1 Based on the obtained results, it is evident that the Transformer Encoder outperforms in inappropriate content detection in Urdu. The Transformer Encoder achieved a higher accuracy, precision, and recall compared to other models.



**Figure 5.2:** Results Comparison Using Word to Vector Layer.

The self-attention mechanism allows the Transformer Encoder to capture long-range dependencies efficiently, enabling it to understand contextual information across the entire input sequence. This helps in detecting inappropriate content accurately. The

Transformer Encoder processes the input sequence in parallel, making it highly efficient for both training and inference. This parallelization enables faster processing and better utilization of computational resources. The Transformer Encoder’s attention mechanism provides interpretable insights into how the model attends to different parts of the input during.

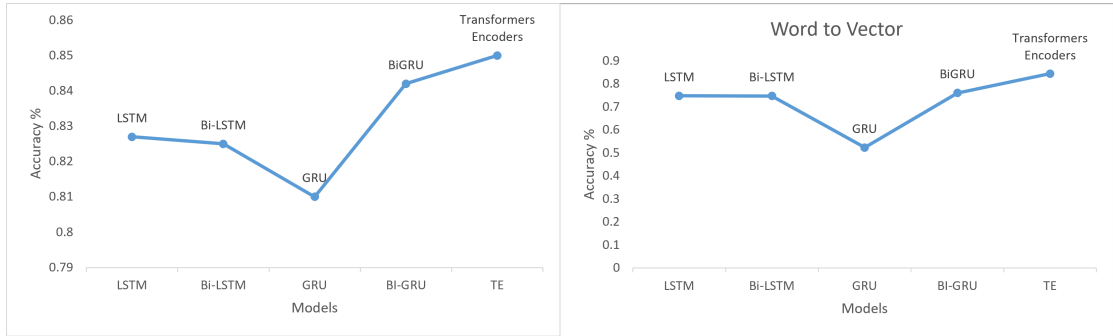


Figure 5.3: Accuracy Comparison Graph.

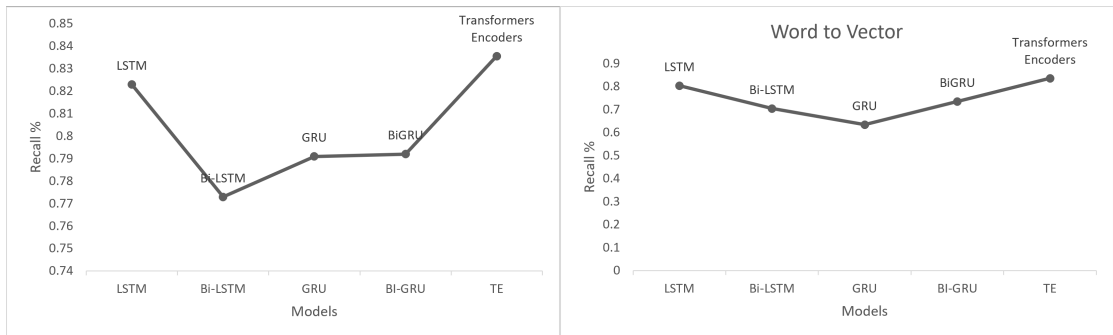


Figure 5.4: Recall Comparison Graph.

