Received June 1, 2025, August 1, 2025, date of publication September 1, 2025.

Digital Object Identifier 10.21608/ijaici.2025.358840.1009

A Survey of Deepfake Text Detection and Security Attack Prediction via Sentiment Analysis

Norhan A. Farouk[12], Sara Sweidan [2], Mohamed Taha [3]

[1]Artificial Intelligence Department, Faculty of Computers and Artificial Intelligence, Benha University, Egypt.
[2]Future Academy - Higher Future Institute for Specialized Technological Studies, 29 Ismailia Desert Rd, El Shorouk - Cairo 6363040, Egypt.
[3]Department of Computer Science, Faculty of Computers and Artificial Intelligence, Benha University, Egypt.

Corresponding author: Norhan A. Farouk (e-mail: norhan.farouk@fa-hists.edu.eg).

ABSTRACT This survey examines the emerging intersection of deepfake text detection, security attack prediction, and sentiment analysis on social media platforms. We analysis recent advances in machine learning approaches for identifying machine-generated content, particularly focusing on short-form text like tweets. The survey covers state-of-the-art models including transformer-based architectures, attention mechanisms, and specialized frameworks like DeBERTa. We also explore how sentiment analysis can serve as an early warning system for potential security threats.

INDEX TERMS Sentiment Analysis, Security Attack, Twitter, Deepfake, Machine-generated Text, Classification.

I. INTRODUCTION

A. MOTIVATION

The proliferation of machine-generated text on social media platforms presents growing challenges for information integrity and security. Statistics indicate that Twitter alone processes over 9,000 tweets per second, making it a critical battleground for both legitimate communication and potential manipulation. During the 2016 US Presidential Election, an estimated 19 million bot accounts posted election-related content, demonstrating the scale of automated influence operations.

B. RESEARCH OBJECTIVES

This survey aims to:

- Analyses current approaches to detecting machine-generated text on social media: In today's fast-changing digital environment, creating fake visual, aural, and textual content gravely jeopardizes public confidence, political stability, and information integrity [1][2]. Fake news has become a serious issue for societies and a significant task for those combating disinformation [3][4].
- Evaluate the effectiveness of sentiment analysis in predicting security attacks: This paper surveys texts produced using deepfake technology and investigates their impact on society, politics, the economy, and technology. This study also demonstrates how shifts in user sentiment can be strong indicators of imminent security attacks by leveraging sentiment analysis. The results of our proposed framework, which combines DeBERTa model architecture and sentiment analysis, show efficient and effective classification of fake content on social media, achieving a superior accuracy of 97% in identifying malicious intent and potential security threats.
- Compare performance metrics across different detection methodologies: The performance of the proposed method is also compared against other deep learning models such as Gradient Boosting Classifier, Random Forest, Logistic Regression, Decision Tree, Naive Bayes Classifier, Random Forest, SVM, k-means clustering, Logistic Regression, BERT, CNN, RNN, Semi-supervised learning, SPBERT and RoBERTa model. And also used Bag of Words (BoW), and TF-IDF features for text data transformation. This is for displaying the effectiveness and highlighting the advantages of accurately addressing the task at hand with the proposed model. Experimental results indicate that

the design of the DeBERT architecture, coupled with the utilization of the preprocessing data techniques, allows for efficient and effective classification of the tweet and decides if generated by a human account or a bot account.

- Identify key challenges and future research directions: Artificial intelligence techniques known as 'deepfakes' have improved and made this production process more accessible [5]. While visual and audio deepfakes have received significant attention, text-based deepfakes, which are increasingly employed to alter sentiment and elicit emotional reactions, deserve greater investigation these altered sentiments in online conversations can serve as catalysts for security attacks, spreading misinformation to amplify divisive or destabilizing narratives.

II. Background and Functions

A. EVOLUTION OF TEXT GENERATION MODELS

The development of text generation capabilities has progressed from simple rule-based systems to sophisticated transformer-based models. Key milestones include:

