*International Journal of Data Science and Analysis*
2025, Vol. 11, No. 2, pp. 23-45
https://doi.org/10.11648/j.ijdsa.20251102.12

Research Article

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

**Kowshik Sankar Roy***, **Farhana Akter Bina**

Department of Statistics and Data Science, Jahangirnagar University, Savar, Dhaka, Bangladesh

## 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. Additionally, the rapid evolution of deceptive tactics makes traditional methods less effective, necessitating new approaches that can adapt to emerging misinformation patterns. In response to the challenges, 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 Bi-LSTM technologies in a concatenation mode, enhanced by advanced preprocessing steps, including BERTweet, for effective tokenization and contextual understanding. An ablation study highlights the individual contributions of the Bi-LSTM and Transformer components, as well as their combined effect, demonstrating their critical roles in enhancing the model's performance. Compared to conventional classifiers, including various CNN, LSTM, Bi-LSTM, BERT and Transformer configurations, the proposed model demonstrates superior performance, as evidenced by comprehensive statistical testing. TweetGuard achieves an accuracy of 94.02%, an F1-score of 93.84%, and a ROC-AUC score of 0.9614 on the TruthSeeker dataset. Additional metrics, such as a Matthews Correlation Coefficient (MCC) of 0.8802 and a fake news detection rate of 93.70%, also demonstrate the model's stability and robustness. Its effectiveness and generalizability 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.

## Keywords

Fake News Detection, Transformer Architecture, Bi-LSTM, Social Media Misinformation, Natural Language Processing, Deep Learning, Security, Social Security

*Corresponding author: kowshikroy777@gmail.com (Kowshik Sankar Roy)

**Received:** 22 April 2025; **Accepted:** 3 May 2025; **Published:** 10 June 2025

# 1. Introduction

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 [1]. 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 [2].

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 [3].

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 [4, 5]. 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 [6]. 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 [7]. 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 [8].

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 [9]. 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 [10].

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 research utilizes a newly published benchmark dataset, "TruthSeeker," which is notable for its inclusion in this work as it is new and not yet widely adopted in the latest literature. The overall contribution of this work has been established around five folds. These are stated below:

A novel hybrid architecture named “TweetGuard”, has been developed that combines Transformer and Bidirectional LSTM architectures in a concatenation mode, with the Transformer block serving as the backbone, tailored for detecting fake news on Twitter. This approach leverages the strengths of both architectures to effectively handle short, noisy, and context-rich textual data.

Advanced text processing techniques have been introduced to optimize the accuracy of fake news detection. A robust pipeline for text cleaning and standardization was implemented, coupled with BERTweet tokenization. This integration enhances the model's ability to capture nuanced contextual information crucial for accurate classification of misinformation in tweets.

An ablation study has been conducted to highlight the individual contributions of the Bi-LSTM and Transformer components, as well as their combined effect. This study demonstrates the critical roles these components play in enhancing the model's performance for detecting fake news.

Thorough evaluations have been conducted using multiple newly integrated datasets focused on fake news detection. This comprehensive approach demonstrates the robustness and generalizability of the model across various types of content and data sources, reinforcing its reliability in real-world applications.

A detailed benchmarking analysis has been included against several models currently used in the field. Through rigorous comparative assessments with statistical tests, the superior performance and adaptability of the hybrid model in different scenarios have been highlighted, underscoring its potential as a leading solution for detecting misinformation on social media platforms like Twitter.

The structure of the remainder of this paper is outlined as follows: Section 2 reviews related works in the field. Section 3 presents the complete architecture of our proposed model along with a description of the datasets used. Section 4 details all the experimental evaluations conducted in this study. Experimental settings are discussed in Section 5, while Section 6 provides a comprehensive analysis of the experimental results and discusses the implications of the findings. The paper concludes with a summary of the work and final thoughts in Section 7.

## 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 [11]. 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 [12]. 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 [13].

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 [14, 15]. 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 [16]. 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 [17]. In [18] 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. In [19], authors 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 [20]. The study by [21] 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 [22, 23]. 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 [24]. The authors of [25] 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%. In [26], authors propose 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 [27]. 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. The authors of [28] 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.

Recent advancements in fake news detection have been significantly driven by fine-tuned transformer-based models. BERT (Bidirectional Encoder Representations from Transformers) and its variants, such as RoBERTa and DistilBERT, have been widely adopted for their ability to capture deep contextual semantics and achieve high accuracy across various NLP tasks. Several studies have demonstrated the effectiveness of such models in misinformation classification. For instance, the GBERT framework combines the strengths of BERT and GPT to achieve a high F1-score of 96.23% on real-world datasets, showing the power of hybrid transformer models in capturing generative and contextual features [29]. Another notable work, DeepTweet, leverages transformer-based embeddings and a tailored attention mechanism to outperform traditional deep learning approaches in fake news detection on Twitter, further confirming the efficacy of transformer-only architectures in handling social media text [30]. However, despite their strong performance, these transformer-based models tend to be computationally intensive and less efficient for real-time applications. This creates a trade-off between performance and efficiency. The proposed hybrid model aims to bridge this gap by combining a transformer encoder with a Bi-LSTM, leveraging both contextual attention and sequential dependency, while maintaining competitive performance with lower computational cost.

## 3. Proposed Approach

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 three consecutive sub-sections. Section 3.1 provides a summary of the proposed model and its overarching structure. Section 3.2 details the properties of the dataset utilized in the research. In Section 3.3, 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 3.4.

### 3.1. Proposed Architecture

As depicted in the Figure 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.