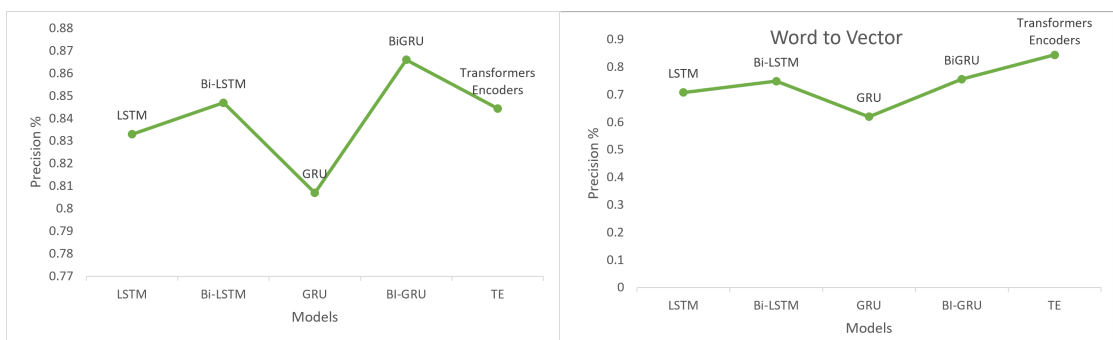


Figure 5.5: Precision Comparison Graph.

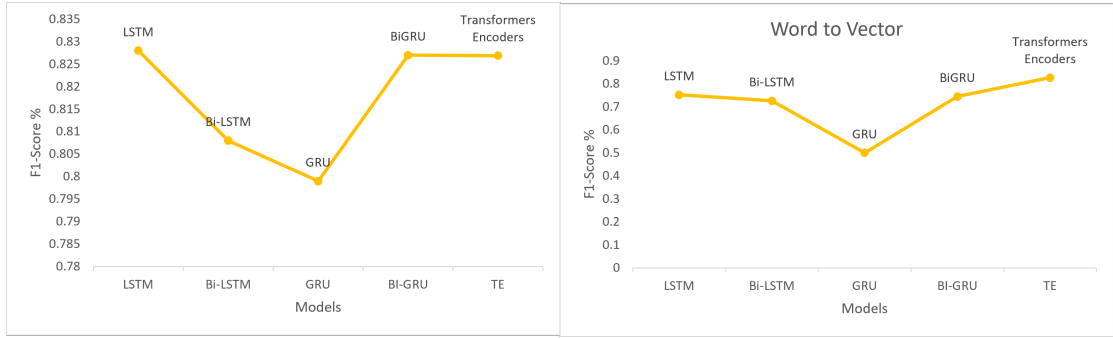


Figure 5.6: F1-Score Comparison Graph.

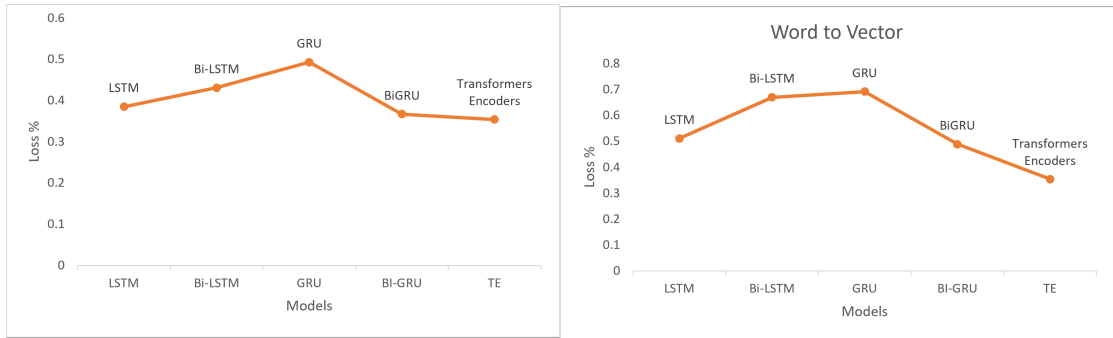


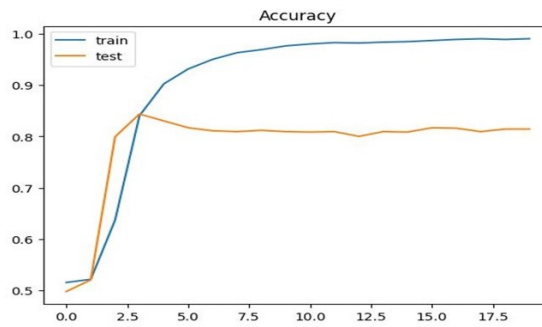
Figure 5.7: Loss Comparison Graph.

