Th_ASDS_23_40

TweetGuard: Combining Transformer and LSTM Architectures for Fake News Detection in Large-Scale Tweets

A thesis report submitted to Department of Statistics and Data Science, Jahangirnagar University, Savar, Bangladesh, in partial fulfillment of the requirement for the degree of “Master of Science” in Applied Statistics and Data Science

Kowshik Sankar Roy

ID: 20231040

June 2024

Department of Statistics and Data Science

Jahangirnagar University

Savar, Dhaka-1342, Bangladesh

TweetGuard: Combining Transformer and LSTM Architectures for Fake News Detection in Large-Scale Tweets

Author

Kowshik Sankar Roy
ID: 20231040
Department of Statistics and Data Science
Jahangirnagar University

Supervisor

Farhana Akter Bina
Assistant Professor
Department of Statistics and Data Science
Jahangirnagar University

External

Coordination Committee
Department of Statistics and Data Science
Jahangirnagar University

June 2024
Department of Statistics and Data Science
Jahangirnagar University
Savar, Dhaka- 1342, Bangladesh.

Statement of Originality

I, Kowshik Sankar Roy, declare that this dissertation titled "TweetGuard: Combining Transformer and LSTM Architectures for Fake News Detection in Large-Scale Tweets" and the work presented in it are my own and have been generated by me as the result of my own original research. This research work has yet to be previously submitted for a degree or diploma in any university. To the best of my knowledge, this dissertation does not contain any material previously published or written by another person except where due reference is made in the text. Any help received during this research has been acknowledged, and I have cited all sources of information used.

Kowshik Sankar Roy

June 2024

THIS THESIS IS DEDICATED TO MY MOTHER, WHOSE LOVE AND SUPPORT HAVE BEEN MY GREATEST INSPIRATION.

Acknowledgements

First and foremost, I want to express my gratitude to Almighty God, whose grace and mercy enabled me to complete my dissertation work successfully. I would like to convey my deepest gratitude to my supervisor, Farhana Akter Bina, for her invaluable guidance, encouragement, and unwavering support throughout my research. Her expertise and insights have been instrumental in shaping this dissertation. I am also grateful to the faculty members of the Department of Statistics and Data Science at Jahangirnagar University, whose knowledge and dedication have contributed significantly to the completion of my thesis work. Finally, I am profoundly grateful to my family for their unconditional love and support, especially my mother, whose sacrifices and belief in me have always been my greatest motivation.

This journey would not have been possible without each and every one of you. Thank you.

Author

Abstract

The proliferation of misinformation on platforms like Twitter, where rapid dissemination can significantly impact public discourse, underscores the urgent need for effective automated fake news detection systems. These systems are crucial in preventing the spread of falsehoods and maintaining informational integrity. Traditionally, one of the challenges in developing such systems has been the lack of comprehensive benchmark datasets, which are essential for reliably training and testing detection models. In response to this challenge, a robust model named "TweetGuard" has developed, leveraging the 'TruthSeeker' dataset, a recently published benchmark offering a rich collection of annotated tweets. This dataset provides a solid foundation for training and refining our detection techniques. The proposed model employs a novel classification architecture that integrates transformer and LSTM technologies, enhanced by advanced preprocessing steps, including BERTweet, for effective tokenization and contextual understanding. Compared to traditional classifiers, including various CNN, LSTM, Bi-LSTM, and Transformer configurations, the proposed model demonstrates superior performance. Its effectiveness and robustness are further validated through rigorous testing across three additional fake news datasets, confirming its reliability and adaptability in diverse informational settings. This evaluation not only highlights our model's superior ability to identify and classify misinformation accurately but also establishes a new benchmark for automated fake news detection on social media platforms.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

Chapter Outline

1.1 Overview

1.2 Motivation

1.3 Problem Statement

1.4 Objectives

1.5 Thesis Outline

1.1 Overview

In the age of digitalization, the rise of social media platforms has revolutionized the way information is disseminated and consumed, enabling users to share news, opinions, and updates in real-time. It's the medium of communication, information dissemination, networking, marketing & advertising, entertainment, education, social activism, and so on. According to the January 2024 global overview by Datareportal, social media usage continues to surge, representing an astounding 62.3% of the worldwide population now active on social platforms. The total number of users has reached 5.04 billion, marking a significant increase of 266 million new users within the past year (Kemp, 2024). Alongside the benefits of instant connectivity, social media has also become a breeding ground for the rapid spread of misinformation, commonly referred to as "fake news." Misinformation or Fake content poses a significant threat to public discourse, trust in institutions, and democratic processes, as false or misleading information can influence public opinion, sway elections, and even incite violence.

Misinformation, defined as incorrect or misleading information, is increasing online, facilitated by technological advancements that make it easier to manipulate photos and videos. Researchers at MIT have discovered that fake news spreads up to 10 times faster than accurate reporting on social media platforms. This phenomenon occurs because sensational and misleading posts often garner more attention and engagement than subsequent corrections. Algorithms on social media platforms further exacerbate the spread of misinformation by prioritizing content that generates high levels of interaction, thereby fueling networks of ongoing misinformation. These algorithms are designed to prioritize engagement rather than ensuring access to high-quality information, resulting in the rapid dissemination of sensationalized stories and opinions (Micich & Cross, 2023).

The term "fake news," although only officially added to the Oxford English Dictionary in 2019, has seen a significant increase in usage, with a 365% rise from 2016 to 2017 alone. A poll conducted in January 2020 across multiple countries revealed that only 38% of respondents trust news most of the time, indicating a decline in public trust in news sources. Moreover, more than half of the global sample expressed concerns about the accuracy of information on the internet, particularly regarding news (Flood, 2018; Newman, 2020). Statistics further illustrate the pervasive nature of misinformation. In the United States, 67% of individuals have encountered fake news on social media, with 10% knowingly sharing such content. This

widespread dissemination of misinformation is increasingly recognized as a significant societal issue, with Min-Seok Pang, an associate professor at Temple University's Fox School of Business, describing it as a "life-and-death" matter that erodes trust and respect within society (Orbanek, 2021). Min-Seok Pang's research sheds light on disseminating fake news, revealing that social media users who verify their identity and receive a verified badge often contribute to spreading misinformation. Additionally, fake news posts containing videos are more likely to be reported by users, indicating a more significant skepticism towards video content online (Wang et al., 2021). In 2023, the prevalence of misinformation across digital and traditional media formats has become a pressing concern. Surveys indicate that 66% of U.S. consumers perceive most social media news as biased, with bots contributing significantly to the spread of COVID-19 misinformation online. This pervasive dissemination of false information poses significant risks to public health and democracy. Journalists recognize misinformation as a severe threat to public discourse, with 94% viewing fabricated news as a significant problem in America. Despite concerns about potential constraints on press freedoms, trust in mainstream news media remains polarized, emphasizing the need for collaborative solutions to combat misinformation and preserve journalistic integrity. The influence of social media platforms in disseminating misinformation is substantial, with billions of users worldwide. Surveys indicate that a significant percentage of U.S. news consumers unknowingly share fake news or misinformation on social media, underscoring the urgent need for solutions to address this issue and restore trust in information sources (Fake News Statistics & Facts (2023) — Redline Digital, 2023).

X, formerly and colloquially known as Twitter, is a prominent social media platform with a user base exceeding 500 million, placing it among the world's largest social networks. Users can share text messages, images, and videos, historically referred to as "tweets." The platform boasts over 330 million monthly active users and more than 192 million daily active users, generating around 500 million tweets per day. Regarding news consumption, 23% of Americans use Twitter as a news source, with 12% regularly accessing news content on the platform, ranking it the fifth-most-popular social network for news consumption in the United States. It serves as a medium for spreading misinformation or fake news, particularly concerning sensitive topics such as the US election, political issues, COVID-19, the Russia-Ukraine war, and similar issues. The challenge of detecting fake news is particularly acute on platforms like Twitter, where the brevity of posts, the rapid pace of information dissemination, and the prevalence of user-generated content make it difficult to distinguish between factual

news and fabricated stories. Traditional approaches to fake news detection, such as manual fact-checking and rule-based algorithms, are often labor-intensive, time-consuming, and limited in scalability.

Despite ongoing efforts by researchers and practitioners to combat fake news, its detection remains a complex and evolving problem, particularly within the context of social media platforms like Twitter. In this study, we aim to address the pressing need for effective fake news detection on Twitter by leveraging advanced natural language processing (NLP) techniques and deep learning algorithms. By focusing on the unique challenges posed by social media platforms, such as the brevity of tweets, the presence of user-generated content, and the rapid dissemination of information, we aim to contribute to the growing body of literature on fake news detection and advance our understanding of how state-of-the-art NLP models can be applied to address real-world challenges in online misinformation detection. In order to address these challenges, researchers and practitioners have turned to advanced natural language processing (NLP) techniques and machine learning algorithms, particularly RNN. Some advanced RNN or CNN algorithms, such as Long-Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Convolutional Neural Network (CNN), have been used to perform text analysis. CNN is used for spatial information extraction, while RNNs are utilized for capturing long-term dependencies and temporal patterns. In most recent years, Transformer-based models, such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer), have emerged as powerful tools for language understanding and generation tasks. These models, which leverage self-attention mechanisms to capture long-range dependencies in text data, have achieved state-of-the-art performance on various NLP benchmarks, including language translation, sentiment analysis, and text classification.

1.2 Motivation

In today's digital age, social media platforms like Twitter have revolutionized how information is shared and consumed, with billions of users engaging in real-time communication. However, this rapid information exchange has also facilitated the widespread dissemination of misinformation, or "fake news," which can have severe consequences on public trust, societal stability, and democratic processes. The proliferation of fake news undermines the credibility of legitimate information sources and has the potential to influence public opinion, electoral outcomes, and even incite violence.

Despite considerable efforts by researchers and practitioners, fake news detection remains a formidable challenge, particularly on platforms characterized by brief, user-generated content that spreads at an unprecedented pace. Traditional methods, such as manual fact-checking and rule-based algorithms, are not only labor-intensive and time-consuming but also struggle to keep up with the sheer volume and speed of social media content.

Given these challenges, there is an urgent need for innovative and efficient approaches to identify and mitigate the impact of fake news. This research is motivated by the desire to develop a cutting-edge solution that leverages the latest advancements in natural language processing and deep learning to enhance the detection of fake news on Twitter. By combining transformer architecture with Bidirectional LSTM, the proposed model aims to improve the accuracy and robustness of fake news detection, ultimately contributing to a more informed and trustworthy digital information landscape. This work not only addresses a critical societal issue but also pushes the boundaries of current technological capabilities in the realm of misinformation detection.

1.3 Problem Statement

The proliferation of social media platforms has transformed information dissemination, resulting in an unprecedented surge in the generation and sharing of content. While these platforms offer numerous benefits, such as instant communication and widespread reach, they also facilitate the rapid spread of misinformation, commonly referred to as "fake news." This phenomenon poses a significant threat to public discourse, trust in institutions, and democratic processes, as false or misleading information can influence public opinion, sway elections, and incite violence. Detecting fake news on social media, particularly on platforms like Twitter,

presents unique challenges. The brevity of tweets, the rapid pace of information dissemination, and the prevalence of user-generated content make it difficult to distinguish between factual and fabricated stories. Traditional approaches to fake news detection, such as manual fact-checking and rule-based algorithms, are often labor-intensive, time-consuming, and limited in scalability.

Despite significant research efforts, there remains a critical need for robust and scalable solutions to detect fake news effectively. Current methods often struggle with the diverse and dynamic nature of social media content, and many models lack the adaptability required to handle the rapid evolution of misinformation tactics. Moreover, the absence of a comprehensive benchmark data set further complicates the task. To address these challenges, this research aims to develop an advanced fake news detection system leveraging the strengths of both transformer architectures and Long Short-Term Memory (LSTM) networks. By integrating these techniques, we aim to create a hybrid model capable of accurately detecting fake news in large-scale tweets, considering the nuanced and contextual nature of social media content. The proposed model will be evaluated against multiple datasets, including the newly introduced Truth Seeker dataset, to assess its robustness and generalizability across various content types and domains. This research seeks to contribute to the ongoing efforts in combating misinformation by providing a scalable, efficient, and effective solution for fake news detection on social media platforms.