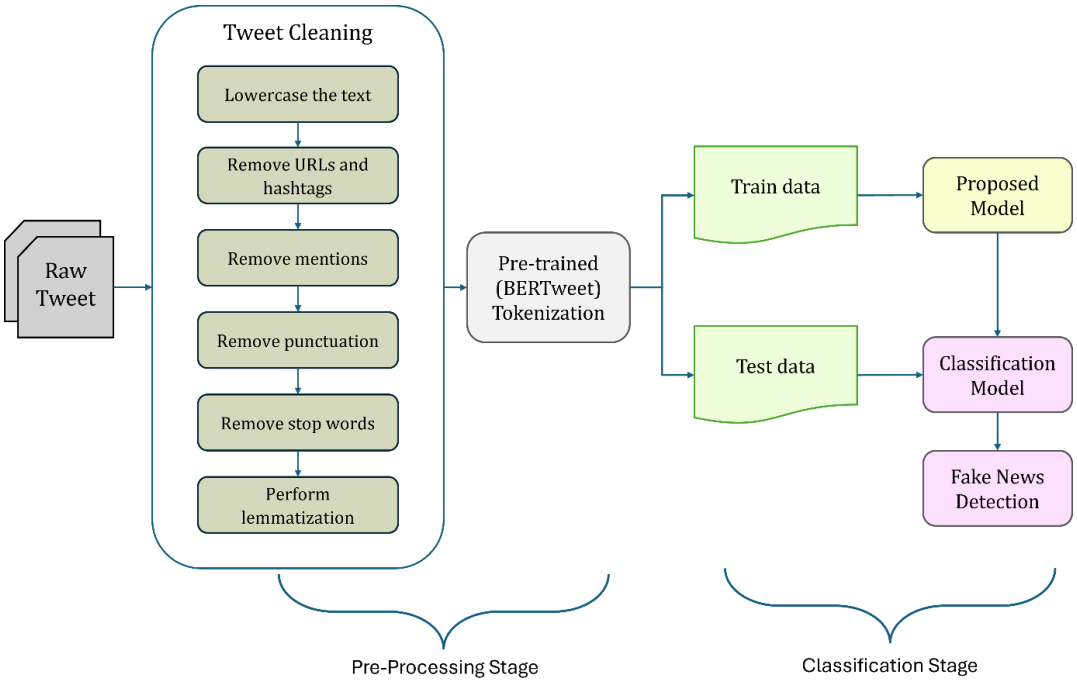



***Figure 1.*** *Flow diagram of overall process of the proposed method.*

## 3.2. 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 [31].

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 is 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 1.

***Table 1.*** *Dataset description.*

| Feature List | Description |
|---|---|
| author | Represents the author of the statement. |
| statement | Denotes the headline of a news article. |

| Feature List | Description |
|---|---|
| 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. |

## 3.3. 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 2 is a step-by-step example to illustrate the tweet cleaning stage.

***Table 2.*** *Preprocessing Steps for Tweet Text.*

| 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 restrictions! visit for details. ☺" | "president @user announced new covid-19 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" |

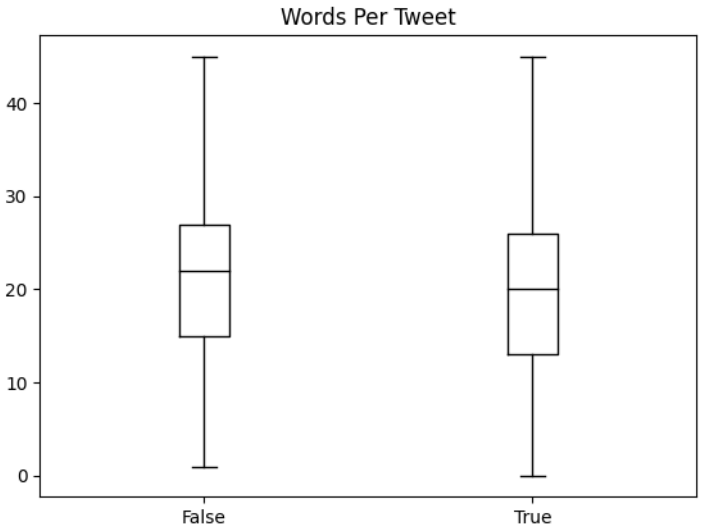



***Figure 2.*** *Word Count Distribution in Genuine and Fake News Tweets.*

From Figure 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. The truthfulness value has been then converted into a label-encoded format to pass through the classifier.

***Table 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 Figure 3 illustrates that, after label conversion, the class distribution is nearly balanced, with fake news posts comprising 2% fewer instances than true news posts.

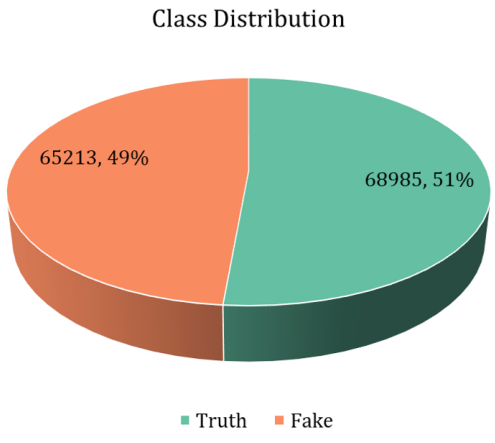



***Figure 3.*** *Distribution of class labels for true and fake news within the dataset.*

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 4 below.

***Table 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 |

## 3.4. 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 Figure 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 5.

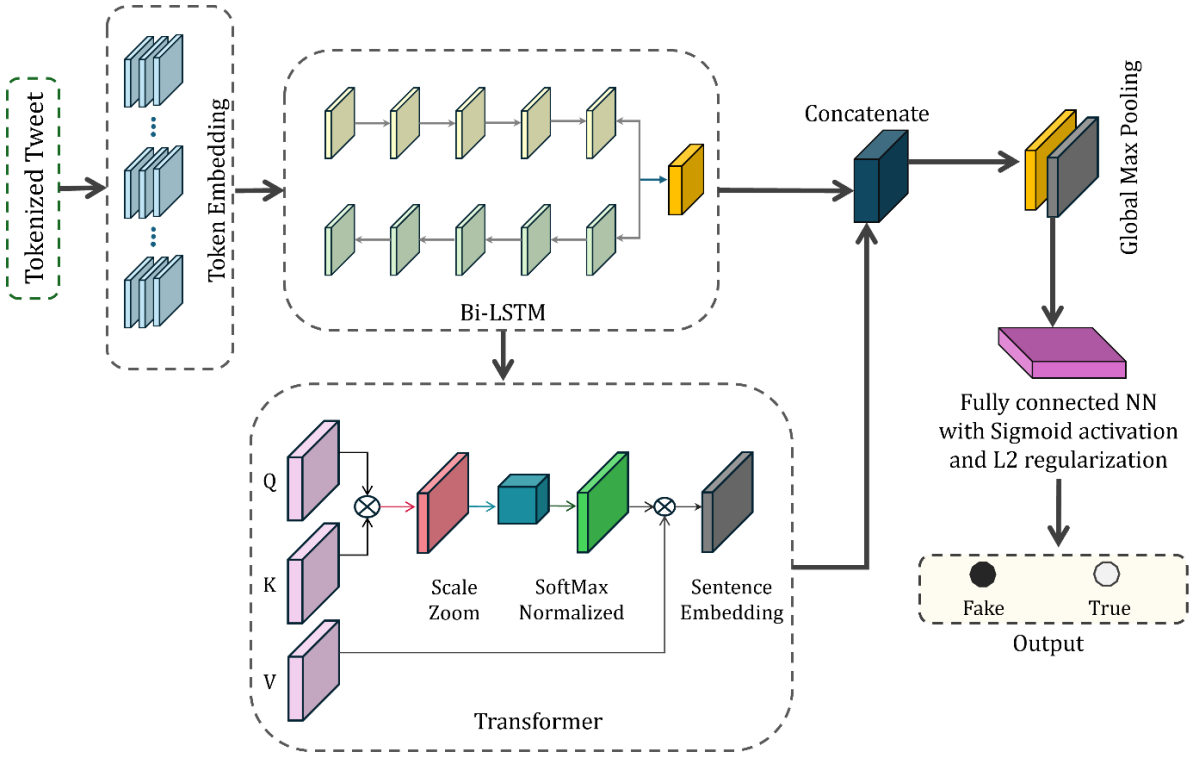



***Figure 4.*** *Architecture of the proposed hybrid model.*

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 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 |

### 3.4.1. 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. The operations of the layers are described in detail below, accompanied by the flow diagram shown in Figure 5.