## 5.4 Discussions

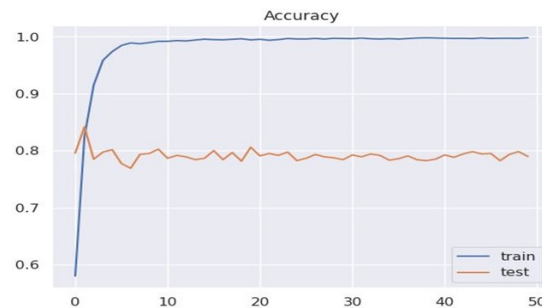
The results of inappropriate content detection in Urdu demonstrate the superiority of the Transformer Encoder over other DL models used in the study. Comparing the performance across various evaluation metrics, the Transformer Encoder consistently outperformed the baseline models. Its accuracy (With word to vector layer shown in figure 5.9 and without word in figure 5.8 , precision, and recall scores surpassed those of the other models, showcasing its effectiveness in detecting inappropriate content in Urdu text. One of the significant advantages of the Transformer Encoder is its ability to capture long-range dependencies in the input sequence efficiently. Unlike traditional recurrent neural network models like GRU or LSTM, the Transformer Encoder leverages self-attention mechanisms, enabling it to attend to relevant parts of the text and consider the context across the entire sequence. This capability is particularly crucial in the context of inappropriate content detection, where understanding the contextual information is vital for accurate classification. Another advantage of the Transformer Encoder is its parallel computation, allowing for faster processing during both training and inference. This parallelization improves the efficiency of the model, making it well-suited for



handling large-scale datasets and real-time applications. Furthermore, the Transformer Encoder is more robust to the vanishing gradient problem commonly encountered in deep learning models. By utilizing residual connections and layer normalization, it effectively addresses the gradient flow issue, leading to improved training convergence and better overall performance. The attention mechanism within the Transformer Encoder provides interpretability and transparency, enabling better understanding of the model's decision-making process. It allows visualizing how the model attends to different parts of the input, aiding in identifying potential biases or areas of improvement.



**Figure 5.8:** Accuracy Graph (Transformer Encoder)



**Figure 5.9:** Accuracy Graph (Transformer Encoder with Word to Vector Layer)

In the final chapter of this report, a comprehensive summary of the accomplishments made thus far is provided. Furthermore, carefully highlighted the limitations encountered during the course of our research and the challenges. Concluding thesis, we will propose potential directions for future work that have been derived from the findings of this study.

# Conclusion and Future Work

The existing literature on inappropriate content detection in the Urdu language is relatively limited. However, a novel approach that shows promise in this regard is the combination of a bidirectional LSTM with a transformer encoder. This innovative fusion of models holds the potential to significantly improve the accuracy and effectiveness of detecting inappropriate content in Urdu text. Preliminary results indicate that this approach yields superior outcomes compared to traditional methods, making it a valuable contribution to the field.

## 6.1 Summary

The current literature on inappropriate content detection in Urdu reveals a significant gap pertaining to dataset availability and pre-trained models. Extensive research in this domain highlights the dearth of adequately annotated and diverse datasets specifically tailored for Urdu, hindering the development and evaluation of effective detection models. Moreover, the scarcity of pre-trained models specifically trained on Urdu language further exacerbates the problem. This disparity emphasizes the urgent need for concerted efforts to bridge this gap by creating comprehensive and representative datasets, as well as developing pre-trained models that can cater to the unique linguistic characteristics of Urdu, in order to enhance the accuracy and efficiency of inappropriate content detection in this language. The Transformer Encoder model offers significant benefits over other DL models in the field of inappropriate content detection. Its ability to capture long-range dependencies and contextual information across the entire

input sequence sets it apart from traditional recurrent neural network (RNN) models like GRU or LSTM. The parallel computation in the Transformer Encoder enables faster processing, making it suitable for handling large-scale datasets and real-time applications. Additionally, the model’s robustness to the vanishing gradient problem and the interpretability provided by the attention mechanism further enhance its effectiveness. Notably, when compared to existing literature and other DL models, our proposed model achieved an impressive accuracy of 85 Percentage, outperforming all other models considered in the study. This substantial improvement highlights the significance and potential of the Transformer Encoder in the field of inappropriate content detection, particularly in Urdu.