1.4 Objectives

In this research, we propose a novel hybrid model of fake news detection system from tweet using transformer and Bidirectional LSTM in a concatenation mode. Here transformer block serves as the default backbone of the proposed hybrid in architecture. The overall contribution of this work has been established around four folds. These are stated below:

- **Novel Classification Model**: A novel classification model has been proposed in the context of fake news detection from tweets, fusing transformer architecture and LSTM to develop a hybrid model architecture for detecting misinformation.
- **Text Cleaning and Tokenization**: Conventional text cleaning methods have been employed to remove noise and standardize text, ensuring accurate and meaningful word representations, while BERTweet, a pre-trained language model, has been utilized for

tokenization to capture the contextual nuances of tweets effectively, enabling more accurate and robust fake news detection.

- **Robustness Assessment**: For the purpose of assessing the robustness of the model, three additional fake news datasets have been utilized for the task of misinformation detection. Each label detection scheme has been evaluated and analyzed using a set of performance metrics.
- **Performance Evaluation**: In order to verify the strength and superiority of the suggested model, performance metrics have been evaluated among a group of deep learning classifiers, as well as the state-of-the-arts models currently used in the field.

1.6 Thesis Outline

The structure of the remainder of this thesis is outlined as follows:

Chapter 1: Introduction

This chapter provides an overview of the research, including the motivation, problem statement, and objectives of the study. It sets the context for the need for effective fake news detection on social media platforms, particularly Twitter.

Chapter 2: Literature Reviews

This section features a comprehensive literature review in the field of fake news detection, highlighting key methodologies and advancements in natural language processing and machine learning techniques relevant to this research.

Chapter 3: Proposed Approach

In this chapter, the complete architecture of the proposed hybrid model combining transformer and Bidirectional LSTM is presented. A detailed description of the datasets used for training and testing the model is also provided.

Chapter 4: Experimental Setup

The experimental settings, including data preprocessing, model configuration, and training procedures, are discussed in this chapter to provide a clear understanding of the methodological framework.

Chapter 5: Evaluation

This chapter details the experimental evaluations conducted in this study, including the implementation of the proposed model and the metrics used to assess its performance.

Chapter 6: Results and Discussion

A comprehensive analysis of the experimental results is provided in this chapter, along with a discussion of the implications of the findings, comparing the proposed model's performance with other state-of-the-art methods.

Chapter 7: Conclusion

The final chapter summarizes the work, highlighting the key contributions and findings of the research. It also provides final thoughts and suggestions for future work in the field of fake news detection.

Chapter 2 Literature Review

Chapter Outline

2.1 Overview

2.2 Related Works

2.1 Overview

The Literature Review section of this thesis provides an in-depth examination of existing research on fake news detection, particularly focusing on social media platforms like Twitter. It begins by defining fake news and discussing its evolution, followed by a detailed analysis of various detection methods ranging from traditional manual techniques to advanced machine learning and deep learning approaches, including the use of natural language processing (NLP) and transformer-based models like BERT and GPT. This section critically assesses each method's effectiveness and limitations, highlighting the strengths and shortcomings in current practices. By reviewing both theoretical frameworks and applied studies, it identifies gaps in the existing literature and sets the groundwork for the research to explore new approaches in enhancing the accuracy and efficiency of fake news detection using the latest technological advancements.

2.2 Related Works

The detection of fake news, a crucial challenge in the domain of digital information, has garnered significant attention due to its profound impact on society, politics, and public opinion. This section reviews existing research and methodologies developed to identify and mitigate the spread of false information. Firstly, it examines the various definitions and classifications of fake news as proposed by scholars, providing a foundational understanding necessary for exploring detection techniques. Subsequently, the review focuses on the evolution of these detection methods, ranging from early manual fact-checking processes to advanced computational approaches leveraging machine learning and natural language processing. Through this exploration, the section highlights key advancements and discusses the comparative effectiveness of different strategies in various contexts.

The concept of fake news detection as a distinct field of study began to gain significant attention in the early 21st century, particularly around the mid-2010s. However, efforts to identify and combat misinformation are not entirely new and have roots in various historical contexts where propaganda and misinformation were prevalent. In terms of formalized approaches and the application of technology to detect fake news, this area really began to develop alongside the rise of social media platforms, which became prevalent in the late 2000s and early 2010s. The 2016 U.S. presidential election was a pivotal moment that thrust the issue of fake news into the global spotlight (Allcott & Gentzkow, 2017). This event underscored the potential for

misinformation to spread widely and rapidly, influencing public opinion and political outcomes on a large scale. The technological response to detect and mitigate fake news started to incorporate more sophisticated tools from fields such as artificial intelligence and machine learning shortly thereafter. Researchers and technologists began to systematically apply computational techniques to identify patterns and indicators of misinformation. This included the development and implementation of algorithms that could analyze vast amounts of data quickly, a necessity given the scale and speed of information dissemination on platforms like Facebook, Twitter, and others. Thus, while the roots of identifying false information go back much further, the focused academic and technological pursuit of fake news detection as we understand it today really began to emerge in the 2010s, with significant developments occurring over the past decade.

The rule-based approach to detecting fake news involves creating manually crafted rules to identify patterns and anomalies typically found in false information, such as sensational language or contradictions to verified facts (Conroy, Rubin, & Chen, 2015). Initially effective, this method relies heavily on expert input and can quickly become outdated as misinformation evolves. Consequently, rule-based systems have been largely supplanted by machine learning (ML) techniques, which learn from large datasets to recognize subtle patterns indicative of fake news. These ML models offer greater scalability and adaptability, automatically updating their understanding as new information becomes available, thus maintaining relevance in the face of evolving misinformation tactics (Shu, Sliva, Wang, Tang, & Liu, 2017).

Conventional machine learning (ML) approaches to fake news detection typically involve feature engineering followed by the application of algorithms such as logistic regression, support vector machines, or decision trees. These techniques require manual extraction of relevant features from the data, such as word frequency, style markers, or metadata, which are then used to train a model to classify news as fake or real (Zhou & Zafarani, 2018). While effective, conventional ML approaches can be limited by the quality and comprehensiveness of the manually selected features, which may not capture all nuances of deceptive content. In Baarir and Djeffal (2021), a system for fake news detection is proposed using machine learning techniques. Term frequency-inverse document frequency (TF-IDF) of bag of words and n-grams are used as the feature extraction technique, and Support Vector Machine (SVM) is employed as the classifier. Additionally, a dataset of fake and true news is proposed for training the system. Abdulrahman and Baykara (2020) proposed a classification study, where four traditional methods were applied to extract features from texts: term frequency-inverse

document frequency (TF-IDF), count vector, character level vector, and N-Gram level vector. Ten different machine learning and deep learning classifiers were employed to categorize the fake news dataset: Random Forest (RF), K-Nearest Neighbors (KNN), Linear Support Vector Machine (LSVM), Logistic Regression (LR), Naive Bayes (NB), Adaboost, XGBoost, Artificial Neural Network (ANN), Recurrent Neural Network with Long Short-Term Memory (RNN+LSTM), and Convolutional Neural Network with Long Short-Term Memory (CNN+LSTM). The results demonstrated that fake news with textual content can be effectively classified, with CNN+LSTM showing particularly strong performance. The study achieved an accuracy range of 81% to 100% across different classifiers. The limitation of traditional machine learning approaches has led to the adoption of deep learning techniques, which can automatically discover the representations needed for detection from raw data, bypassing the need for manual feature engineering. Deep learning models, particularly those using architectures like recurrent neural networks (RNNs) and convolutional neural networks (CNNs), leverage large volumes of data to learn complex patterns and dependencies that are highly indicative of fake news. The shift to deep learning has resulted in models that are not only more accurate but also better at generalizing across different datasets, thereby significantly enhancing the robustness and effectiveness of fake news detection systems (Ruchansky, Seo, & Liu, 2017). The study by Sastrawan et al. (2022) evaluates deep learning methods for fake news detection using CNN, Bi-LSTM, and Res-Net architectures combined with pre-trained word embeddings. The models were trained on four datasets enhanced by data augmentation through back-translation to address class imbalances. Results showed that Bi-LSTM outperformed the other models on all datasets due to its superior ability to analyze contextual information from sequences, crucial for identifying the complex language in fake news.

The transformer-based approach, exemplified by pre-trained models like BERT (Bidirectional Encoder Representations from Transformers), leverages attention mechanisms to capture contextual relationships between words, significantly enhancing fake news detection. Unlike RNNs and LSTMs that process data sequentially and struggle with long sequences, transformers handle all words simultaneously, improving both speed and contextual understanding. These models, pre-trained on vast datasets, can be efficiently fine-tuned with specific fake news data, providing robust detection capabilities while addressing the scalability and latency issues associated with older models (Vaswani et al., 2017; Devlin et al., 2018). The study analyzed emotion in ideological and political education by integrating a gated recurrent unit (GRU) with an attention mechanism. Leveraging BERT's strengths, a bidirectional GRU

with a long focusing attention mechanism was used to extract both specific and global information. This complementary approach improved the accuracy of emotion detection. The model's validity and adaptability were confirmed using several fine-grained, publicly available emotion datasets (Shen & Fan, 2022). Alghamdi et al. (2023) explore the performance of various machine learning techniques, including fine-tuning pre-trained models like BERT and COVID-Twitter-BERT (CT-BERT), for detecting COVID-19 related fake news. By evaluating the efficacy of additional neural network layers such as CNN and Bi-GRU on top of these models, the study finds that the combination of Bi-GRU with CT-BERT, especially with selective parameter adjustments, delivers exceptional results, achieving a state-of-the-art F1 score of 98%. Rahman et al. (2023) proposes a Textual Similarity Analysis (TSA) method that leverages pre-trained models like GloVe and BERT, along with transformer based Seq2Seq, to assess the authenticity of news content. Their results indicate that these pre-trained models significantly outperform traditional encoding methods, achieving 98% accuracy compared to 77%-93%. Furthermore, the study evaluates various deep learning techniques, finding that transformers with 8 and 16 multi-heads outperform LSTM and GRU models, with accuracies of 98% and 97% respectively. This research underscores the effectiveness of advanced encoding and transformer architectures in TSA-based fake news detection, providing a robust foundation for future studies in this area.

The field of fake news detection is increasingly attracting attention, yet it faces significant challenges, primarily due to the scarcity of high-quality resources. This includes limited availability of comprehensive datasets and a dearth of published literature, which are crucial for developing and testing detection methods (Kumar & Arora, 2021). These constraints hinder progress by complicating the training and validation of algorithms designed to identify and counteract fake news effectively. This research addresses the challenge of automatically detecting fake content on social media platforms like Twitter, where manual fact-checking is impractical due to the volume of daily tweets. Dadkhah et al. (2023) addressed the challenge of automatically detecting fake content on social media platforms like Twitter, where manual fact-checking is impractical due to the volume of daily tweets. The research involved creating a comprehensive ground-truth dataset using a combination of Politifact, expert labeling, and crowdsourcing via Amazon Mechanical Turk, resulting in over 180,000 labeled tweets from 2009 to 2022. This dataset facilitated both five- and three-label classifications. Various machine learning and deep learning models, particularly those based on BERT, were applied to assess the accuracy of detecting real versus fake tweets. Additionally, the DBSCAN text clustering

algorithm and the YAKE keyword creation algorithm were used to analyze topics and their relationships. The research also included an analysis of Twitter users in the dataset, evaluating their bot score, credibility score, and influence score to identify any patterns related to the truthfulness of tweets. The findings demonstrate significant improvements in model performance for short-length texts in real-life classification tasks, such as detecting fake content on twitter.