Early approaches using RNNs and LSTMs: The development of text generation models has hugely progressed in the development from early approaches utilizing Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks [6]. These models have put the foundation for more advanced techniques in natural language processing (NLP) [7][8]. Initially, RNNs were used for their capacity to handle sequential data and capture temporal relationships in text [9] [10]. However, they had difficulty with long-range relationships due to vanishing gradient concerns. The invention of LSTMs addressed these constraints by integrating memory cells that can retain information across longer sequences, thus improving text production quality [11] [12]. Singh et al. [13] discuss the evolution of text generation models, highlighting early approaches using Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTMs), which laid the groundwork for more advanced techniques by addressing issues of sequence prediction and context retention. and this paper aims to examine deep learning methods for text generation in NLP and Assess the strengths, limitations, and performance of various models. Jiancong Zhu [14] discusses Early text generation models utilizing Recurrent Neural Networks (RNNs) for sequential memory, processing sequences element by element. Long Short-Term Memory (LSTM) networks advanced this by addressing the vanishing gradient problem, enhancing the ability to remember long-term dependencies in the generated text. and the objectives of this study are to dissect underlying mechanisms of Seq2Seq models in NLP and Provide insights into applications across real-world scenarios. Research indicates that increasing the amount of LSTM cells can enhance predictive performance, but excessive complexity can lead to overfitting [11]. Additionally, optimizing vocabulary size is critical for effective text production, since a smaller, well-defined vocabulary can produce better results [11]. While RNNs and LSTMs have long been used in text creation, the industry has shifted toward transformer-based models, which provide better performance and efficiency when dealing with complicated language tasks. This shift emphasizes the continuous progress and refining of text generation strategies in NLP [18].

Introduction of the transformer architecture: The introduction of the transformer architecture heralded a substantial shift in text generation models, profoundly altering natural language processing (NLP) and generative AI. This architecture, noted for its capacity to analyse data sequences concurrently, has facilitated the creation of robust models such as BERT and GPT, which excel at producing human-like writing and comprehending context. these are Key Features of Transformer Architecture:

- **Parallel Processing:** In contrast to RNNs and LSTMs, transformers concurrently process all tokens in a sequence, hence improving efficiency and performance [15].
- **Attention Mechanism:** The self-attention mechanism allows models to weigh the value of distinct words in a context, enhancing contextual comprehension [16].
- **Scalability:** Transformers may be extensively scaled up, allowing the development of large language models (LLMs) capable of handling massive amounts of data [17].

Han Xu et al. [19] discuss the evolution of text generation models, highlighting the introduction of the Transformer architecture. This architecture revolutionized NLP by enabling parallel processing and improving context understanding, leading to the development of influential models like BERT and GPT. The objective of this study was to provide a comprehensive guide to modern language model architectures and explore future directions in AI research and applications. Raghuraj Singh [20] discusses the traces of the evolution of language transformer models, highlighting the introduction of the transformer architecture, which utilizes self-attention mechanisms and positional encoding, significantly enhancing text generation capabilities compared to previous models, and thereby transforming natural language processing. This study aims to explore the history and impact of language transformer models and analyse the architectures, strengths, and weaknesses of major models.

Development of pre-trained models like BERT and GPT: The evolution of text generation models, particularly with the advent of pre-trained models like BERT and GPT, has significantly transformed natural language processing (NLP). These models leverage advanced architectures to enhance text coherence, contextual understanding, and generative capabilities [21]. The integration of BERT's

bidirectional context with GPT's autoregressive structure has led to hybrid models that outperform traditional approaches in various NLP tasks [22] [23]. Key Developments in BERT and GPT:

- **BERT (Bidirectional Encoder Representations from Transformers):**
  - Excels in tasks requiring deep contextual understanding, such as sentiment analysis and named entity recognition [23].
  - Utilizes a masked language modelling approach, allowing it to predict missing words based on the surrounding context [24].
- **GPT (Generative Pre-trained Transformer):**
  - Focuses on text generation, making it suitable for creative writing and conversational AI [23].
  - Employs an autoregressive model, generating text sequentially, which enhances fluency and coherence [25].
- **Hybrid Models and Their Impact:**
  - Recent research has introduced hybrid models that combine BERT and GPT capabilities, such as the BERT-GPT-4 model, which achieves superior performance in generating coherent and contextually accurate text [26].
  - These models demonstrate improved metrics like Perplexity and BLEU, indicating advancements in natural language generation [26].

Minghua Zhang [27] focuses on pre-trained models, particularly BERT, highlighting its significant impact on NLP tasks and the subsequent development of various BERT-based models. It emphasizes the importance of transfer learning in enhancing text generation capabilities within the field. The primary objective of this research was to classify NLP-PTMs based on BERT variants and analyse research methods and experimental results. Rohit Pandey et al [28]. focus on generative AI-based text sgeneration methods using the pre-trained GPT-2 model, highlighting its ability to generate contextually relevant text by learning linguistic patterns from extensive corpora, which reflects the evolution of text generation models. The main objectives of the study are to train models to generate contextually relevant text and learn linguistic patterns from extensive corpora for text generation. Salıcı et al. [23] discuss the evolution of text generation models, highlighting BERT's bidirectional context evaluation for tasks like sentiment analysis and GPT's autoregressive structure excelling in text production, emphasizing their revolutionary impact on NLP and the need for further development in low-resource languages. This study aims to examine the effects of BERT and GPT in NLP and discuss challenges in low-resource languages like Turkish. FB AlShannaq et al. [25] explores the evolution of Generative Pre-trained Transformers (GPT) models, comparing them with other AI language models like BERT, highlighting their development, capabilities in generating human-like text, and implications for natural language understanding and conversational AI. The objectives of this study are designed to explore the development and capabilities of GPT models and examine the limitations and ethical concerns of GPT usage. Lucas Georges & David Samuel [24] discuss the evolution of text generation models by merging masked language modelling (MLM) and causal language modeling (CLM) into a hybrid model, GPT-BERT, which outperforms traditional models and showcases improved capabilities even with limited data. This study aims to merge masked and causal language modeling techniques. and evaluate hybrid model performance on BabyLM Challenge 2024.