I. 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 $X=\{x_1, x_2, \ldots, x_n\}$ 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:

$$Embedding\ (x_i) = E[Vocab(x_i)] \tag{1}$$

$E$ 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:

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

$$PE(pos, 2i+1) = \cos(\frac{pos}{10000^{2i/d_{model}}}) \tag{3}$$

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 $i$:

$$PE(pos) = \begin{bmatrix} PE(pos, 0) \\ PE(pos, 1) \\ \vdots \\ PE(pos, d_{model} - 1) \end{bmatrix} \tag{4}$$

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

$$Final\ \ Embedding\ \ (x_i) = Embedding\ \ (x_i) + PE(x_i) \tag{5}$$

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,

$$Q = X.W_Q,\ K = X.W_K,\ V = X.W_V \tag{6}$$

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:

$$Q_i = Split(Q, d_{model}/h),\ K_i = Split(K, d_{model}/h),\ V_i = Split(V, d_{model}/h) \tag{7}$$

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

$$Attention_i(Q,K,V) = softmax\left(\frac{Q_i.K_i^T}{\sqrt{d_k}}\right) * V_i \tag{8}$$

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:

$$MultiHead\ \ Output = Concatenate(Output_1, Output_2, \ldots, Output_h).W_0 \tag{9}$$

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,

$$LayerNorm_1(MHAttention\ \ Output_i) = LayerNorm(MHOutput_i + Residual_1) \tag{10}$$

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,

$$FFNN(x) = ReLU(X.W_1 + b_1).W_2 + b_2 \tag{11}$$

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,

$$LayerNorm_2(FFNN\ \ Output_i) = LayerNorm(FFNNOutput_i + Residual_2) \tag{12}$$

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.

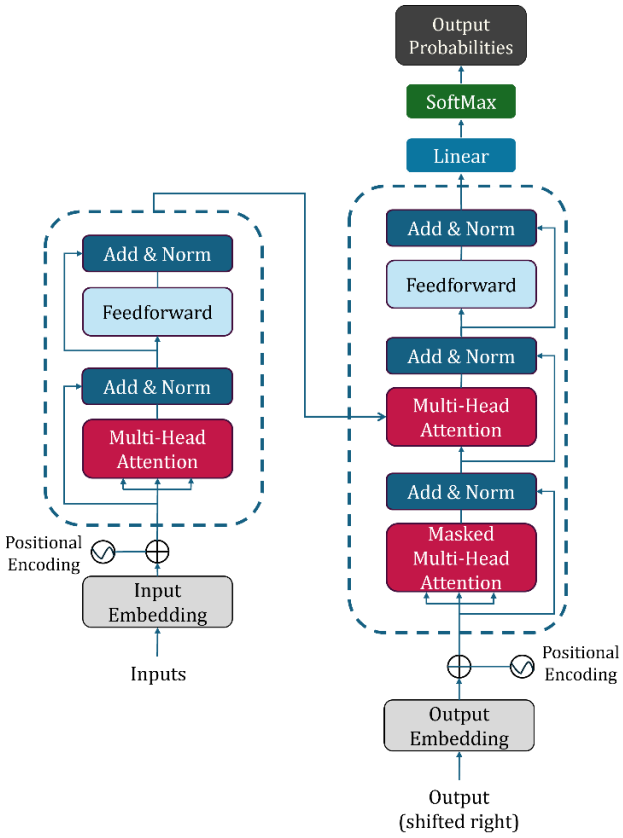



*Figure 5. The structural design of the Transformer.*

II. 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,

$$SelfAttention_i = MultiHead(Q_i, K_i, V_i) \quad (13)$$

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

$Y_{i-1}$ 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.

$$EncoderDecoderAttention_i = Attention(Q_i, K_{enc}, V_{enc}) \quad (14)$$

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

$Z$ 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.

$$FFNN_i = FFNN(Y_{i-1}) \quad (15)$$

$$LayerNorm_1 = LayerNorm(Y_{i-1} + SelfAttention_i) \quad (16)$$

$$LayerNorm_2 = LayerNorm(LayerNorm_1 + EncDecAttention_i) \quad (17)$$

$$Y_i = LayerNorm_2 + FFNN_i \quad (18)$$

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:

$$P = softmax(Y_N) \quad (19)$$

Where the final output of the decoder stack is $Y_N$.

### 3.4.2. 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 [32]. This makes it particularly effective for tasks where the context surrounding each word is crucial. The operational flow diagram of both an LSTM and a Bi-LSTM is illustrated in Figure 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.

$$f_t = \sigma(W_f . [h_{t-1}, x_t] + b_f) \quad (20)$$

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.

$$i_t = \sigma(W_i . [h_{t-1}, x_t] + b_i) \quad (21)$$

$$\widetilde{C_t} = \tanh(W_C . [h_{t-1}, x_t] + b_C) \quad (22)$$

$i_t$ 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.

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \quad (23)$$

$$h_t = o_t.\tanh(C_t) \quad (24)$$

$o_t$ 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

$$C_t = f_t.C_{t-1} + i_t.\widetilde{C}_t \quad (25)$$

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.