Chapter 3 Proposed Approach

Chapter Outline

3.1 Overview

3.2 Dataset Description

3.3 Problem Statement

3.4 Pre-processing Stage

3.5 Proposed Model

3.1 Overview

In this research, we introduce a novel hybrid method for detecting fake news using a combination of transformer architecture and Bi-LSTM. To provide a comprehensive overview and insight into the entire workflow and architecture of our proposed approach, this section is subdivided into five consecutive sub-sections. An overview of the proposed approaches has been stated in section 3.1. Section 3.2 provides a summary of the proposed model and its overarching structure. Section 3.3 details the properties of the dataset utilized in the research. In Section 3.4, an in-depth description of the text cleaning pre-processing steps for the model is presented. A comprehensive analysis of the hybrid model developed for fake news detection is subsequently discussed in Section 3.5.

3.2 Proposed Architecture

As depicted in Fig. 3.1, our proposed model comprises two primary components: a text pre-processing unit and a hybrid transformer model for the classification stage. The initial segment of the pre-processing unit is pivotal for text analysis, involving tweet cleaning. Subsequently, in the second segment, the cleaned tweets are transformed into tokens, which are then converted into vectors. Following the completion of the pre-processing unit, the numeric representations of the texts are fed into the hybrid transformer model, which plays a central role in recognizing misinformation.

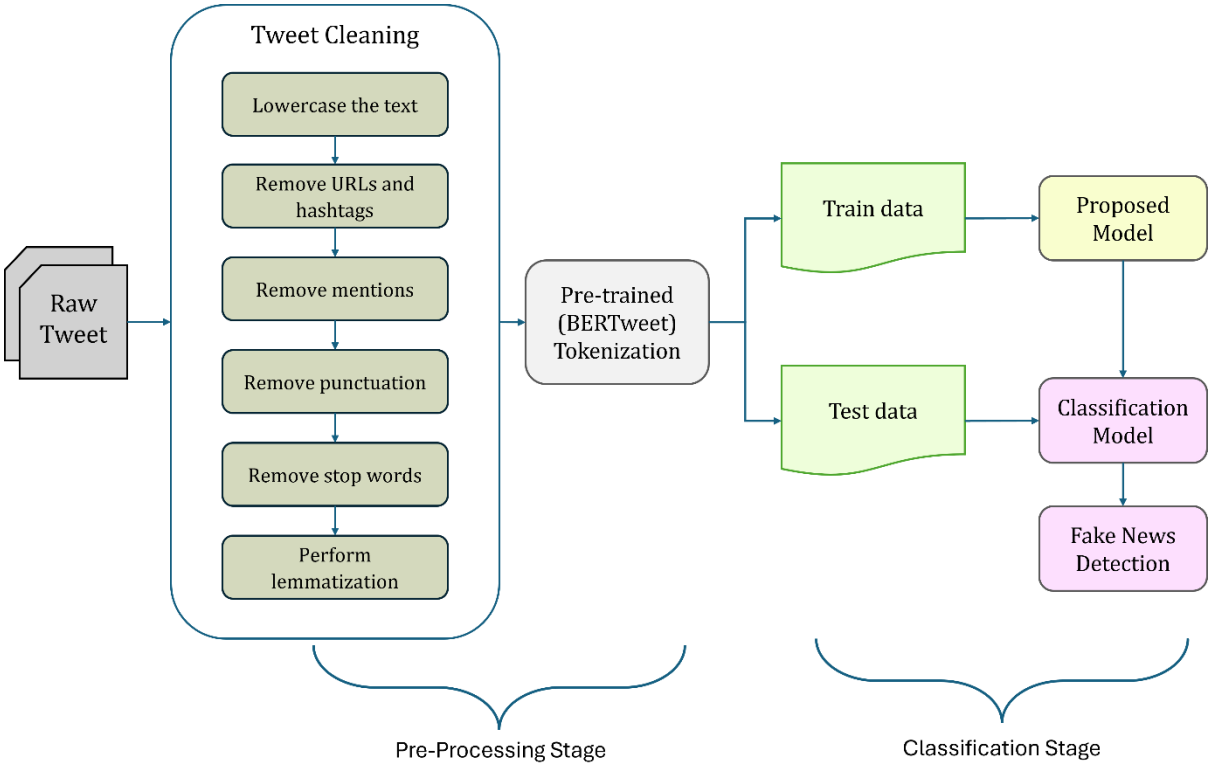


Fig. 3.1: Flow diagram of overall process of the proposed method

3.3 Dataset Description

In order to assess the effectiveness and reliability of any text analysis model, the availability of an appropriate dataset is crucial. Such a dataset should encompass a sufficient quantity of accurately labeled data reflecting real-world networks. Typically, researchers acquire data from social media platforms to conduct semantic analysis, yet the availability of benchmark datasets online remains scarce. The TruthSeeker dataset stands out as one of the most comprehensive benchmark datasets, comprising over 180,000 labeled Tweets spanning from 2009 to 2022.

The data for the TruthSeeker dataset was obtained through the crawling of tweets related to real and fake news sourced from the Politifact Dataset. By utilizing ground truth values and conducting targeted crawling for tweets associated with these topics (achieved by manually generating keywords linked to the news under scrutiny to be input into the Twitter API), over 186,000 tweets were extracted (prior to final processing). These tweets encompassed 700 instances each of real and fake news.

Subsequently, employing crowdsourcing via Amazon Mechanical Turk, a majority opinion regarding the degree of alignment between the tweet content and the authenticity of the news

source statement was generated. Following this, a majority agreement algorithm was applied to ascertain the validity of the associated tweets, resulting in classification into three and five category columns based on their alignment with the real or fake news source statements.

The main dataset directory, named "TruthSeeker2023" comprises two distinct .csv files:

1. Truth_Seeker_Model_Dataset: This file contains the features described in the preceding section on the TruthSeeker Dataset. It is tailored for utilization with Transformer model-based NLP models.
2. Features_For_Traditional_ML_Techniques: This file encompasses the 50+ features outlined in the Feature Dataset section. It is intended for use with classical machine learning techniques that require numerous features as input rather than generating features from data.

In this research, our focus was primarily on the first dataset, which is predominantly suitable for transformer-based models or large language models (LLMs). Below is the description of each feature of the 'Truth_Seeker_Model_Dataset', as presented in Table 3.1.

Table 3.1: Dataset Description

Feature List	Description
author	Represents the author of the statement.
statement	Denotes the headline of a news article.
target	Indicates the ground truth value of the statement.
BinaryNumTarget	Target is converted to binary where True encoded as 1 and False encoded as 0.
manual_keywords	Comprises manually created keywords utilized for searching Twitter.
tweet	Contains Twitter posts related to the associated manual keywords.
5_label_majority_answer	Presents the majority answer utilizing 5 labels: Agree, Mostly Agree, Disagree, Mostly Disagree, Unrelated.
3_label_majority_answer	Displays the majority answer utilizing 3 labels: Agree, Disagree, Unrelated.

3.4 Pre-processing stage

The pre-processing unit begins with cleaning the tweet, which is essential in NLP and LLM tasks for normalizing text, reducing noise, removing irrelevant information, and standardizing word representations, thereby leading to more accurate analysis and modeling results. This procedure conducts a sequence of essential text pre-processing steps to ready tweet data for analysis and modeling. Initially, it converts the tweet text to lowercase to ensure uniformity in representation. Then, it removes URLs and hashtags to eliminate extraneous information. Additionally, mentions are replaced with a generic "@user" tag to anonymize user identities and maintain privacy. Optionally, emojis are removed to further streamline the text. Punctuation is stripped to focus on the core content, while extra spaces are eliminated to enhance readability. Stop words, such as common words like "the" or "and" can be optionally removed to reduce noise in the data. Finally, lemmatization reduces words to their base form for consistency and simplifies subsequent analysis. These procedures collectively ensure that the tweet data is standardized, cleaned, and optimized for various NLP tasks, facilitating more accurate and effective analysis and modeling processes. Table 3.2 is a step-by-step example to illustrate the tweet cleaning stage.

Table 3.2: Text cleaning steps for tweets

Step	Operation	Tweet	Processed Tweet
1	Lowercase the text	"President @official announced new COVID-19 restrictions! Visit https://govupdates.com for details. #COVID19 #StaySafe 😊"	"president @official announced new covid-19 restrictions! visit https://govupdates.com for details. #covid19 #staysafe 😊"
2	Remove URLs and hashtags	"president @official announced new covid-19 restrictions! visit https://govupdates.com for details. #covid19 #staysafe 😊"	"president @official announced new covid-19 restrictions! visit for details. 😊"
3	Remove mentions	"president @official announced new covid-19	"president @user announced new covid-19 restrictions! visit for details. 😊"

		restrictions! visit for details. 😊"	
4	Remove emojis	"president @user announced new covid-19 restrictions! visit for details. 😊"	"president @user announced new covid-19 restrictions! visit for details. "
5	Remove punctuation	"president @user announced new covid-19 restrictions! visit for details. "	"president @user announced new covid19 restrictions visit for details "
6	Remove extra spaces	"president @user announced new covid19 restrictions visit for details "	"president @user announced new covid19 restrictions visit for details"
7	Remove stop words	"president @user announced new covid19 restrictions visit for details"	"president @user announced new covid19 restrictions visit details"
8	Perform lemmatization	"president @user announced new covid19 restrictions visit details"	"president @user announce new covid19 restriction visit detail"

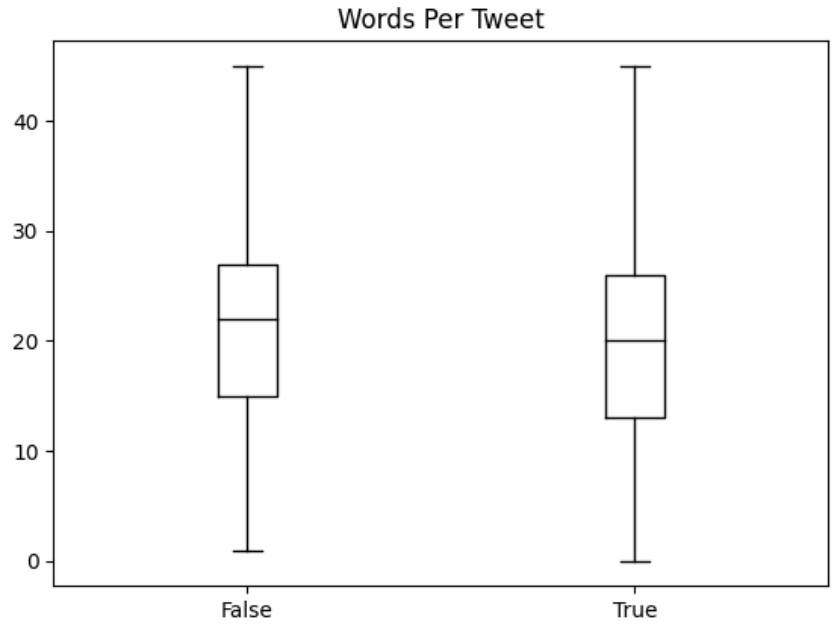


Fig. 3.2: Word Count Distribution in Genuine and Fake News Tweets

From Fig. 3.2, it is observed that for each class, most pre-processed tweets are around 20 words long, and the longest tweets are well below the maximum context size of the transformer model.

After tweet cleaning, tokenization becomes necessary to convert the text into a format suitable for natural language processing tasks. This step breaks down the text into individual tokens or words, enabling the model to understand the context and semantics of the text. The use of an auto tokenizer simplifies this process by automatically selecting the appropriate tokenization strategy based on the input data. Choosing BERTweet as the auto tokenizer is significant because it is specifically designed for Twitter data, capturing the nuances and informal language often found in tweets. BERTweet's pre-trained model, based on the BERT architecture, offers contextualized embeddings that capture the semantic meaning of words in the context of a tweet. This makes BERTweet a suitable choice for fake news detection tasks, where understanding the subtleties of language is crucial. In the data flow process, the cleaned tweet is passed through the auto tokenizer, which tokenizes the text and converts it into BERTweet-compatible input format. The output consists of tokenized representations of the tweet, ready to be fed into padding and sequencing mechanism. Pad sequencing is necessary to ensure that all input sequences have the same length, as neural networks require fixed-length inputs. This process involves adding padding tokens to shorter sequences and truncating longer sequences to a maximum length. In this context, the input consists of tokenized representations of tweets, while the output is a padded and sequenced format ready for further processing by the model. This step ensures consistency in the input data format, facilitating efficient training and inference.