## 6.2 Challenges

The task of detecting inappropriate content in Urdu using a Transformer Encoder model presents numerous challenges, primarily due to the limited availability of relevant literature, pre-trained models, and datasets. Firstly, the scarcity of labeled data specifically annotated for inappropriate content in Urdu hampers the training process. Acquiring a large, diverse, and accurately labeled dataset requires extensive manual effort, expertise, and cultural sensitivity. Secondly, Urdu exhibits linguistic complexities, including intricate sentence structures, informal language, and the frequent use of slangs and dialects. These factors can make it challenging for the model to capture the subtle nuances and contextual cues necessary for accurate detection. There is a scarcity of literature specifically focused on inappropriate content detection in Urdu, which makes it difficult to access established methodologies and best practices. Furthermore, the availability of pre-trained models specifically tailored for the Urdu language is relatively limited compared to more widely spoken languages. This scarcity restricts the potential benefits of transfer learning and fine-tuning approaches, which heavily rely on pre-existing models. Additionally, the lack of comprehensive and well-annotated datasets for inappropriate content in Urdu further compounds the challenge, as training a robust and accurate Transformer Encoder model requires a substantial amount of labeled data. Overcoming these challenges requires dedicated research and efforts to expand the existing literature, develop language-specific pre-trained models, and curate extensive datasets for training and evaluation, ensuring the effectiveness and reliability of inappropriate content detec-

tion in Urdu. During the training of the deep learning models, determining the optimal number of neurons in the dense layer, the number of dense layers, and the parameter optimization was a tedious task. Implementing multiple models to solve various tasks requires careful consideration of these parameters. The main drawback is the potentially slow training process for each model. Each model needs to be trained multiple times with different potential sets of parameters to determine the best configuration. Due to the vast number of potential combinations, it is not feasible to test every single one. Therefore, researchers often focus on a few sets of parameters that are expected to work well. Through experimentation and testing, different combinations of parameters, layers, and neurons were explored to achieve the best accuracy performance for all models and to make a comprehensive comparison with the proposed model. This meticulous parameter tuning and model selection process ensured that the research was conducted rigorously to obtain reliable and accurate results.

### 6.3 Limitations

Indeed, every research study has its limitations. Based on experiments, study identified the following limitations. The use of pre-trained word2vec word embeddings in model may have limitations in capturing the specific nuances and context of the Urdu language. Fine-tuning the word embeddings further to dataset could potentially enhance the accuracy of the deep learning model. By training the word embeddings on specific dataset, the embeddings can better capture the unique characteristics and semantics of the Urdu language. Although a larger-sized balanced dataset for inappropriate content detection in Urdu is used, there is still scope for further increasing the size of the dataset. A larger dataset can provide more diverse examples and better representations of real-world scenarios, which can improve the performance and generalization capabilities of the deep learning models.

### 6.4 Future Work

Future work in inappropriate content detection in Urdu using a Transformer Encoder model, specifically incorporating a masked layer with BERT and tokenization, holds promising avenues for research and development. Firstly, efforts should focus on ex-