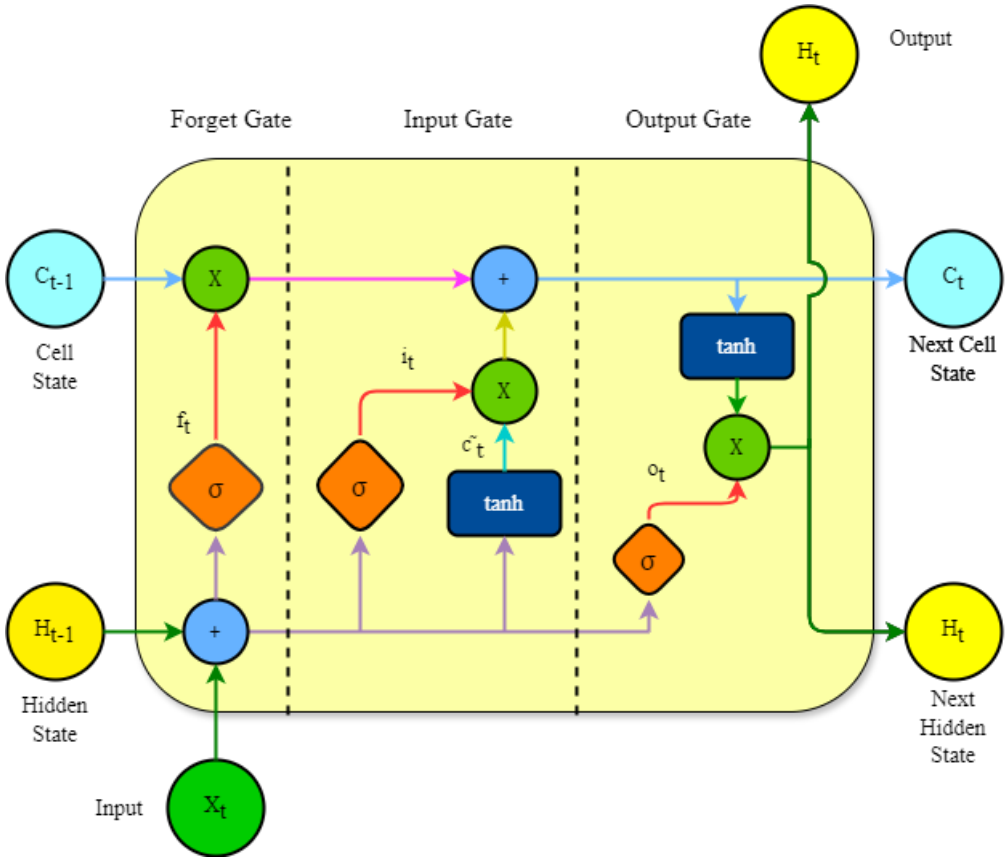
$$\overrightarrow{h_t} = LSTM\ (x_t, \overrightarrow{h_{t-1}}, \overrightarrow{C_{t-1}}) \quad (26)$$

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.

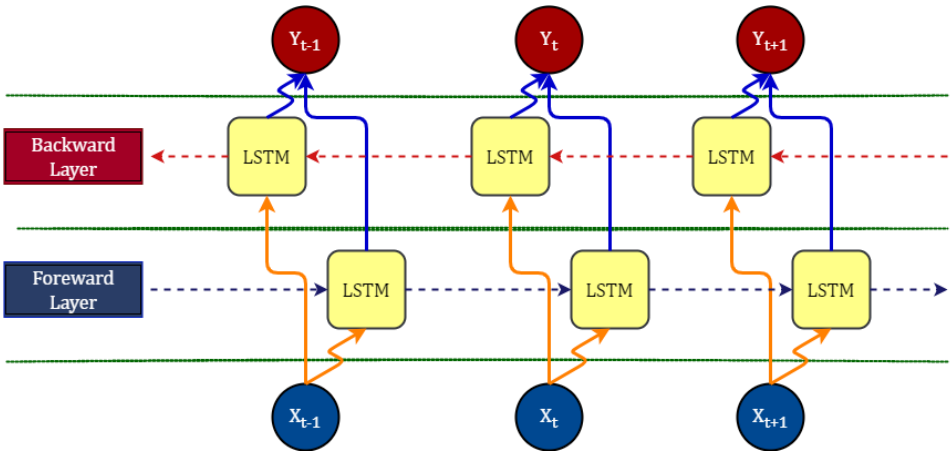
$$\overleftarrow{h_t} = LSTM\ (x_t, \overleftarrow{h_{t-1}}, \overleftarrow{C_{t-1}}) \quad (27)$$

Concatenation: Combines the information from both forward and backward passes. Concatenates the hidden states from the forward and backward LSTMs for each time stept.

$$h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \quad (28)$$




(a)




(b)

***Figure 6.*** *The structural design of (a) LSTM and (b) Bi-LSTM.*

The output shapes and parameters for each layer are summarized in Table 6 where each layer is sequentially connected to the previous layer. It delineates how the configuration of each layer, from input to output, is systematically documented in Table 6. Each layer in the model, such as the embedding, Bidirectional LSTM, Transformer block, Global MaxPooling and final dense layer, is meticulously outlined with its specific output shape and parameter count. This sequential connection implies that the data flow through the network follows a defined path, where the output of one layer serves as the input to the subsequent layer, culminating in a comprehensive understanding of how information is processed and transformed within the model architecture.

***Table 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 |

## 4. Experimental Setup

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 7 displays the Python environment and hardware configuration used in the study.

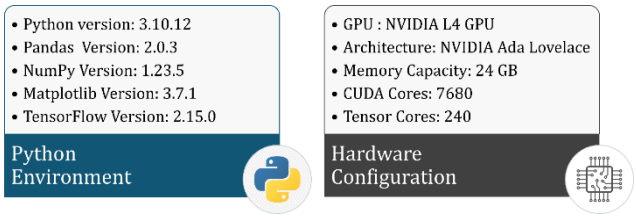



***Figure 7.*** *Experimental settings.*

## 5. Evaluation

As mentioned earlier, the target classes are 'true' and 'fake' news, making the proposed model a binary classification model. The efficacy of a binary classification model in any detection approach fundamentally depends on its evaluation metrics, as outlined by the confusion matrix. A confusion matrix offers a detailed snapshot of a classification algorithm's performance, presenting crucial comparative information. In this study, six widely recognized performance metrics have been derived from the detection model's confusion matrix, which is discussed below. Specifically, in the realm of fake news detection, the confusion matrix encompasses four key outcomes: True Positives, True Negatives, False Positives, and False Negatives – as illustrated in Figure 8.

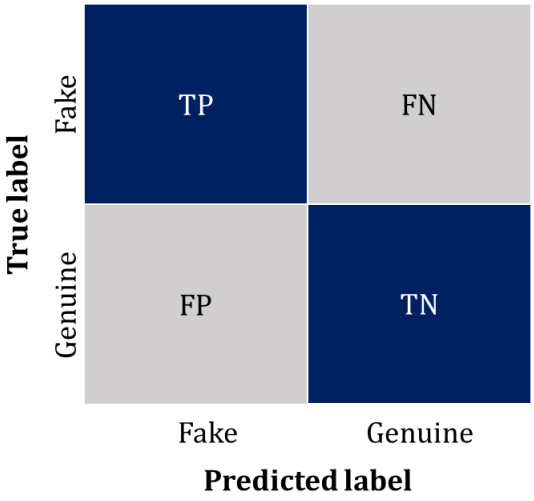



***Figure 8.*** *Confusion matrix of fake news detection model.*

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.