The subsequent task in the data pre-processing unit involves creating a categorical label column, where a definitive truthfulness value is allocated. The criteria for label conversion are outlined in Table 3.3. The truthfulness value has been then converted into a label-encoded format to pass through the classifier.

Table 3.3: Conversion table for label representation

Target	Majority Answer	Truthfulness (Label)
True	Agree	True (0)
True	Disagree	False (1)
False	Agree	False (1)
False	Disagree	True (0)

The below Fig. 3.3 illustrates that, after label conversion, the class distribution is nearly balanced, with fake news posts comprising 2% fewer instances than true news posts.

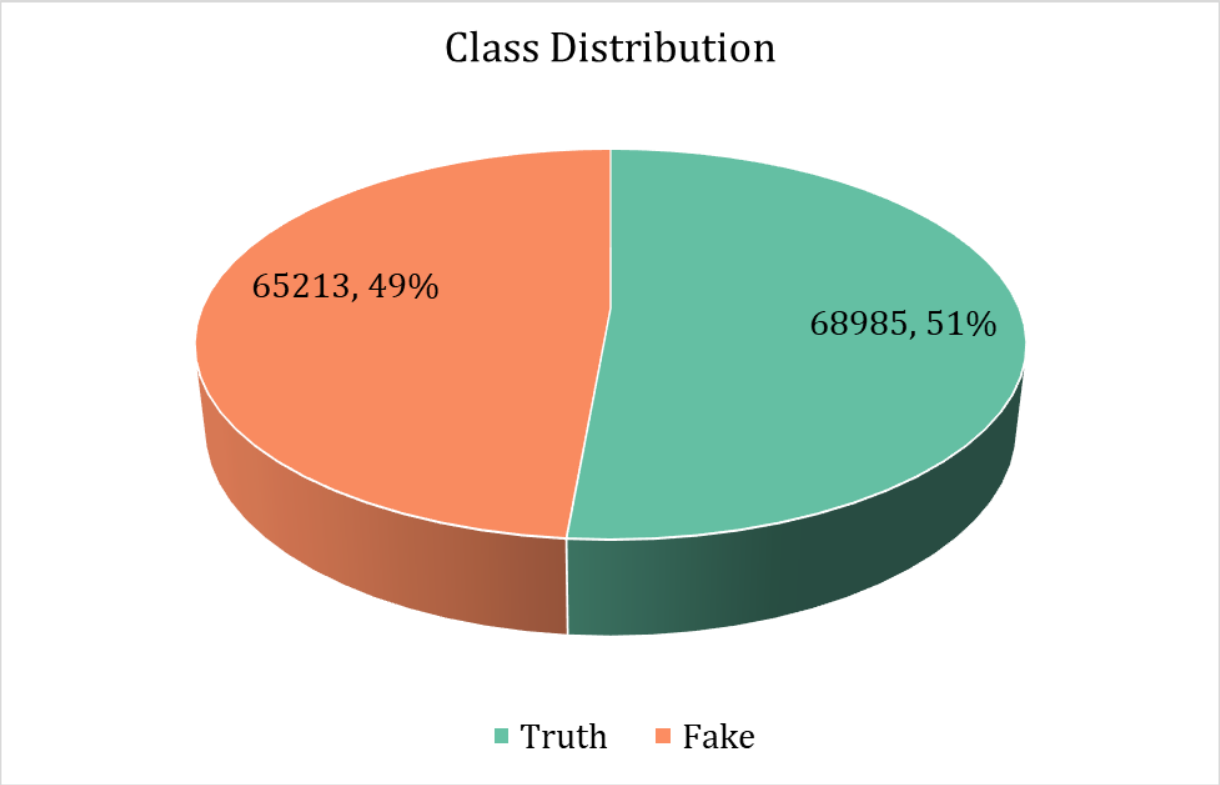


Fig. 3.3: Distribution of class labels for true and fake news within the dataset

After label conversion, the padded and sequenced tweets were divided into training and test data. This division was conducted using a random seed of 42, ensuring reproducibility. The dataset was split 80% to 20% for training and testing. The breakdown of the dataset after splitting is provided in Table 3.4 below.

Table 3.4: Breakdown of the dataset after splitting into train and test set

Tweet Category	Total	Train Set	Test Set
Fake News	65,213	52,154	13,059
Genuine News	68,985	55,204	13,781
Total News	134,198	107,358	26,840

3.5 Proposed Model

The proposed model for detecting fake news from tweets is a sophisticated hybrid neural network that integrates both recurrent and transformer-based architectures that have been visualized in Fig. 3.4. It begins with an embedding layer that transforms the input tweet text into dense vector representations. These embeddings are processed by a Bidirectional LSTM layer to capture long-range dependencies in the sequence, followed by a Dropout layer to mitigate overfitting. Additionally, a Transformer block, designed to focus on different parts of the input sequence through self-attention mechanisms, processes the embeddings in parallel. The outputs from the LSTM and Transformer block are concatenated, combining the strengths of both architectures. This concatenated representation is then passed through a Global Max Pooling layer to extract the most significant features, followed by another Dropout layer for regularization. The final Dense layer, with a sigmoid activation function, performs the binary classification to distinguish between fake and real news. The model is optimized with the Adam optimizer and employs L2 regularization to enhance generalization. The additional properties of the hybrid model have been demonstrated in Table 3.5.

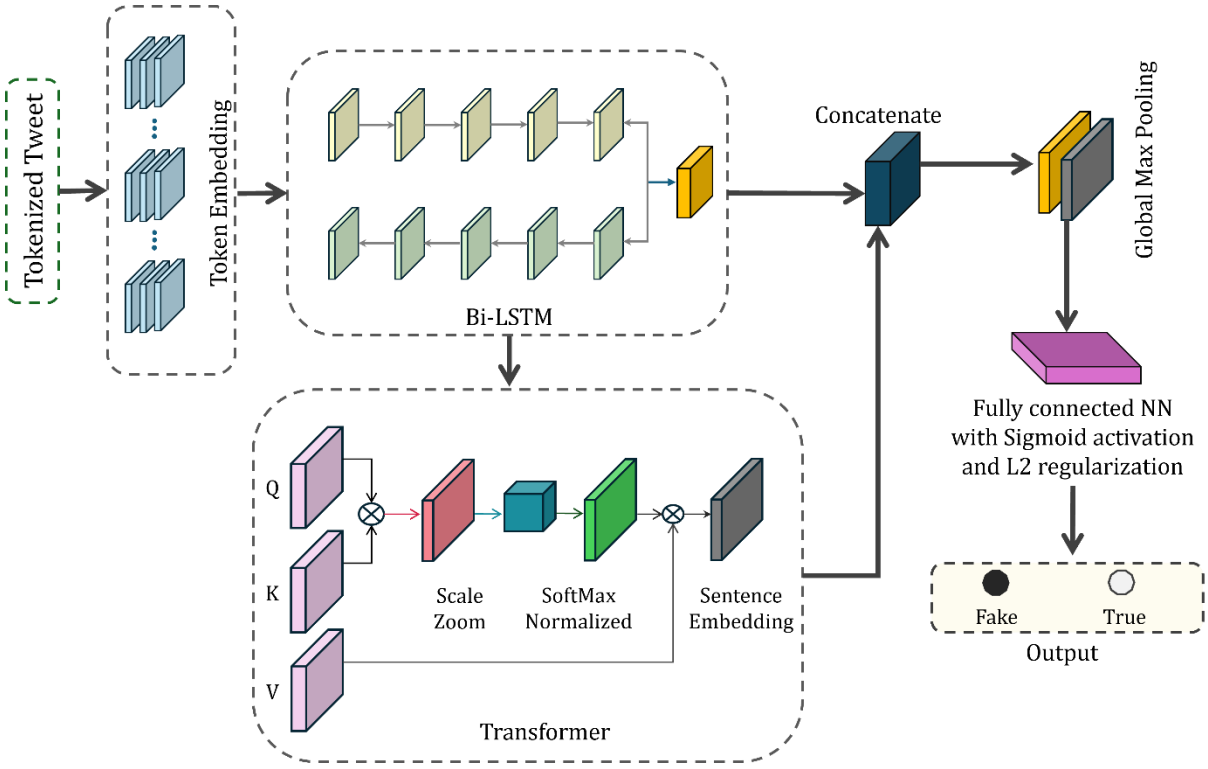


Fig. 3.4: Architecture of the proposed hybrid model

The use of concatenation in the model is significant as it integrates the strengths of both LSTM and Transformer architectures, creating a richer and more comprehensive feature set. The Bidirectional LSTM captures sequential dependencies and temporal patterns, while the Transformer block excels at capturing long-range dependencies and global context through self-attention mechanisms. By concatenating their outputs, the model leverages diverse representations, enhancing feature extraction, robustness, and generalization. This combination allows the model to access complementary insights, making it more flexible in learning and better equipped to handle complex patterns in tasks particularly fake news detection.

Table 3.5: Hyperparameters and their values for the hybrid proposed model.

Hyperparameters	Functions / Values
Embedding	Dimension = 512
Bi-LSTM	Activation = tanh, Neurons = 64
Transformer Block	Number of Heads = 8, Embedding Dimension = 512, Feed Forward dimension = 2048
Dense	Activation = Sigmoid, Neuron = 1
Regularization	L2, λ=0.01
Dropout	0.2
Batch Size	128
Learning Rate	0.01
Epochs	10
Optimizer	Adam
Cost Function	Binary Cross Entropy

3.5.1 Transformer Block

The Transformer block in the model is like a super attentive reader that carefully weighs the importance of different words in a tweet, helping the model understand which parts are most crucial for detecting fake news. It's like having a detective who can spot subtle clues and connections between words, even if they're far apart in the text. By doing this, the model can create a detailed map of the tweet's meaning, making it better at distinguishing between real and fake news. This layer works alongside other components like the LSTM to provide a

comprehensive understanding of the tweet's content, ultimately boosting the model's accuracy in identifying misinformation.

The transformer model architecture revolutionized the field of natural language processing (NLP). It's a neural network architecture based entirely on attention mechanisms without any recurrent or convolutional layers. The core operation of a transformer model is mainly maintained by two parts, these are encoder stack and decoder stack. Their operation is described below and visualized in Fig. 3.5.

A. Encoder Stack

The encoder stack consists of multiple identical layers, each containing two main sub-layers: a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. Each sub-layer has a residual connection around it, followed by layer normalization. This structure enables the Transformer to capture complex dependencies and contextual information from the input sequence. Here's a breakdown of the components in a typical Transformer encoder stack:

I. Input Embeddings: The input to the Transformer model is a sequence of tokens, typically represented as word embeddings and each token is represented by a d-dimensional vector, where d is the embedding dimension. Let X=$\{x_1, x_2, \dots, x_n\}$ be the input token sequence, where n is the sequence length, and each token is represented by a one-hot encoded vector. This vector has the same size as the vocabulary size V.

The mathematical expression for obtaining the embedding vector $Embedding\ (x_i)$ for token x_i from the input token sequence X using and embedding matrix E is:

$$Embedding\ (x_i) = E[Vocab(x_i)] \tag{3.1}$$

E is an embedding matrix of size $V * d_{model}$ where d_{model} is the dimension of the model typically same as embedding space, then $Embedding\ (x_i)$ is a vector of size d_{model} representing toke x_i in the continuous embedding space.

Here, $Vocab(x_i)$ denote the index of toke x_i in the vocabulary.

II. Positional Encoding: Since the Transformer doesn't have recurrence or convolution to maintain order information, positional encodings are added to the input embeddings to provide

information about the position of tokens in the sequence. The positional encoding is a vector added to the embedding of each token based on its position.

Let's denote the positional encoding function as $PE(pos, 2i)$ for the $2i$-th dimension and $PE(pos, 2i + 1)$ for the $(2i + 1) - th$ dimension, pos is the position and i is the dimension index. The positional encoding for position pos and dimension index i is computed as follows:

$$PE(pos, 2i) = \sin(\frac{pos}{10000^{2i/d_{model}}}) \tag{3.2}$$

$$PE(pos, 2i + 1) = \cos(\frac{pos}{10000^{2i/d_{model}}}) \tag{3.3}$$

Here, the factor $\frac{1}{10000^{2i/d_{model}}}$ ensures that each dimension of the positional encoding has a different frequency.