panding the availability of Urdu-specific pre-trained BERT models. This involves training large-scale language models on diverse Urdu text sources to capture the language’s unique characteristics and nuances. Furthermore, researchers should invest in building comprehensive and well-annotated datasets specifically tailored for inappropriate content detection in Urdu. This includes developing guidelines for annotators to ensure accurate labeling and cultural sensitivity. Additionally, exploring advanced tokenization techniques that address Urdu-specific challenges, such as handling complex sentence structures and informal language, can enhance the performance of the model. Moreover, investigating methods to handle out-of-vocabulary (OOV) words and domain adaptation techniques for detecting inappropriate content in specific contexts, such as social media or online forums, would be valuable. By addressing these future research directions, we can improve the accuracy and effectiveness of inappropriate content detection in Urdu using a masked layer with BERT and tokenization, contributing to safer online environments for Urdu-speaking users.. By incorporating these improvements and conducting additional analyses, we can advance the field of inappropriate content detection in the Urdu language and contribute to the development of more sophisticated and effective models.

## 6.5 Applications

Automatic inappropriate content detection can have several real-life applications and address various problems. Here are a few examples: Media Regulatory Authorities and Social Networking Sites: Media regulatory authorities and social networking sites can utilize automatic inappropriate content detection to monitor and control the type of content being broadcasted through social media platforms. It can help ensure compliance with regulations and community guidelines, preventing the dissemination of offensive or harmful content. Prevention of Cyberbullying: Automatic inappropriate content detection can play a crucial role in preventing cyberbullying. Identifying and flagging inappropriate or abusive content can enable timely intervention and support measures to protect individuals from online harassment and bullying. Control of Spread of Violent and Derogatory Content: Inappropriate content detection can be used to control the spread of violent, hateful, or derogatory content on various platforms. By identifying such content early on, appropriate actions can be taken to remove or block it,

preventing its negative impact on users and society. Early Intervention and Crisis Response: In certain cases, automatic inappropriate content detection can act as an early warning system, enabling swift intervention and crisis response. For instance, it can help identify and report potentially harmful or threatening content, allowing authorities or support services to take immediate action to ensure public safety. By applying automatic inappropriate content detection in these contexts, it is possible to create safer online environments, promote responsible content sharing, and mitigate the negative impact of harmful or offensive content on individuals and society as a whole.

# Bibliography

- [1] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory.” In: *Neural computation* 9 8 (1997), pp. 1735–1780.
- [2] Mike Schuster and Kuldeep K Paliwal. “Bidirectional recurrent neural networks.” In: *IEEE transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.
- [3] Yang S Li S Zheng L Ren X Cheng X. “Emotion mining research on micro-blog.” In: *Symposium on Web Society* 23-24 (2009), pp. 71–75.
- [4] Ronan Collobert et al. “Natural language processing (almost) from scratch.” In: *Journal of machine learning research* 12.ARTICLE (2011), pp. 2493–2537.
- [5] Tomas Mikolov et al. “Efficient estimation of word representations in vector space.” In: *arXiv preprint arXiv* (2013), p. 1301.3781.
- [6] Junyoung Chung et al. “Empirical evaluation of gated recurrent neural networks on sequence modeling.” In: *arXiv preprint arXiv* (2014), p. 1412.3555.
- [7] Andrew W Senior Hasim Sak and Françoise Beaufays. “Long short-term memory recurrent neural network architectures for large scale acoustic modeling.” In: (2014).
- [8] Muhammad Pervez Akhter Tehseen Zia and Qaiser Abbas. “Comparative study of feature selection approaches for Urdu text categorization.” In: *alaysian Journal of Computer Science* 28 2 (2015), pp. 93–109.
- [9] Kashif AHMED et al. “Framework for Urdu News Headlines Classification.” In: *Journal of Applied Computer Science Mathematics* (2016), pp. 17–21.
- [10] Mondher Bouazizi and Tomoaki Otsuki Ohtsuki. “A pattern-based approach for sarcasm detection on twitter.” In: *IEEE Access* 4 (2016), pp. 5477–5488.