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.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{29}$$

Precision: Precision measures the proportion of correctly identified fake news tweets out of all tweets classified as fake.

$$Precision = \frac{TP}{TP+FP} \tag{30}$$

Recall: Recall measures the proportion of true positive cases out of all actual positive cases, reflecting the model's ability to identify all instances of fake news accurately. This metric is also referred to as Sensitivity, True Positive Rate, or Detection Rate.

$$Recall/TPR/Sensitivity = \frac{TP}{TP+FN} \tag{31}$$

F1-Score: The F1-Score is the harmonic means of precision and recall, offering a single metric that balances the trade-off between these two measures.

$$F1\ Score = \frac{2\times Precision\times Recall}{Precision+Recall} \tag{32}$$

False Positive Rate (FPR): False Positive Rate (FPR) measures the proportion of genuine news articles that are incorrectly classified as fake news by the model.

$$FPR = \frac{FP}{FP+TN} \tag{33}$$

Specificity: Specificity measures the proportion of genuine news tweets that the model accurately identifies as not being fake. It highlights the model's capability to correctly distinguish true news from fake news. This metric is also referred to as the True Negative Rate (TNR).

$$\text{Specificity/TNR} = \frac{TN}{TN+FP} \tag{34}$$

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.

$$\text{Error Rate} = \frac{FP+FN}{TP+TN+FP+FN} = 1 - Accuracy \tag{35}$$

Negative Precision: Negative Precision (or precision of the negative class) measures the proportion of instances correctly predicted as real out of all instances predicted as genuine

news.

$$Negative\ \ Precision\ \ = \frac{TN}{TN+FN} \quad (36)$$

G-mean1: G-mean1 is the traditional G-mean, which balances the recall (sensitivity) of the positive (fake) and negative (genuine) classes.

$$G-mean1 = \sqrt{Sensitivity * Specificity} \quad (37)$$

G-mean2: G-mean2 extends the concept by incorporating precision along with recall. This variation aims to provide a more comprehensive evaluation by considering the positive predictive value (precision) in addition to the true positive rate (recall).

$$G-mean2 = \sqrt{Precision * Recall} \quad (38)$$

Matthews Correlation Coefficient (MCC): Takes into account true and false positives and negatives and is generally regarded as a balanced measure even if the classes are of very different sizes.

$$MCC = \frac{(TP*TN-FP*FN)}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}} \quad (39)$$

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, F1-score, recall, precision, gmean1, gmean2, MCC 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.

ROC Curve: The Receiver Operating Characteristic (ROC) curve is a graphical representation used to evaluate the diagnostic performance of a binary classifier system by varying its discrimination threshold. The ROC curve is generated by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at different threshold levels. It is especially valuable for comparing the performance of multiple models or classifiers.

ROC-AUC Score: The ROC-AUC score, a value between 0 and 1, reflects how well the model can tell the difference between real and fake news tweet. A higher score indicates better discrimination ability.

Cohen Kappa Coefficient: Cohen's Kappa score is a measure of inter-rater agreement or classification performance that accounts for the possibility of agreement occurring by chance.

$$k = \frac{p_o - p_e}{1-p_o} = \frac{2*(TP*TN-FP*FN)}{(TP+FP)*(FP+TN)+(TP+FN)*(FN+TN)} \quad (40)$$

Where $p_o$ is the observed agreement, and $p_e$ is the expected agreement by chance.

## 6. Results and Discussion

The fake news detection model from tweets relies on the evaluation metrics scores. This section has been organized into five sub-sections for better understanding. Section 6.1 highlights the performance characteristics for fake news detection model, while Section 6.2 offers an ablation study that underscores the significance of the Bi-LSTM and Transformer components individually and their combined effect. Section 6.3 provides a comparative analysis of the overall results across individual deep-learning classifiers in the area. After that, 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. Lastly, Section 6.5 presents a statistical analysis across multiple models, offering a detailed examination of the performance variations and validating the robustness of the proposed system.

### 6.1. Classification Results of the Proposed Model

This section finds the results of the fake news detection model including different evaluation metrics scores which have been derived from confusion matrix. The confusion matrix of the proposed model incorporating transformers and Bi-LSTM has been displayed in Figure 9.

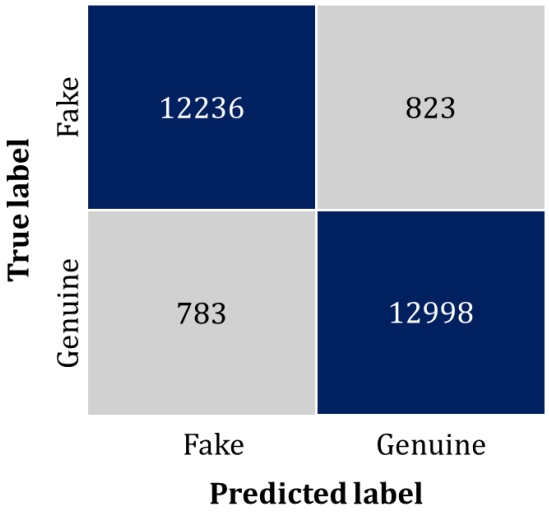



***Figure 9.*** *Confusion matrix for the proposed model.*

The performance of the proposed model from tweets has been quantitatively assessed using a set of evaluation metrics which have been previously discussed. Table 7 below summarizes these metrics and their respective values which have been derived from confusion matrix in Figure 9.

*Table 7. Experimental results for the fake news detection model.*

| Evaluation Metrics | Value |
|---|---|
| Accuracy | 94.02 % |
| Error Rate | 5.98 % |
| Recall | 93.70 % |
| Precision | 93.99 % |
| Negative Precision | 94.05% |
| F1-Score | 93.84 % |
| FPR | 5.68 % |
| Specificity | 94.32 % |
| G-mean1 | 94.01% |
| G-mean2 | 93.84% |
| MCC | 0.8802 |
| ROC-AUC | 0.9614 |

The experimental results underscore the robust performance of our fake news detection model across various metrics. Achieving an accuracy of 94.02% highlights the overall correctness of our predictions, complemented by a low error rate of 5.98%. Notably, our model demonstrates a high recall of 93.70%, effectively capturing the majority of actual positive cases (fake news), while maintaining a precise identification with a precision of 93.99%. Specificity, measuring the model's ability to correctly identify true negatives among all actual negatives, stands at 94.32%. This underscores its reliability in discerning genuine news tweets. Furthermore, the model exhibits a balanced F1-Score of 93.84%, emphasizing its consistency in achieving both high precision and recall. The false positive rate (FPR) of 5.68% indicates a minimal occurrence of false alarms, crucial for maintaining credibility in fake news detection. The high ROC-AUC score of 0.9614 further validates the model's exceptional ability to discriminate between fake and authentic tweets. Moreover, the model's negative predicted value, reflective of its capability to avoid false negatives and accurately identify true negatives, stands at 94.05%. This metric highlights the model's proficiency in distinguishing real news accurately. The geometric mean (G-mean), computed at 94.01% and 93.84% for G-mean1 and G-mean2 respectively, reinforces that the model maintains balance across both positive and negative classes without bias towards either. Finally, the Matthews correlation coefficient (MCC), measuring the overall correlation between predicted and observed classifications, is notably high at 0.8802. This metric indicates the model's strong predictive accuracy by accounting for true positives, true negatives, false positives, and false negatives.