So, the final positional encoding vector for position pos is the concatenation of $PE(pos, 2i)$ and $PE(pos, 2i + 1)$ for all dimensions i:

$$PE(pos) = \begin{bmatrix} PE(pos, 0) \\ PE(pos, 1) \\ \vdots \\ PE(pos, d_{model} - 1) \end{bmatrix} \tag{3.4}$$

This encoding is then added elementwise to the token embeddings before feeding them into the model.

$$Final\ Embedding\ (x_i) = Embedding\ (x_i) + PE(x_i) \tag{3.5}$$

III. Multi-Head Self Attention Mechanism: The input of multi head attention sublayer of the first layer of the encoder stack is a vector that contains the embedding and the positional encoding of each word. This mechanism allows each word in the sequence to attend to all other words, capturing dependencies and relationships within the sequence. The self-attention mechanism computes attention scores between each pair of words and generates weighted sums for each word, based on these scores.

The input sequence X is projected into three different vectors: $Query\ (Q)$, $Key(K)$ and $Value\ (V)$ matrices. These matrices are obtained by multiplying the input sequence by learned weight matrices. Let $W_Q, W_K,$ and W_V denote the learned weight matrices for query, key, and value projections respectively. The projected sequences are denoted as,

$$Q = X.W_Q\ , K = X.W_K, V = X.W_V \tag{3.6}$$

- Query represents the focus or interest at a specific point in the sequence.
- Key acts like a memory or index in the sequence. It encodes information about other parts of the sequence that might be relevant to the current query.
- Value holds the actual information from each position in the sequence.

Each of the $Query\ (Q)$, $Key(K)$ and $Value\ (V)$ matrices are split into h heads (multiple heads), resulting in $Q_i, K_i, and\ V_i\ for\ i = 1,2, \ldots . h.$ This allows the model to attend to different parts of the input sequence independently.

Mathematically, the splitting is done along the last dimension (embedding dimension) to obtain h sets of query, key, and value matrices:

$$Q_i = Split(Q, d_{model}/h)\ , K_i = Split(K, d_{model}/h), V_i = Split(V, d_{model}/h) \tag{3.7}$$

For each head i, attention weights are computed as follows:

$$Attention_i(Q, K, V) = softmax\left(\frac{Q_i . K_i^T}{\sqrt{d_k}}\right) * V_i \tag{3.8}$$

Here, d_k is the dimensionality of the key vectors.

The outputs from all heads are concatenated and then projected back to the original embedding dimension using a linear transformation. All outputs form each head are then concatenated and multiplied by another learned weight matrix W_0 to obtain the final output of the multi-head self-attention mechanism:

$$MultiHead\ Output = Concatenate(Output_1, Output_2, \ldots, Output_h)\ .W_0 \tag{3.9}$$

For each position i in the sequence, the output of the multi-head self-attention mechanism is passed through a layer normalization operation. Generally, Layer normalization is applied to stabilize the training process. The operation is expressed as,

$$LayerNorm_1(MHAttention\ Output_i) = LayerNorm(MHOutput_i + Residual_1) \tag{3.10}$$

Where $Residual_1$ represents the residual connection from the input to the multi-head self-attention mechanism.

After the self-attention mechanism, each position applies a simple feed-forward neural network independently and identically. The FFNN consists of two linear transformations with ReLU activation function in between. The FFNN output is computed as,

$$FFNN(x) = ReLU(X.W_1 + b_1).W_2 + b_2 \quad (3.11)$$

Where $W_1, W_2, b_1\ and\ b_2$ are learnable parameters.

The output of the feed-forward neural network (FFNN) is passed through another layer normalization operation. The output of the $LayerNorm_2$ is expressed as,

$$LayerNorm_2(FFNN\ Output_i) = LayerNorm(FFNNOutput_i + Residual_2) \quad (3.12)$$

Where $Residual_2$ represents the residual connection from the output of the multi-head self-attention mechanism to the input of the FFNN.

After that. The output of the $LayerNorm_2$ is sent back to the next layer of the encoder stack and multi-head attention layer of the decoder stack.

B. Decoder Stack:

The Transformer decoder stack is responsible for generating the output sequence, leveraging both the encoder's output and previously generated tokens. It consists of several identical layers, each with three main sub-layers: masked multi-head self-attention, multi-head attention over the encoder's output, and a fully connected position-wise feed-forward neural network. Similar to the encoder, each sub-layer has a residual connection and is followed by layer normalization.

I. Input Embedding and Positional Encoding: Like the encoder, the input to the decoder is also a sequence of tokens. Each token is first embedded and then combined with positional encoding to capture its position in the sequence.

II. Masked Multi-Head Self-Attention: Unlike the encoder, the decoder's self-attention layer is masked to prevent attending to future positions. This is crucial during training, as the model is auto regressive, meaning it predicts one token at a time and should not have access to future tokens.

The self-attention mechanism in the decoder computes attention scores only for positions before the current position.

Mathematically, the masked self-attention output is computed similarly to the encoder, but with a mask applied to prevent attending to future positions. The operation expressed as,

$$SelfAttention_i = MultiHead(Q_i, K_i, V_i) \tag{3.13}$$

Where $Q_i = Query(Y_{i-1})$, $K_i = Key(Y_{i-1})$ and $V_i = Value(Y_{i-1})$

And Y_{i-1} is the output of the previous decoder layer.

III. Multi-Head Cross-Attention Mechanism: In the decoder, the cross-attention mechanism attends to the encoder's output. The process is similar to the self-attention mechanism, but queries come from the previous decoder layer, and keys and values come from the encoder output.

$$EncoderDecoderAttention_i = Attention(Q_i, K_{enc}, V_{enc}) \tag{3.14}$$

Where, $Q_i = Query(Y_{i-1})$, $K_i = Key(Z)$ and $V_i = Value(Z)$

And Z is the output of the encoder stack.

IV. Feed-Forward Neural Network and Residual Connections: Like encoder, the decoder has a feed-forward neural network after the attention layers, followed by residual connections and layer normalization.

$$FFNN_i = FFNN(Y_{i-1}) \tag{3.15}$$

$$LayerNorm_1 = LayerNorm(Y_{i-1} + SelfAttention_i) \tag{3.16}$$

$$LayerNorm_2 = LayerNorm(LayerNorm_1 + EncDecAttention_i) \tag{3.17}$$

$$Y_i = LayerNorm_2 + FFNN_i \tag{3.18}$$

Where Y_i is the output of $i - th$ decoder block.

Finally, the decoder's output is projected into a vocabulary-sized space using a linear transformation followed by a SoftMax activation, producing a probability distribution over the vocabulary for the next token. Mathematically, the output probability distribution P is computed as:

$$P = softmax(Y_N) \tag{3.19}$$

Where the final output of the decoder stack is Y_N.

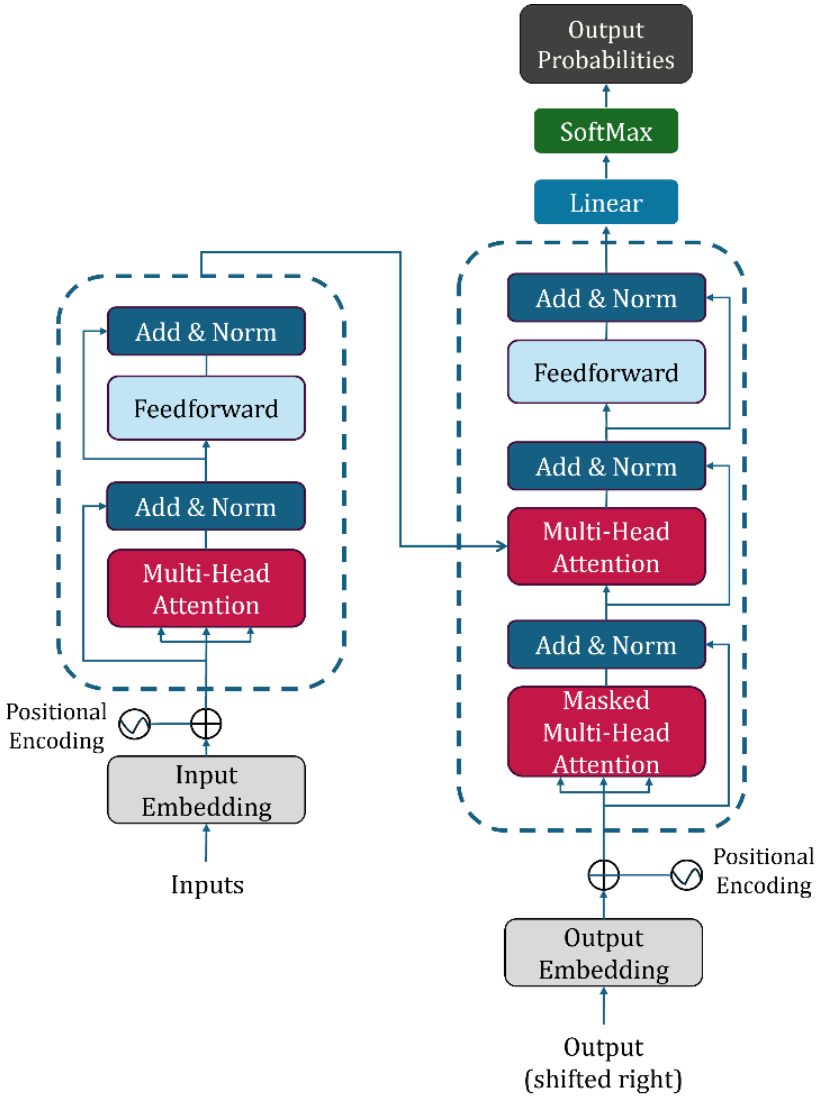


Fig. 3.5: The structural design of the Transformer

3.5.2 Bidirectional LSTM Cell

In Natural Language Processing (NLP), a Bidirectional Long Short-Term Memory (Bi-LSTM) network is a type of recurrent neural network (RNN) that processes the input sequence in both forward and backward directions, capturing contextual information from both past and future states for each position in the sequence. This makes it particularly effective for tasks where the context surrounding each word is crucial. The operational diagram of LSTM and Bi-LSTM cell has been presented in Fig. 3.6.

Here are the core equations governing an LSTM cell at time step t:

I. Forget Gate: The forget gate's role is to decide which information from the previous cell state should be discarded. It selectively forgets parts of the previous cell state based on the

current input and the previous hidden state. It uses a mechanism to analyze the input and previous hidden state to generate a value (between 0 and 1) for each piece of information in the cell state. A value close to 0 means the information will be largely forgotten, while a value close to 1 means it will be mostly retained.

$$f_t = \sigma(W_f \,.[h_{t-1}, x_t] + b_f) \tag{3.20}$$

Here, f_t is the forget gate's activation vector, W_f is the weight matrix for the forget gate, $b_f$is the bias,

σ is the sigmoid function,

h_{t-1} is the hidden state from the previous time step, and

x_t is the input at the current time step.

The output f_t is a vector of values between 0 and 1, indicating how much of each component of the cell state C_{t-1} (previous cell) should be forgotten.

II. Input Gate: The input gate determines which new information from the current input should be added to the cell state. It evaluates the current input and the previous hidden state to generate a value for each piece of the new information. Additionally, it creates a candidate for the new cell state, representing potential new information. The input gate uses these evaluations to update the cell state by adding new relevant information.

$$i_t = \sigma(W_i \,.[h_{t-1}, x_t] + b_i) \tag{3.21}$$

$$\widetilde{C_t} = \tanh\,(W_C \,.[h_{t-1}, x_t] + b_C) \tag{3.22}$$

i_t is the input gate activation, and $\widetilde{C_t}$ is the candidate cell state. W_i and b_i are the weight matrix and bias for the input gate, while W_C and b_C are for the candidate cell state. The candidate cell state $\widetilde{C_t}$ contains new information, which will be added to the cell state based on the input gate's decision.