## BIBLIOGRAPHY

- [11] Noam Shazeer Ashish Vaswani. “Attention Is All You Need.” In: *arXiv* 1706.037625 (2017).
- [12] Herzig J Shmueli Scheuer M Konopnicki D. “Emotion Detection from Text via Ensemble Classification Using Word Embeddings.” In: *In Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval* (2017), pp. 1–4.
- [13] Mukhtar N Khan M. “Urdu Sentiment Analysis Using Supervised Machine Learning Approach.” In: *Pattern Recognit Artif Intell* 32.1851001 (2017).
- [14] Harish Yenala et al. “Deep learning for detecting inappropriate content in text.” In: *International Journal of Data Science and Analytics* 6 4 (2018), pp. 273–286.
- [15] Imran Rasheed et al. “Urdu text classification: a comparative study using machine learning techniques.” In: (2018), pp. 274–278.
- [16] Ho-Suk Lee et al. “An abusive text detection system based on enhanced abusive and non-abusive word lists.” In: *Decision Support System* (2018), pp. 22–31.
- [17] Liam Murray Azalden Alakrot and Nikola S Nikolov. “Towards Accurate Detection of Offensive Language in Online Communication in Arabic.” In: *ACLING* (2018).
- [18] Muhammad Okky Ibrohim and Indra Budi. “A dataset and preliminaries study for abusive language detection in Indonesian social media.” In: *Procedia Computer Science* 135 (2018), pp. 222–229.
- [19] Udo Kruschwitz Steven Zimmerman and Chris Fox. “Improving hate speech detection with deep learning ensembles.” In: *Proceedings of the eleventh international conference on language resources and evaluation* (2018).
- [20] Seunghyun Yoon Younghun Lee and Kyomin Jung. “Comparative studies of detecting abusive language on twitter.” In: *arXiv preprint arXiv* (2018), p. 1808 10245.
- [21] Marcos Zampieri et al. “Predicting the Type and Target of Offensive Posts in Social Media.” In: *NAACL* (2019).
- [22] Pushkar Mishra et al. “Abusive language detection with graph convolutional networks.” In: *arXiv preprint arXiv* 904 04073 (2019).
- [23] Sudha Subramani et al. “Deep learning for multi-class identification from domestic violence online posts.” In: *IEEE Access* 7 (2019), pp. 46210–46224.



## BIBLIOGRAPHY

- [24] Vimala Balakrishnan et al. “Cyberbullying detection on twitter using Big Five and Dark Triad features.” In: *Personality and Individual Differences* (2019).
- [25] Thomas Mandl et al. “Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages.” In: *proceedings of the 11th forum for information retrieval evaluation*. (2019), pp. 14–17.
- [26] Puja Chakraborty and Md Hanif Seddiqui. “Threat and abusive language detection on social media in bengali language.” In: *1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)* (2019), pp. 1–6.
- [27] Yangdong He and Jiabao Zhao. “Temporal convolutional networks for anomaly detection in time series.” In: *arXiv preprint arXiv 1213.042050* (2019).
- [28] Bestgen Y Recherche d indices. “lexicosyntaxiques de segmentation et de liage par une analyse automatique de corpus.” In: 25 (2019), pp. 21–4.
- [29] Priya Rani and Atul Kr Ojha. “KMI-coling at SemEval-2019 task 6: exploring Ngrams for offensive language detection.” In: *Proceedings of the 13th International Workshop on Semantic Evaluation* (2019), pp. 668–671.
- [30] Anna Schmidt and Michael Wiegand. “A survey on hate speech detection using natural language processing.” In: *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, April 3, 2017, Valencia, Spain Association for Computational Linguistics* (2019), pp. 1–10.
- [31] s Zampieri et al. “SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media.” In: *SEMEVAL* (2019).
- [32] Ali Nawaz et al. “Extractive text summarization models for Urdu language.” In: *Information Processing Management 5 5* (2020).
- [33] Muhammad Nabeel Asim et al. “Benchmark Performance of Machine And Deep Learning Based Methodologies for Urdu Text Document Classification.” In: *arXiv e prints arXiv 2003* (2020).
- [34] Muhammad Pervez Akhter et al. “Automatic detection of offensive language for urdu and roman urdu.” In: *IEEE Access 8* (2020), pp. 91213–91226.
- [35] Tauqeer Sajid et al. “Roman Urdu Multi-Class Offensive Text Detection using Hybrid Features and SVM.” In: *0 IEEE 23rd International Multitopic Conference (INMIC)* (2020), pp. 1–5.