Table 1: Literature review

Emergence of specialized models for social media content: The growth of text generation models has resulted in the emergence of specialized models designed specifically for social media content, reflecting the platform's particular traits and expectations. These models use complex architectures and training techniques to generate text that is interesting and contextually relevant for users [29]. Wang Ziwen et al. [30] discuss the development of a conditional language generation model that incorporates Big Five Personality feature vectors, enabling the generation of personalized, human-like short texts for social media, addressing the limitations of traditional statistical language models. Yuting Guo et al. [31] The emergence of specialized models like SocBERT highlights the evolution of text generation models tailored to social media. By pretraining on diverse platforms like Twitter and Reddit, these models enhance performance on social media-specific NLP tasks compared to general models. This study aims to develop a language model for social media text and benchmark against existing transformer-based models. Rong Ma et al. [32] discuss the development of specialized text generation models, particularly focusing on improving social media content quality through backtracking patterns, enhancing coherence and readability, and addressing issues like cyclic repetition and incoherence in generated texts. The objective of this study is to address improved text generation quality using backtracking patterns and enhance the coherence and readability of generated social text.

Paper	Challenges	Methods Used	Results
[13]	Cohesion, fluidity, and semantic coherence in text generation. Limitations of various deep learning models assessed.	Recurrent Neural Networks (RNNs) Convolutional Neural Networks (CNNs) Transformers	Assesses strengths and limitations of deep learning models. Identifies trends and areas for further investigation.
[14]	Vanishing and exploding gradient problems in RNNs. Complexity of language and dynamic sequence variations.	Examination of RNNs, LSTMs, and Transformers. Comparative analysis of sequence-to-sequence models.	Highlights strengths and limitations of sequence-to-sequence models. Discusses advancements from RNNs to BERT in NLP.
[30]	Machine-like generated texts lack anthropomorphic characteristics. Need for individualized text generation based on personality traits.	Conditional language generation model with BFP features vectors. Long short memory network (LSTM) and convolution neural network (CNN).	Generated Chinese short texts exhibit discriminative personality styles. Texts are syntactically correct and semantically smooth with appropriate emoticons.
[23]	Structural features of Turkish complicate model application. Limitations of datasets hinder performance in low-resource languages.	Discussion of BERT and GPT model structures. Analysis of Turkish-specific models and data augmentation techniques.	BERT excels in text classification and sentiment analysis. GPT is superior in text production and creative writing.
[27]	-	Classification of research objects and methods. Experimental analysis of NLP-PTMs.	Investigate NLP-PTMs motivated by BERT. Suggests future directions for PTM development.
[31]	Influence of training time on PLMs' performance. Limited budget affecting GPU usage during training.	Developed masked language model (MLM) for pretraining. Classification model for benchmarking social media text tasks.	SocBERT outperformed BERT on most classification tasks. SocBERT-base and SocBERT-final performed similarly across tasks.
[32]	Cyclic repetition in generated text. Incoherence in text generated by pre-trained models.	Backtracking pattern for phrase joint sentences. Quality evaluator based on BERT for sentence level.	Improved text coherence and readability close to human writing. Enhanced generation quality without affecting model training.
[24]	Small model sizes and limited training datasets. Results may not scale outside BabyLM constraints.	Hybrid training of masked and causal language modeling. Evaluation on BabyLM Challenge 2024 benchmarks.	Hybrid pre-training outperforms masked-only and causal-only models. The best results were achieved with lower causal-to-masked ratios.
[28]	-	Generative AI-based text generation Utilizes pre-trained GPT-2 model	Trained models generate contextually relevant text based on linguistic patterns. Models learn grammar, vocabulary, phrases, and styles from extensive data.