III. Output Gate: The output gate controls what information from the cell state is passed to the hidden state, which in turn is used as output at the current time step and input to the next time step. It evaluates the current input and the previous hidden state to generate a value that determines which parts of the cell state will form the new hidden state. This hidden state represents the output for the current time step and is used in subsequent steps.

$$o_t = \sigma(W_o . [h_{t-1}, x_t] + b_o) \tag{3.23}$$

$$h_t = o_t . \tanh(C_t) \tag{3.24}$$

o_t is the output gate activation. W_o and b_o are weight matrix and bias for the output gate. The hidden state h_t is calculated by multiplying the output gate activation o_t with the tanh of the current cell state C_t. The cell state C_t is updated using the formula

$$C_t = f_t . C_{t-1} + i_t . \widetilde{C_t} \tag{3.25}$$

In a Bidirectional LSTM, two LSTM networks are used: one processes the sequence forward (from start to end), and the other processes it backward (from end to start). The final output at each time step t is a combination of both forward and backward LSTM outputs.

For a given input sequence $x = (x_1, x_2, \ldots, x_T)$:

- **Forward LSTM**: It processes the input sequence from start to end, capturing dependencies from past to future. Sequentially updates hidden states based on current input and previous hidden state.

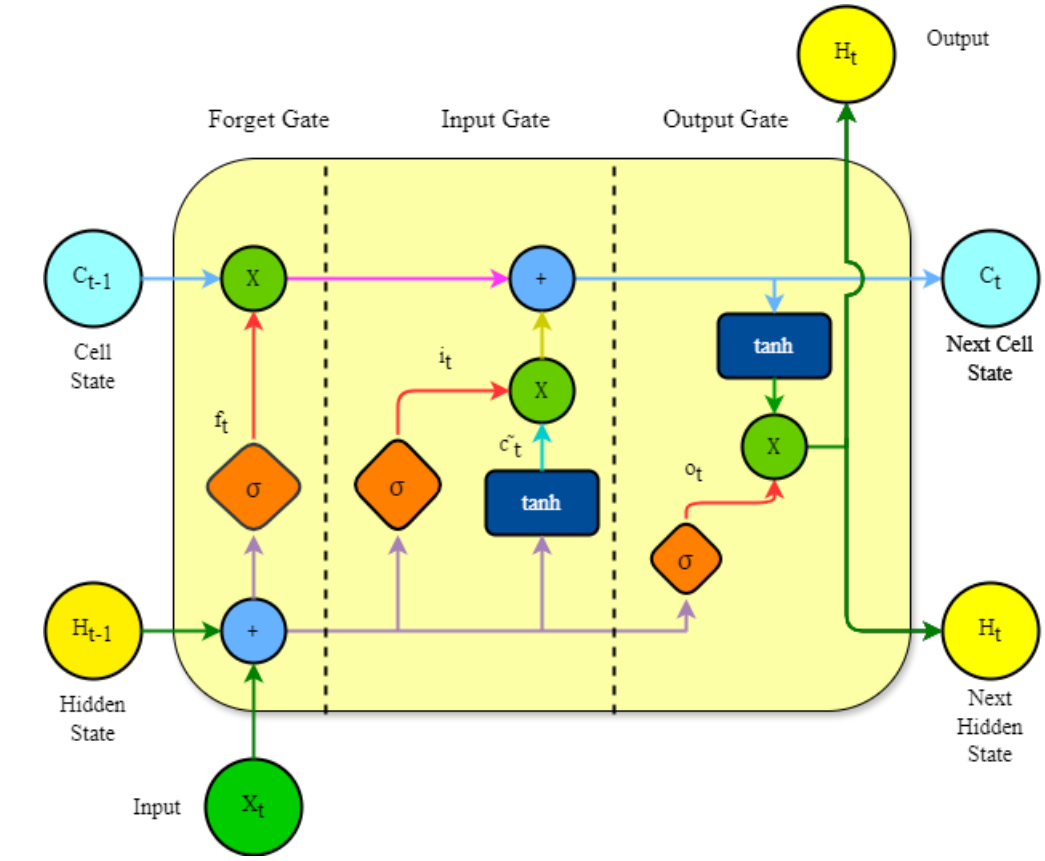
$$\overrightarrow{h_t} = LSTM\ (x_t, \overrightarrow{h_{t-1}}, \overrightarrow{C_{t-1}}) \tag{3.26}$$

- **Backward LSTM**: It processes the input sequence from end to start, capturing dependencies from future to past. Sequentially updates hidden states based on current input and subsequent hidden state.

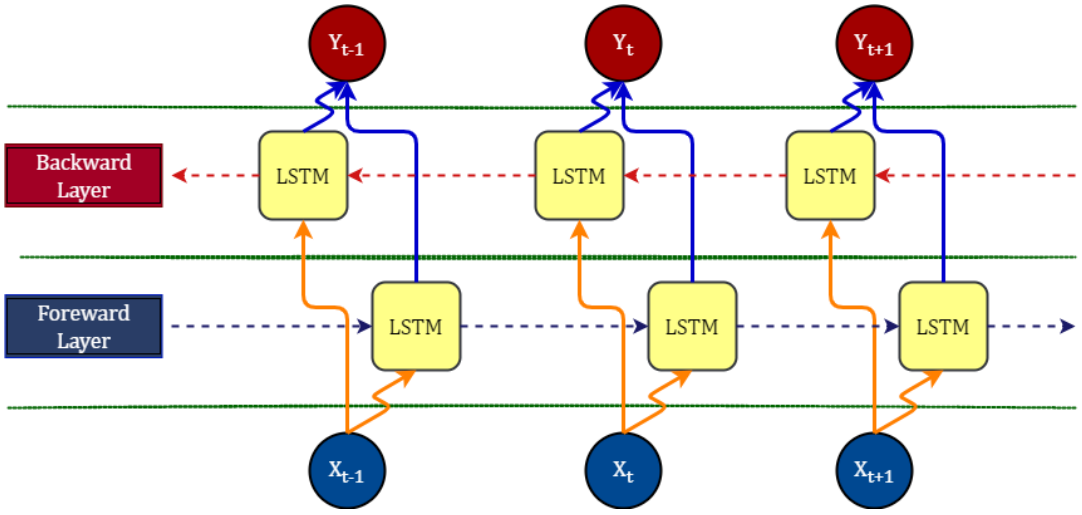
$$\overleftarrow{h_t} = LSTM\ (x_t, \overleftarrow{h_{t-1}}, \overleftarrow{C_{t-1}}) \tag{3.27}$$

- **Concatenation**: Combines the information from both forward and backward passes. Concatenates the hidden states from the forward and backward LSTMs for each time step t.

$$h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \tag{3.28}$$



(a)



(b)

Fig. 3.6: The structural design of (a) LSTM and (b) Bi-LSTM

The output shapes and parameters for each layer are summarized in Table 3.6.

Table 3.6: Summary of the model architecture

Layers	Output Shapes	Parameters
Input Layer	(None, 92)	0
Embedding	(None, 92, 512)	3,276,852
Bidirectional LSTM	(None, 92, 128)	295,424
Dropout	(None, 92, 128)	0
Transformer Block	(None, 92, 512)	6,828,544
Concatenate (Transformer, Bi-LSTM)	(None, 92, 640)	0
GlobalMaxPooling1D	(None, 640)	0
Dropout	(None, 640)	0
Dense	(None, 1)	641

Chapter 4 Experimental Setup

Chapter Outline

4.1 Overview

4.2 Experimental Settings

4.1 Overview

The experimental settings, including data preprocessing, model configuration, and training procedures, are discussed in this chapter to provide a clear understanding of the methodological framework. All programming tasks in this project have been conducted using the Python language, and the Google Colab Pro environment has been employed to facilitate the work. A detailed description of these uses has been provided in the following section.

4.2 Experimental Settings

This paper extensively utilizes Python programming language version 3.10.12 in conjunction with the Pandas library tool version 2.0.3, NumPy version 1.23.5, and Matplotlib version 3.7.1. These widely recognized software libraries are renowned for their effectiveness in data analysis and visualization tasks, making them a pivotal component of the research endeavor. The operational functions are exclusively conducted within the Google Colab Pro environment, which boasts a robust hardware configuration with more memory and longer runtimes than the free version, allowing for more intensive computations. All deep learning operations are executed using the TensorFlow framework version 2.15.0, ensuring compatibility and optimal performance across the board.

The research incorporates a Transformer architecture augmented with Bi-LSTM, a computational process that is complex and time-consuming when executed on a CPU. In NLP or LLM tasks, transformer architecture greatly benefits from GPU acceleration. GPUs, or Graphics Processing Units, are optimized for parallel computations, which are pervasive in deep learning algorithms owing to their extensive matrix operations. With their self-attention mechanisms and multi-layered architecture, transformer models often demand substantial computational resources, particularly during training. They are utilizing GPUs results in faster training times than CPUs, enabling researchers and practitioners to experiment with larger models and datasets efficiently. This acceleration is particularly evident when working with large Transformer-based models.

In this research, Google Colab Pro with an L4 GPU has been utilized to enhance the robustness of the computational training process and reduce time consumption. The GPU type primarily employed is the NVIDIA L4 GPU, featuring the NVIDIA Ada Lovelace architecture. This architecture boasts a higher memory capacity of 24 GB, 7680 CUDA cores, and 240 Tensor

cores. It represents one of NVIDIA's latest GPU releases, tailored to offer high performance for AI and machine learning tasks. The L4 GPU is precisely engineered to significantly improve computational performance, rendering it well-suited for training large machine learning models, executing deep learning algorithms, and conducting complex data analyses. Figure 4.1 displays the Python environment and hardware configuration used in the study.

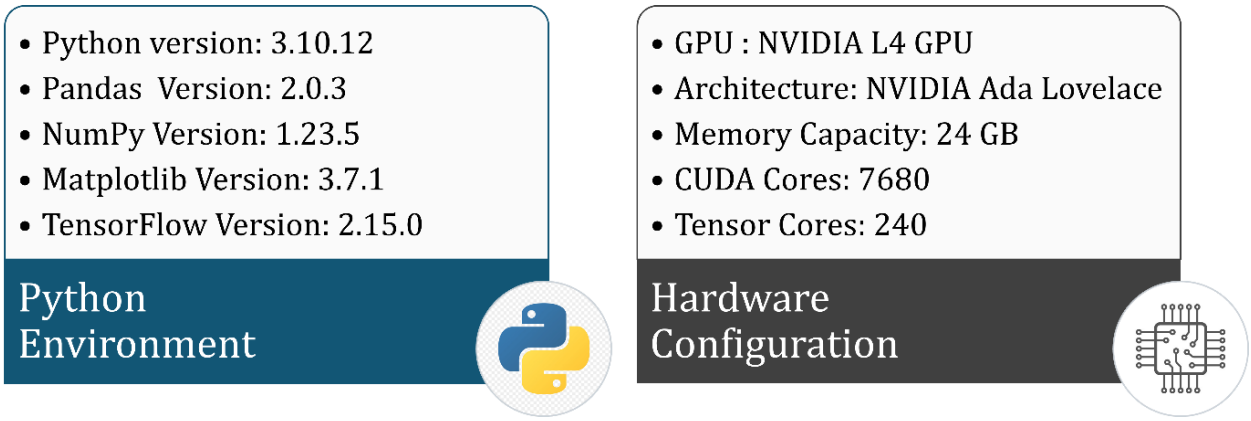


Fig. 4.1: Experimental settings

Chapter 5 Evaluation

Chapter Outline

5.1 Overview

5.2 Evaluation Methods

5.1 Overview

In the context of model performance analysis, evaluation metrics are quantitative measures used to assess the effectiveness of machine learning or statistical models in making predictions. These metrics help determine how well a model performs against real-world data, based on criteria relevant to the specific problem it's designed to solve. The evaluation metrics which are utilized in the research work for model performance has been described in the next section.