In summary, these comprehensive metrics collectively affirm the effectiveness and reliability of our hybrid Transformer and Bi-LSTM architecture in detecting fake news in large-scale tweet datasets. The model not only excels in accuracy, precision, recall, and specificity but also demonstrates robust discrimination capabilities, as evidenced by the high ROC-AUC and MCC scores.

Table 8 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 8. Detection rate for each class.*

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

## 6.2. Ablation Study of Model Components

To illustrate the importance of each component in the hybrid model, we have performed ablation studies, which involved systematically removing or isolating major components to assess their individual contributions to the overall performance. The results of this study are summarized in Table 9. These studies highlight the significance of the Bi-LSTM and Transformer components individually and their combined effect.

*Table 9. Experimental results from ablation studies.*

| Evaluation Metrics | Bi-LSTM Only | Transformer Only | Concatenation of Bi-LSTM and Transformer |
|---|---|---|---|
| Accuracy | 89.65% | 92.17% | 94.02 % |
| Error Rate | 10.35% | 7.83% | 5.98 % |
| Recall | 86.37% | 91.49% | 93.70 % |
| Precision | 91.86% | 92.34% | 93.99 % |
| Negative Precision | 87.78% | 92.01% | 94.05% |
| F1-Score | 89.03% | 91.91% | 93.84 % |
| FPR | 7.25% | 7.19% | 5.68 % |
| Specificity | 92.75% | 92.81% | 94.32 % |
| G-mean1 | 89.50% | 92.14% | 94.01% |
| G-mean2 | 89.07% | 91.91% | 93.84% |

| Evaluation Metrics | Bi-LSTM Only | Transformer Only | Concatenation of Bi-LSTM and Transformer |
|---|---|---|---|
| MCC | 0.7938 | 0.8433 | 0.8802 |

The ablation study highlights the significance of each component in the proposed hybrid model. When using only the Bi-LSTM, the model achieved an accuracy of 89.65%, demonstrating its ability to capture sequential dependencies and temporal patterns but falling short in grasping long-range dependencies. The Transformer alone improved the accuracy to 92.17%, showcasing its strength in capturing long-range dependencies and global context through self-attention mechanisms, yet it struggled with sequential dependencies. Combining Bi-LSTM and Transformer outputs resulted in the highest accuracy of 94.02%, illustrating the complementary strengths of both architectures. This combined approach not only enhanced feature extraction, robustness, and generalization but also effectively handled complex patterns in fake news detection, achieving superior results across all metrics.

In summary, the ablation studies underscore the importance of both Bi-LSTM and Transformer components in the proposed hybrid model. Their combined effect significantly enhances the model's ability to detect fake news effectively, demonstrating the value of integrating these architectures for improved performance.

## 6.3. Comparative Analysis with Other Deep Learning Classifiers

The performance of the proposed model on a broader scale has been evaluated by reconstructing the entire workflow for several deep-learning classifiers. In this reconstruction, all preprocessing stages for the end-to-end workflow remain unchanged. However, instead of the proposed hybrid classification model, CNN and RNN-based classifiers have been employed to detect fake news. A comparative analysis has been conducted using the 'Truthseeker' dataset for fake news detection. The observational results for the model have been examined across a comprehensive set 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 10.

***Table 10.*** *Comparison against other deep learning classifiers and fine-tuned transformers with proposed model.*

| Evaluation Metrics | MLP | LSTM + Bi-LSTM | CNN + Bi-LSTM | Transformer + CNN | Transformer + LSTM | BERT | Proposed |
|---|---|---|---|---|---|---|---|
| Accuracy (%) | 89.02 | 91.89 | 90.75 | 93.11 | 93.51 | 93.99 | 94.02 |
| Error Rate (%) | 10.98 | 8.11 | 9.25 | 6.89 | 6.49 | 6.01 | 5.98 |
| Recall (%) | 91.06 | 90.86 | 88.57 | 91.88 | 92.59 | 93.69 | 93.70 |
| Precision (%) | 86.98 | 92.34 | 92.12 | 93.82 | 93.99 | 93.93 | 93.99 |
| Negative Precision (%) | 91.13 | 91.47 | 89.55 | 92.46 | 93.07 | 94.04 | 94.05 |
| F1-Score (%) | 88.97 | 91.59 | 90.31 | 92.84 | 93.28 | 93.81 | 93.84 |
| FPR (%) | 12.92 | 7.14 | 7.18 | 5.73 | 5.61 | 5.73 | 5.68 |
| Specificity (%) | 87.08 | 92.86 | 92.82 | 94.27 | 94.39 | 94.27 | 94.32 |
| G-mean1 (%) | 89.05 | 91.85 | 90.67 | 93.07 | 93.49 | 93.98 | 94.01 |
| G-mean2 (%) | 89.00 | 91.60 | 90.33 | 92.84 | 93.29 | 93.81 | 93.84 |
| MCC | 0.7813 | 0.8376 | 0.8153 | 0.8621 | 0.8702 | 0.8796 | 0.8802 |

From Table 10 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. It is clear that, incorporating deep learning layers including CNN, LSTM or Bi-LSTM with transformer architecture can significantly increase model performance where the proposed model outperforms others.

Although the proposed model outperforms all baseline models across most evaluation metrics, it is important to note that fine-tuned transformer models such as BERT also demonstrated competitive performance. For instance, BERT's F1-score and accuracy were close to those of the proposed

hybrid model. However, BERT incurred significantly higher computational costs in terms of training and inference time due to its large number of parameters. This high resource demand makes it less efficient for real-time or resource-constrained environments. In contrast, the proposed hybrid model provides a more efficient alternative by achieving comparable or superior performance with reduced training and testing time. While fine-tuned transformers like BERT, RoBERTa, or GPT variants may perform well on extremely large-scale datasets, the proposed model offers an optimal balance between accuracy and efficiency, making it particularly suitable for medium-sized datasets such as TruthSeeker.