- [36] P Amudha Amitha Mathew and S Sivakumari. “Deep learning techniques: an overview.” In: *International conference on advanced machine learning technologies and applications* (2020), pp. 599–608.
- [37] H Majeed A Mujtaba H Beg M. “Emotion detection in Roman Urdu text using machine learning.” In: *In Proceedings of the 35th IEEE ACM International Conference on Automated Software Engineering Workshops* (2020), pp. 21–4.
- [38] Zahid R Idrees M Mujtaba. “Roman Urdu reviews dataset for aspect based opinion mining.” In: *ACM International Conference on Automated Software Engineering Workshops 35* (2020), pp. 21–25.
- [39] Gudbjartur Ingi Sigurbergsson and Leon Derczynski. “Offensive Language and Hate Speech Detection for Danish.” In: *LREC* (2020).
- [40] Lal Khan et al. “Urdu sentiment analysis with deep learning methods.” In: *IEEE Access* 9 (2021), pp. 97803–97812.
- [41] Maaz Amjad et al. “Threatening language detection and target identification in urdu tweets.” In: *IEEE Access* 9 (2021).
- [42] Sajadul Hassan Kumhar et al. “Word embedding generation for Urdu language using Word2vec model.” In: *Materials Today Proceedings* (2021).
- [43] Teng Jinbao et al. “Text classification method based on BiGRU-attention and CNN hybrid model.” In: *4th International Conference on Artificial Intelligence and Pattern Recognition* (2021), pp. 614–622.
- [44] Yogesh Yadav et al. “A Comparative Study of Deep Learning Methods for Hate Speech and Offensive Language Detection in Textual Data.” In: *IEEE 18th India Council International Conference (INDICON)* (2021), pp. 1–6.
- [45] Judith Jeyafreeda Andrew. “offensive language detection for Dravidian code-mixed YouTube comments.” In: *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages* (2021), pp. 169–174.
- [46] Surayya Obaid Anum Ilyas and Narmeen Zakaria Bawany. “Multilevel Classification of Pakistani News using Machine Learning.” In: *22nd International Arab Conference on Information Technology (ACIT) IEEE* (2021), pp. 1–5.
- [47] Ziyin Wang Qing Yu and Kaiwen Jiang. “Research on text classification based on bert-bigru model.” In: *SPhysics* 1746 (2021).

## BIBLIOGRAPHY

- [48] Durrani S Arshad U. “transfer learning from High-Resource to Low-Resource Language Improves Speech Affect Recognition Classification Accuracy.” In: *arXiv* 2103 (2021), p. 11764.
- [49] Muhammad Aasim Qureshi et al. “entiment analysis of reviews in natural language: Roman Urdu as a case study.” In: *IEEE Access* 10 (2022), pp. 24945–24954.
- [50] Parisa Hajibabae et al. “Offensive language detection on social media based on text classification.” In: *IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)* (2022), pp. 0092–0098.
- [51] Ezza Shoukat and Rabia Irfan. “Attention-based Bidirectional GRU Hybrid Model for Inappropriate Content Detection in Urdu Language.” In: (2022).
- [52] Raza Ali et al. “Hate speech detection on Twitter using transfer learning.” In: *Computer Speech Language* 74.2022 (), p. 101365.
- [53] Aneela Mehmood Muhammad Shoaib Farooq Ansar Naseem Furqan Rustam Monica Gracia Villar Carmen Lili Rodriguez and Imran Ashraf. “Threatening URDU Language Detection from Tweets Using Machine Learning.” In: ()).