5.2 Evaluation Methods

As mentioned earlier, the target classes are 'true' and 'fake' news, making the proposed model a binary classification model. The effectiveness of the binary model in any detection scheme relies on its evaluation metrics, which are represented by the confusion matrix. A confusion matrix provides a comprehensive overview of a classification algorithm's performance by presenting essential relative information. In this work, six well-known performance metrics have been derived from the confusion matrix of the detection model which are described below. In the context of fake news detection, the confusion matrix consists of the following four outcomes: True Positives, True Negatives, False Positives, and False Negatives.

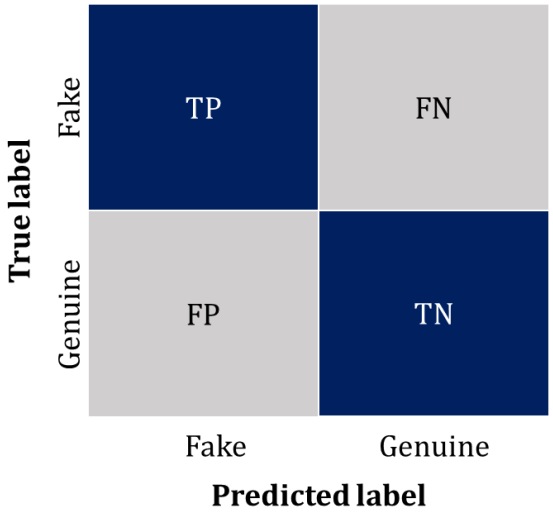


Fig. 5.1: Confusion matrix of fake news detection model

- True Positives (TP): The number of fake news tweets correctly identified as fake by the model.
- True Negatives (TN): The number of genuine news tweets correctly identified as genuine by the model.
- False Positives (FP): The number of genuine news tweets incorrectly identified as fake by the model. This is also known as a Type I error.
- False Negatives (FN): The number of fake news tweets incorrectly identified as genuine by the model. This is also known as a Type II error.

I. Accuracy: Accuracy measures the proportion of correctly classified instances (both fake and genuine news tweets) out of the total instances. It gives an overall effectiveness of the model.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{5.1}$$

II. Precision: Precision measures the proportion of true positive predictions out of all positive predictions. It indicates how many of the news tweets identified as fake are actually fake.

$$Precision = \frac{TP}{TP+FP} \tag{5.2}$$

III. Recall: Recall (also known as Sensitivity or True Positive Rate or Detection Rate) measures the proportion of true positive cases out of all actual positive cases. It reflects the model's ability to identify all actual fake news articles.

$$Recall/TPR/Sensitivity = \frac{TP}{TP+FN} \tag{5.3}$$

IV. F1-Score: The F1-Score is the harmonic means of precision and recall. It provides a single metric that balances the trade-off between precision and recall, especially useful when the dataset is imbalanced.

$$F1\ Score = \frac{2\times Precision\times Recall}{Precision+Recall} \tag{5.4}$$

V. False Positive Rate (FPR): FPR measures the proportion of actual negatives (genuine news) that are incorrectly identified as positive by the model.

$$FPR = \frac{FP}{FP+TN} \tag{5.5}$$

VI. Specificity: Specificity, also known as the true negative rate (TNR), measures the proportion of actual true news tweets that are correctly identified by the model. It focuses on the model's ability to correctly reject true news as not fake.

$$\text{Specificity/TNR} = \frac{TN}{TN+FP} \tag{5.6}$$

VII. Error Rate: The error rate is the proportion of all predictions that are incorrect. It is a measure of how often the classifier makes a wrong prediction.

$$\text{Error Rate} = \frac{FP+FN}{TP+TN+FP+FN} = 1 - Accuracy \tag{5.7}$$

These metrics, derived from the confusion matrix, provide a robust evaluation framework for assessing the fake news detection model's performance. For optimal model performance, accuracy, precision, recall, F1-score, and specificity should all be high. Conversely, the false positive rate (FPR) and error rate should be low. This combination ensures the model accurately identifies both true and fake news, minimizing incorrect classifications.

VIII. ROC Curve: The Receiver Operating Characteristic (ROC) curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The ROC curve is created by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. It is particularly useful for comparing the performance of different models or classifiers.

IX. ROC-AUC Score: The Area Under the ROC Curve (AUC) quantifies the overall ability of the model to discriminate between fake and genuine classes. The ROC-AUC score ranges from 0 to 1.

Chapter 6 Results and Discussion

Chapter Outline

6.1 Overview

6.2 Classification Results of The Proposed Model

6.3 Comparative Analysis with Other Deep Learning Classifiers

6.4 Evaluation of Model Performance Across Diverse Datasets

6.1 Overview

The fake news detection model from tweets relies on the evaluation metrics scores. This section has been organized into two sub-sections for better understanding. Section 6.1 stated the overview of this chapter. Section 6.2 highlights the performance characteristics for fake news detection, while section 6.3 offers a comparative analysis of the overall results across individual deep-learning classifiers and state-of-the-art models in the field. Lastly, Section 6.4 assesses the model's performance across diverse datasets, providing insights into how well the proposed system adapts to varying data characteristics and conditions. This section aims to establish the robustness and versatility of the model in different informational environments.

6.2 Classification Results of The Proposed Model

This section finds the results of the fake news detection model including different evaluation metrics which have been derived from confusion matrix. The confusion matrix of the proposed model incorporating transformer and Bi-LSTM has been displayed in Fig. 6.1.

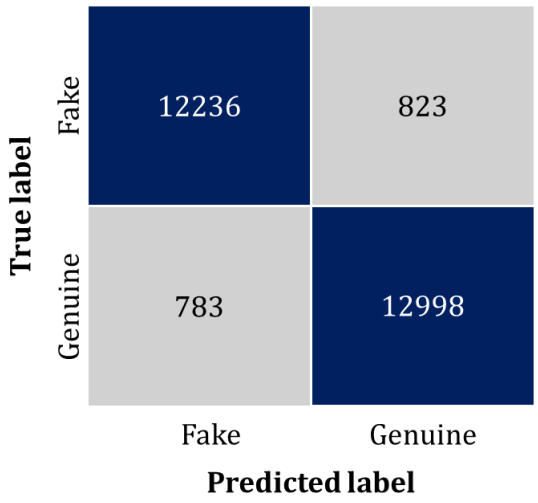


Fig. 6.1: Confusion matrix for the proposed model

The performance of the fake news detection model from tweets has been quantitatively assessed using a set of evaluation metrics which have been previously discussed. Table 6.1 below summarizes these metrics and their respective values which have been derived from confusion matrix in Fig 6.1.

Table 6.1: Experimental results for the fake news detection model

Evaluation Metrics	Value
Accuracy	94.02 %
Error Rate	5.98 %
Recall	93.70 %
Precision	93.99 %
F1-Score	93.84 %
FPR	5.68 %
Specificity	94.32 %
ROC-AUC	0.9614

These metrics collectively demonstrate that the fake news detection model performs well, with high accuracy, precision, recall, and specificity, along with a low error rate and FPR. The balanced F1-Score further indicates the model's reliability and effectiveness in detecting fake news from tweets. Moreover, the ROC-AUC score is close to 1, highlighting the model's excellent performance distinguishing between the fake and truth tweets.

Table 6.2 presents the detection rates for both the 'fake' and 'genuine' news classes. The detection rate is slightly higher for genuine news than for fake news. This discrepancy might be attributed to the greater number of genuine news instances in the test set. Nevertheless, these high detection rates demonstrate the model's robust capability to classify fake and genuine news accurately.

Table 6.2: Detection rate for each class

Class Name	Detection Rate (%)
Fake News	93.6978
Genuine News	94.3183

6.3 Comparative Analysis with Other Deep Learning Classifiers

In order to evaluate the performance of the proposed model on a broader aspect, the entire workflow has been reconstructed for several deep learning classifiers. In this reconstruction, all preprocessing stages for the end-to-end workflow remain unchanged. However, instead of utilizing the proposed hybrid classification model, CNN and RNN-based classifiers have been employed for detecting fake news. A comparative analysis has been conducted using the 'Truthseeker' dataset for fake news detection. The observational results for the model have been inspected across a total of eight evaluation metrics, including accuracy, precision, recall, F1-score, specificity, and FPR. The results of all classifiers, including our proposed approach, are presented in Table 6.3.

Table 6.3: Comparison against other deep learning classifier with proposed model

Evaluation Metrics	MLP (%)	LSTM + Bi-LSTM (%)	CNN + Bi-LSTM (%)	Transformer + CNN (%)	Transformer + LSTM (%)	Proposed (%)
Accuracy	89.02	91.89	90.75	93.11	93.51	**94.02**
Error Rate	10.98	8.11	9.25	6.89	6.49	**5.98**
Recall	91.06	90.86	88.57	91.88	92.59	**93.70**
Precision	86.98	92.34	92.12	93.82	**93.99**	**93.99**
F1-Score	88.97	91.59	90.31	92.84	93.28	**93.84**
FPR	12.92	7.14	7.18	5.73	**5.61**	5.68
Specificity	87.08	92.86	92.82	94.27	**94.39**	94.32

From Table 6.3 the proposed model excels in most metrics, particularly accuracy, recall, precision, and F1-Score. These metrics demonstrate its effectiveness and reliability in detecting fake news from tweets, making it the best choice among the compared models. Its high performance in key areas ensures that it not only correctly identifies fake news but also minimizes errors, making it a robust tool for fake news detection.

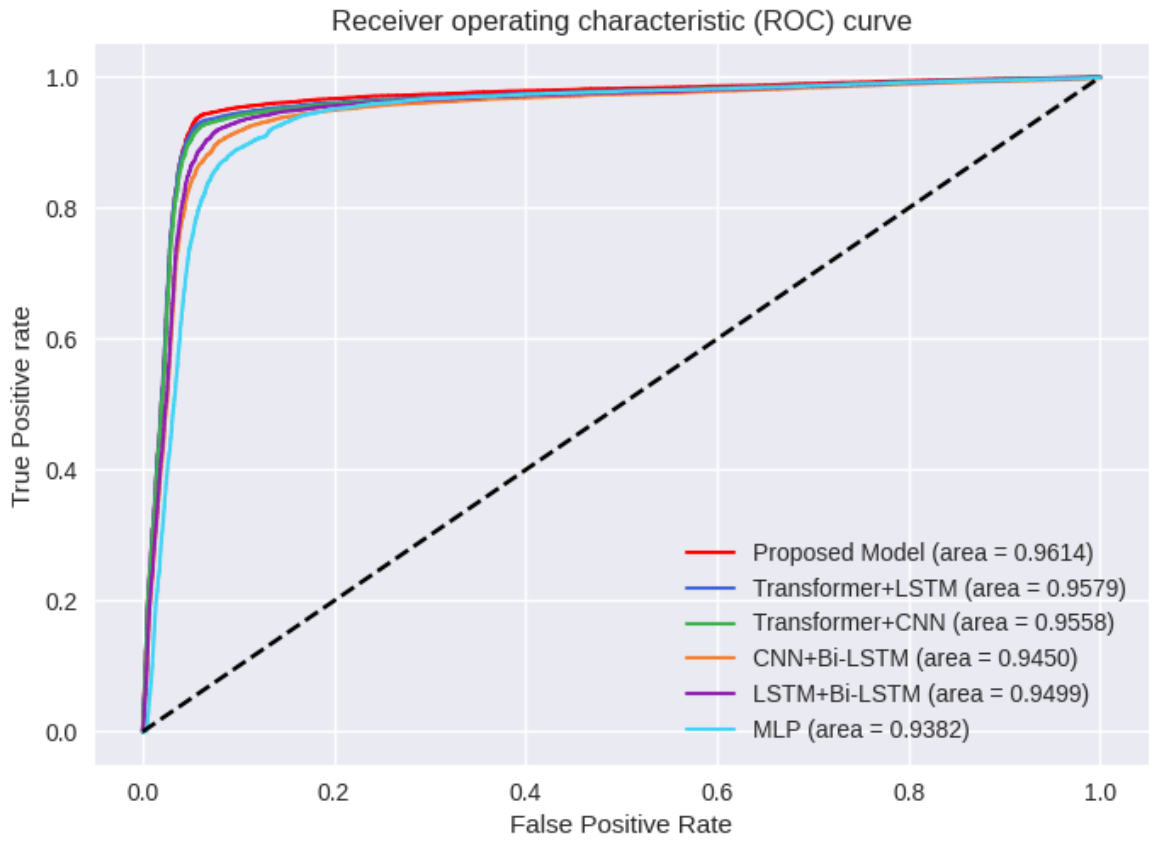


Fig. 6.2: ROC curves for different classifiers

The ROC curve presented in Fig. 6.2 above compares the performance of various fake news detection models. The proposed model, represented by the red line, exhibits the highest area under the curve (AUC) of 0.9614, indicating superior ability to distinguish between true and fake news. Other models, such as Transformer+LSTM (0.9579) and Transformer+CNN (0.9558) also show high performance but fall short compared to the proposed model. The higher AUC value of the proposed model signifies its greater accuracy and reliability in predicting fake news, making it the most effective among the evaluated models.