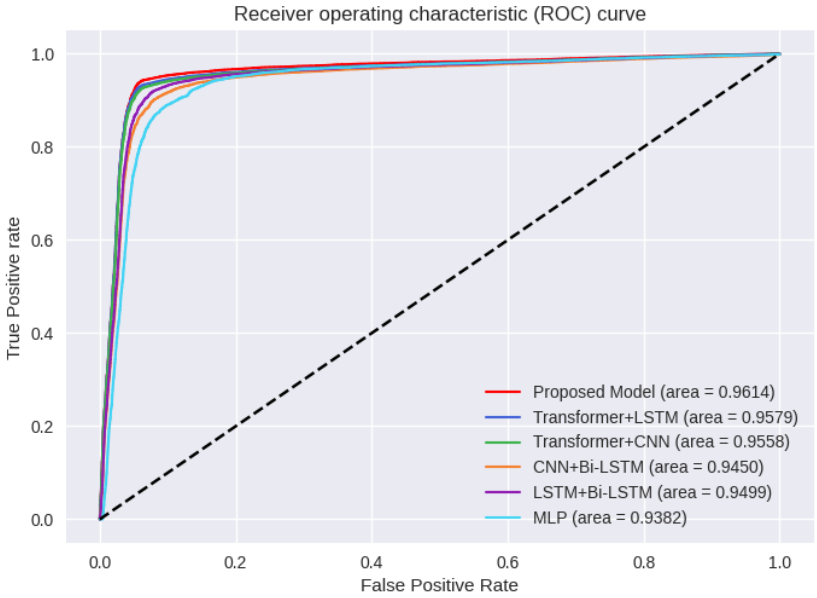



***Figure 10.*** *ROC curves for different classifiers.*

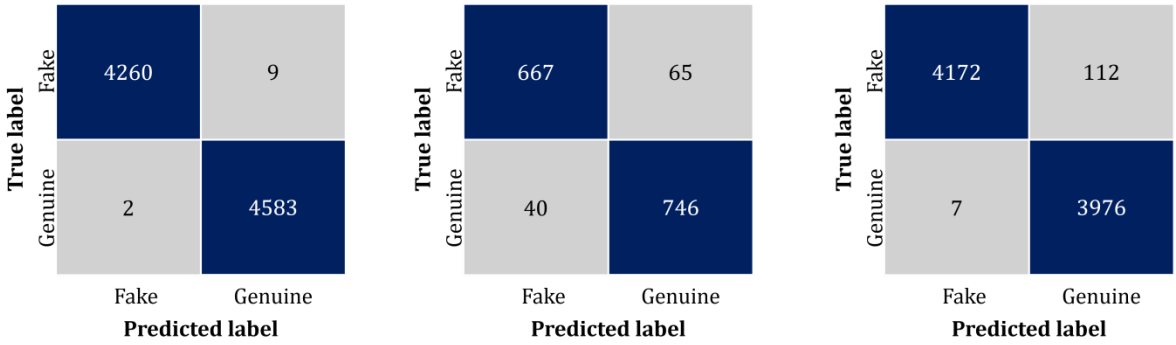
The ROC curve presented in Figure 10 above compares the performance of various fake news detection models. The proposed model, represented by the red line, exhibits the highest area under the curve (AUC) of 0.9614, indicating superior ability to distinguish between true and fake news. Other models, such 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. Comparative Evaluation of Model Performance Across Various 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 11, 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 Figure 11.

Table 11 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.




(a) ISOT Dataset (b) COVID19-FNIR Dataset (c) FND Dataset-English

*Figure 11. Confusion Matrices of the Proposed Model on Different Datasets.*

*Table 11. 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 [33] | Train- 36,539<br>Test- 8,857 | News, Politics | Website, Social Media | 2017 | 99.86 | 99.78 | 99.95 | 99.87 | 99.95 | 0.9999 |
| COVID19-FNIR [34] | Train- 6,070<br>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 [35] | Train- 36,000<br>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<br>Test- 26,840 | Politics, General Events, Health, Crime, Science | Twitter | 2022 | 94.02 | 93.07 | 93.99 | 93.84 | 94.82 | 0.9614 |

## 6.5. Statistical Analysis Across Multiple Models

To illustrate further model's effectiveness, a statistical measure, Cohen's Kappa score have been evaluated. Cohen's Kappa score (K) is a metric used to measure the dependability of agreement between and within raters for categorical data, accounting for the likelihood of chance agreement. This makes it a more precise measure than simple percentage agreement. Although it can be adapted for scenarios involving more than two raters, it is commonly applied in settings with two raters.

In the context of fake news detection, one "rater" is the classification model, while the other "rater" is a human expert who knows the true labels of news articles. Cohen's Kappa assesses how often the model and the expert agree (correctly identifying real and fake news) and how often they disagree (misclassifying

fake news as real and vice versa). By considering the possibility of agreement occurring by chance, Cohen's Kappa provides a measure of the overall agreement and the chance-corrected agreement between the model and the human expert. The following Table 12 provides the Cohen Kappa Scores for various fake news detection classifiers, demonstrating their performance in terms of agreement with a human expert, adjusted for chance.

***Table 12.*** *Comparing several models using Cohen's Kappa Coefficient.*

| Classifier Name | Cohen Kappa Score |
|---|---|
| MLP | 0.7804 |
| Bi-LSTM Only | 0.7924 |
| CNN + Bi-LSTM | 0.8147 |
| LSTM + Bi-LSTM | 0.8375 |
| Transformer Only | 0.8432 |
| Transformer + CNN | 0.8620 |
| Transformer + LSTM | 0.8701 |
| BERT | 0.8796 |
| Proposed Approach | 0.8802 |

Figure 12 represents the comparison of Cohen Kappa score of multiple classifiers. The MLP and Bi-LSTM classifiers achieve a Cohen Kappa score below 0.8, indicating a substantial level of agreement. The CNN + Bi-LSTM, LSTM + Bi- LSTM, Transformer only classifiers indicating moderate level of substantial agreement. When the transformer block is incorporated with another layer including CNN, LSTM or Bi-LSTM and BERT as classifiers achieve almost perfect agreement with greater than 0.85 of Cohen kappa co-efficient. The proposed approach, presumably a hybrid model that leverages the strengths of both Transformer and Bi-LSTM architectures, achieves the highest Cohen Kappa score of 0.8802. This indicates that it has the highest agreement with the human expert, adjusted for chance.