6.4 Evaluation of Model Performance Across Diverse Datasets

Evaluating our model using multiple datasets is beneficial for several reasons. First, it ensures the robustness and generalizability of the model across diverse data sources. Different datasets may have unique characteristics and variations in structure and content, which helps verify that the model performs consistently well in various real-world scenarios. Second, it allows for a comprehensive comparison with existing models and benchmarks, highlighting the strengths

and weaknesses of our approach relative to others in the field. This comparative analysis can reveal insights into the model's performance, such as its ability to handle different types of fake news, scalability, and adaptability to new data. Lastly, by testing multiple datasets, any overfitting issues can be identified, and necessary adjustments can be made to improve the model's accuracy, precision, recall, and overall effectiveness in fake news detection. In Table 6.4, a comparative analysis of different datasets for the proposed approach, along with their descriptions, has been presented. The evaluation metrics for each dataset have been derived from the confusion matrices displayed in Fig. 6.3.

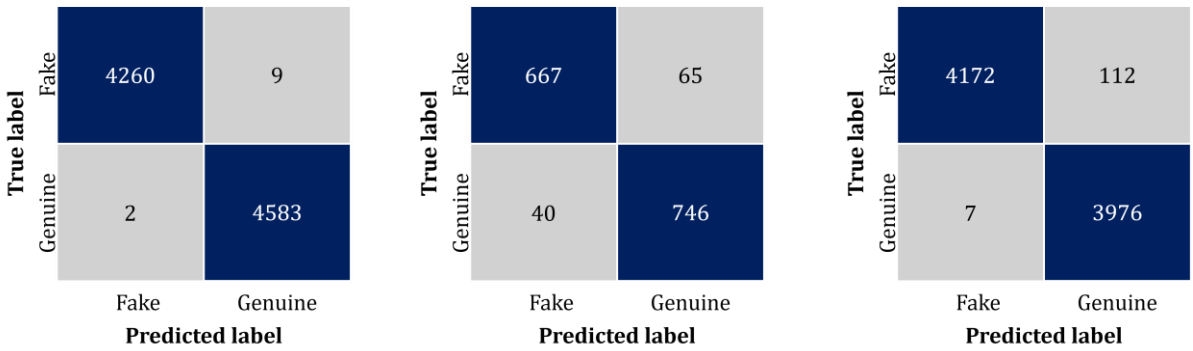


Fig. 6.3: Confusion matrices of the proposed model on different datasets

Table 6.4: Comparative analysis of the model's performance over multiple fake news detection datasets

Dataset Description					Evaluation					
Dataset	**Instances**	**Topic Domain(s)**	**Platform**	**Year**	**Accuracy**	**Recall**	**Precision**	**F1-Score**	**Specificity**	**ROC-AUC Score**
ISOT Fake News Dataset (Ahmed et al., 2017) (Sastrawan et al., 2021)	Train- 36,539 Test- 8,857	News, Politics	Website, Social Media	2017	99.86	99.78	99.95	99.87	99.95	0.9999
COVID19-FNIR (Shukla, 2021)	Train- 6,070 Test- 1,518	COVID-19	Poynter, Twitter	2021	93.08	91.12	94.34	92.70	94.91	0.9827
Fake News Detection Dataset English (Fake News Detection DataSet_English, 2022)	Train- 36,000 Test- 8,267	Journalism, Politics	News, Website, Articles, Twitter	2021	98.56	97.39	99.83	98.59	99.82	0.9995
Truth Seeker	Train- 107,358 Test- 26,840	Politics, General Events, Health, Crime, Science	Twitter	2022	94.02	93.07	93.99	93.84	94.82	0.9614

Table 6.4 shows that the proposed model has performed well across diverse datasets, demonstrating excellent accuracy and ROC-AUC scores, which indicate its effectiveness in distinguishing between genuine and fake news. Despite these dataset's structural differences and varied topic domains, the model maintained robust performance, showcasing its generalizability to real-world scenarios beyond just Twitter. The Truth Seeker dataset, in particular, highlighted the model's adaptability with an accuracy, precision and recall score of 94.04%, 93.07% and 93.99%, respectively, with the ROC-AUC score of 0.9614. While the COVID19-FNIR and Fake News Detection Dataset English also demonstrated robust results, and the ISOT Fake News Dataset exhibited near-perfect scores, these datasets are often less challenging due to narrower topic domains or less noisy data. The Truth Seeker dataset offers a more comprehensive evaluation environment with its diverse range of topics, including politics, general events, health, crime, and science. The fact that our model performs exceptionally well on this dataset, which includes a variety of real-world complexities, underscores its robustness and generalizability. Although the performance on the COVID19-FNIR dataset was slightly lower, likely due to a smaller sample size, the overall results affirm that the proposed model excels in handling diverse data sources and complex real-world contexts, making it a robust and effective solution for fake news detection.

Chapter 7 Conclusion

Chapter Outline

7.1 Conclusion

7.2 Future Work

7.1 Conclusion

In today's social media-driven era, content is generated every second, including a significant amount of fake news and rumors, often without traceable sources. It has become crucial to identify misinformation early to prevent social unrest and stop the spread of falsehoods. While considerable research has been conducted in this field, there still needs to be a benchmark dataset that can support the development of a robust model capable of handling diverse content types. The Truth Seeker dataset, a recent benchmark featuring a wide range of topic domains, has yet to be used in previous research, making it an ideal basis for training, and testing the proposed model to ensure its robustness.

To achieve the research objectives, the study successfully developed and assessed a novel hybrid model that combines transformer architecture and Bidirectional LSTM for effective fake news detection on Twitter. The application of conventional text cleaning methods, along with BERTweet for tokenization, significantly enhanced the model's ability to interpret and analyze the contextual nuances of tweets. This approach has led to more accurate and reliable detection of misinformation. Through rigorous testing across three additional fake news datasets, the model has proven its effectiveness and adaptability in various scenarios of misinformation. A comparative analysis with other deep learning classifiers and state-of-the-art models has further confirmed the superior performance of the proposed model. Overall, this research significantly contributes to fake news detection, providing a dependable and efficient tool to combat misinformation in digital media.

7.2 Future Work

For future research, there are promising avenues to further enhance the effectiveness of models used in fake news detection by integrating and fine-tuning advanced pre-trained models like BERT or RoBERTa. These transformer-based models, renowned for their deep contextual understanding, could be adapted to specific datasets that demand high accuracy, addressing the nuanced complexities of language used in fake news. By fine-tuning these models, researchers can tailor them to capture subtler linguistic cues and hidden semantic patterns that typical models might overlook. Additionally, exploring hybrid models that combine the strengths of different architectural approaches or employing ensemble methods could provide a robust framework that leverages collective insights from multiple models, thereby increasing reliability and predictive power. Such advancements could significantly push the boundaries of

current methodologies, offering more precise and scalable solutions in the rapidly evolving landscape of fake news detection.

References

Ahmed, H., Traore, I., & Saad, S. (2017). Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. Lecture Notes in Computer Science, 127–138. https://doi.org/10.1007/978-3-319-69155-8_9

Alghamdi, J., Lin, Y., & Luo, S. (2023). Towards COVID-19 fake news detection using transformer-based models. Knowledge-based Systems, 274, 110642. https://doi.org/10.1016/j.knosys.2023.110642

Allcott, H., & Gentzkow, M. (2017). Social Media and Fake News in the 2016 Election. the Journal of Economic Perspectives/the Journal of Economic Perspectives, 31(2), 211–236. https://doi.org/10.1257/jep.31.2.211

Conroy, N. K., Rubin, V. L., & Chen, Y. (2015). Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology*, *52*(1), 1–4. https://doi.org/10.1002/pra2.2015.145052010082

Dadkhah, S., Zhang, X., Weismann, A. G., Firouzi, A., & Ghorbani, A. A. (2023). The Largest Social Media Ground-Truth Dataset for Real/Fake Content: TruthSeeker. IEEE Transactions on Computational Social Systems, 1–15. https://doi.org/10.1109/tcss.2023.3322303

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. https://doi.org/10.18653/v1/N19-1423

Fake News Detection DataSet_English. (2022). Datasets at Hugging Face. https://huggingface.co/datasets/ErfanMoosaviMonazzah/fake-news-detection-dataset-English

Fake News Statistics & Facts (2023) — Redline Digital. (2023). https://redline.digital/fake-news-statistics/

Flood, A. (2018, February 9). *Fake news is "very real" word of the year for 2017*. The Guardian. https://www.theguardian.com/books/2017/nov/02/fake-news-is-very-real-word-of-the-year-for-2017

Kemp, S. (2024, January 31). *Digital 2024: Global Overview Report — DataReportal – Global Digital Insights*. DataReportal – Global Digital Insights. https://datareportal.com/reports/digital-2024-global-overview-report

Kumar, S., & Arora, B. (2021). A Review of Fake News Detection Using Machine Learning Techniques. 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC). https://doi.org/10.1109/icesc51422.2021.9532796

Micich, A., & Cross, R. (2023, November 22). *How misinformation on social media has changed news*. U.S. PIRG Education Fund. https://pirg.org/edfund/articles/misinformation-on-social-media/

Newman, N. (2020, June 23). *Overview and Key Findings of the 2020 Digital News Report*. Reuters Institute Digital News Report. https://www.digitalnewsreport.org/survey/2020/overview-key-findings-2020/

Rahman, A. U., Chaudhry, H. N., Asim, M. M., & Kulsoom, F. (2023). A Transformer-based approach for Fake News detection using Time Series Analysis. https://doi.org/10.1109/imtic58887.2023.10178457

Ruchansky, N., Seo, S., & Liu, Y. (2017). CSI: A hybrid deep model for fake news detection. Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 797-806. https://doi.org/10.1145/3132847.3132877

Sastrawan, I. K., Bayupati, I. P. A., & Arsa, D. M. S. (2021, September 14). Fake News Dataset. https://doi.org/10.17632/945z9xkc8d.1

Shen, S., & Fan, J. (2022). Emotion Analysis of Ideological and Political Education Using a GRU Deep Neural Network. Frontiers in Psychology, 13. https://doi.org/10.3389/fpsyg.2022.908154

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining, 19(1), 22–36. https://doi.org/10.1145/3137597.3137600

Shukla, D. (2021, July 22). Covid-19 Fake News Infodemic Research Dataset (CoVID19-FNIR Dataset). IEEE DataPort. https://dx.doi.org/10.21227/b5bt-5244

Truth Seeker Dataset 2023 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2023). https://www.unb.ca/cic/datasets/truthseeker-2023.html

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. Advances in neural information processing systems, *30*, 5998–6008. https://arxiv.org/pdf/1706.03762v5

Wang, S. A., Pang, M. S., & Pavlou, P. A. (2021). Seeing Is Believing? How Including a Video in Fake News Influences Users' Reporting the Fake News to Social Media Platforms. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3909942

Wang, S., Pang, M. S., & Pavlou, P. A. (2018). “Cure or Poison?” Identity Verification and the Spread of Fake News on Social Media. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3249479

Zhou, X., & Zafarani, R. (2020). A Survey of Fake News: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys*, *53*(5), 1–40. https://doi.org/10.1145/3395046