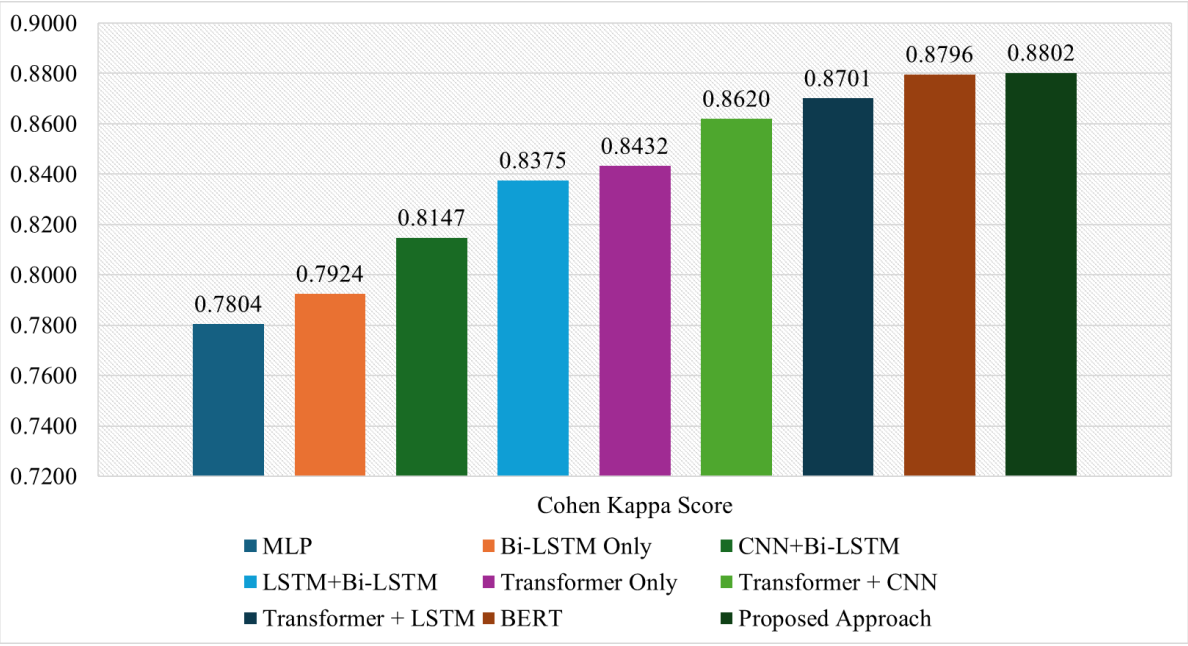



***Figure 12.*** *Comparison of Cohen Kappa Score of several models.*

## 7. Conclusion and Future Work

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.

While TweetGuard demonstrates strong performance on Twitter data, its generalizability to other social media platforms such as Facebook or Reddit may be limited due to platform-specific characteristics. For instance, Facebook posts often contain longer and more structured content, while Reddit discussions include threaded conversations and community-specific jargon. These variations in text style, post length, and discourse structure could affect the model's performance, as it was primarily trained on the linguistic and contextual patterns of tweets. Future research should explore domain adaptation techniques or multi-platform training strategies to improve cross-platform generalizability.

## Abbreviations

| | |
|---|---|
| ML | Machine Learning |
| DL | Deep Learning |
| LR | Logistic Regression |
| SVM | Support Vector Machine |
| DT | Decision Tree |
| RF | Random Forest |
| KNN | K-Nearest Neighbor |
| CNN | Convolution Neural Network |
| MLP | Multi-Layer Perception |
| LSTM | Long Short-Term Memory |
| GRU | Gated Recurrent Unit |
| DNN | Deep Neural Network |
| RNN | Recurrent Neural Network |
| BERT | Bidirectional Encoder Representations from Transformers |
| GPT | Generative Pre-training Transformer |
| NLP | Natural Language Processing |
| LLM | Large Language Model |

## Author Contributions

**Kowshik Sankar Roy:** Conceptualization, Data curation, Formal Analysis, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing

**Farhana Akter Bina:** Data curation, Funding acquisition, Investigation, Project administration, Resources, Supervision

## Conflicts of Interest

The authors declare no conflicts of interest.

## References

[1] 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

[2] 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

[3] 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/

[4] 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

[5] 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/

[6] Orbanek, S. (2021, November 10). Study shows verified users are among biggest culprits when it comes to sharing fake news. Temple Now | news.temple.edu. https://news.temple.edu/news/2021-11-09/study-shows-verified-users-are-among-biggest-culprits-when-it-comes-sharing-fake

[7] 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

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

[9] Udas, P. B., Karim, M. E., & Roy, K. S. (2022). SPIDER: A shallow PCA based network intrusion detection system with enhanced recurrent neural networks. Journal of King Saud University. Computer and Information Sciences, 34(10), 10246–10272. https://doi.org/10.1016/j.jksuci.2022.10.019

[10] Udas, P. B., Roy, K. S., Karim, M. E., & Ullah, S. M. A. (2023). Attention-based RNN architecture for detecting multi-step cyber-attack using PSO metaheuristic. https://doi.org/10.1109/ecce57851.2023.10101590

[11] 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

[12] 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

[13] 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

[14] Roy, K. S., Udas, P. B., Alam, B., & Paul, K. (2025). Unveiling hidden patterns: A deep learning framework utilizing PCA for fraudulent scheme detection in supply chain analytics. *International Journal of Intelligent Systems and Applications*, *17*(2), 14–30. https://doi.org/10.5815/ijisa.2025.02.02

[15] Roy, K. S., & Islam, S. M. R. (2023). An RNN-based Hybrid Model for Classification of Electrooculogram Signal for HCI. Computing, 335–344. https://doi.org/10.47839/ijc.22.3.3228

[16] 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

[17] Roy, K. S., Ahmed, T., Udas, P. B., Karim, M. E., & Majumdar, S. (2023). MalHyStack: A hybrid stacked ensemble learning framework with feature engineering schemes for obfuscated malware analysis. Intelligent Systems With Applications, 20, 200283. https://doi.org/10.1016/j.iswa.2023.200283

[18] Baarir, N. F., & Djeffal, A. (2021). Fake News detection Using Machine Learning. https://doi.org/10.1109/ihsh51661.2021.9378748

[19] Abdulrahman, A., & Baykara, M. (2020). Fake News Detection Using Machine Learning and Deep Learning Algorithms. https://doi.org/10.1109/icoase51841.2020.9436605

[20] 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

[21] 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

[22] 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

[23] 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

[24] 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

[25] 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

[26] 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

[27] 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

[28] 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

[29] Dhiman, P., Kaur, A., Gupta, D., Juneja, S., Nauman, A., & Muhammad, G. (2024). GBERT: a hybrid deep learning model based on GPT-BERT for fake news detection. Heliyon, 10(16), e35865. https://doi.org/10.1016/j.heliyon.2024.e35865

[30] Vallileka, N., Sundaravadivel, P., Karthikeyan, U., Krishnan, R. S., Narayanan, K. L., & Sundararajan, S. (2023). DeepTweet: Leveraging transformer-based models for enhanced fake news detection in twitter sentiment analysis. 2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). https://doi.org/10.1109/I-SMAC58438.2023.10290217

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

[32] Roy, K. S., Karim, M. E., & Udas, P. B. (2022). Exploiting Deep Learning Based Classification Model for Detecting Fraudulent Schemes over Ethereum Blockchain. https://doi.org/10.1109/sti56238.2022.10103259

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

[34] 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

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