B. SENTIMENT ANALYSIS FOR SECURITY

Sentiment analysis has emerged as a valuable tool for security applications by:

Detecting extreme negative sentiment as potential attack indicators: Sentiment analysis has become a critical tool in security applications, notably for detecting severe negative sentiments that may suggest a threat. Organizations can improve their threat detection capabilities by using advanced natural language processing (NLP) tools to spot suspicious behaviors and sentiments across several platforms:

- **Detection of Suspicious Activities:**
  Social Media Monitoring: NLP techniques, such as sentiment analysis, are used to examine social media posts for symptoms of cyberbullying, hate speech, and disinformation, with models reaching high accuracy in detecting suspect content [33]. Gagandeep et al. [33] demonstrate that sentiment analysis, particularly using LSTM networks, effectively identifies unfavorable sentiments in social media posts, flagging potential indicators of suspicious activities like cyberbullying and hate speech, thus enhancing security measures against such threats. their objectives are to design an automated system for detecting suspicious activities and identify unfavorable sentiments like cyberbullying and hate speech. they used Natural language processing for sentiment analysis and a Long short-term memory (LSTM) network for the model, and the results were Improved accuracy, precision, recall, and F1 score ratios and enhanced early detection of suspicious social media behavior.
  Hacker Forums Analysis:
  Deep learning models applied to hacker conversations can anticipate upcoming cyberattacks by recognizing hostile intent, with training accuracies as high as 99.93% [34]. Bulcha Mardassa et al. [34] Discuss sentiment analysis can detect extremely negative sentiments in hacker communications, serving as potential indicators of malicious intent. By analyzing these sentiments, organizations can better understand hacker motivations and prepare for potential future cyberattacks, enhancing their security measures. Their goals were to analysis hacker communications to understand motivations and tactics, and predict future attacks and improve security measures. The methods used in this research paper were deep Neural Network (DNN) with LSTM architecture and text preprocessing techniques like tokenization, stemming, and word embedding and the results were LSTM model achieved 99.93% training accuracy and LSTM model achieved 97.48% validation accuracy.
- **Insider Threat Detection:**
  Sentiment Profiles: Aspect-based sentiment analysis generates extensive profiles of employees, enabling the early detection of malevolent insiders based on their emotional characteristics [35]. sMeichen Liu et al. [35] present a unique insider threat detection approach that uses aspect-based sentiment analysis to identify employees with extremely malevolent emotional inclinations, allowing for the early detection of possible threats. By adding sentiment profiles into current systems, this strategy improves the ability to detect anomalies that indicate security vulnerabilities. To evaluate user behavior and detect malicious intent, the system uses fine-grained aspect-based sentiment analysis and advanced deep learning approaches, such as attention mechanisms. The results show that the framework is effective at detecting potential insider threats early on and enhancing the accuracy of existing anomaly detection systems by including sentiment-based user profiles.
- **Cybersecurity for Critical Infrastructure:**
  Emotional Sentiment Monitoring: Sentiment analysis algorithms can evaluate public attitudes regarding key infrastructure, offering early warning of potential cyber risks [36]. Svitlana Lehominova et al. [36] investigated the role of sentiment analysis in detecting extremely negative sentiments toward vital information infrastructure items, which could serve as early indicators of potential cyber-attack intents. Their goals included improving cybersecurity for vital information infrastructure and using sentiment analysis to detect cyber-attacks early. To better identify emotional attitudes related to cyber risks, researchers used an artificial neural network-based sentiment analysis model that included emoticons. The suggested model had an accuracy of 0.7852, indicating its usefulness in recognizing attitudes associated with potential cyber risks.

While sentiment analysis has substantial advantages in danger identification, it is critical to understand that it must be supplemented with human experience and ethical concerns in order to achieve responsible and accurate judgments [37].

Identifying coordinated manipulation campaigns: Sentiment analysis has become an important method for detecting coordinated manipulation campaigns, notably in the context of cybersecurity and social media monitoring. By analyzing emotional sentiments expressed in online

communications, security applications can identify possible threats and coordinate attacks on key information infrastructure. This capability is boosted by advanced machine learning and deep learning techniques, which enable the processing of massive volumes of data to detect patterns indicative of manipulation. if we talking about applications in Cybersecurity, we can talk about the "**Detection of Threats**" where Svitlana Lehominova et al. [36] discuss using sentiment analysis to detect emotional sentiments towards critical information infrastructure, which can help identify coordinated manipulation campaigns. and they aimed to ensure cyber security for critical information infrastructure and apply sentiment analysis for early detection of cyber threats. so, they used a Sentiment analysis model based on an artificial neural network and Consideration of emoticons to detect emotional attitudes toward cyber-attacks. The result was to create a sentiment analysis model, trained on social media data, for cyber threat detection with an accuracy: of 0.7852 aiding early detection of cyber threats. Also, Svitlana Lehominova et al. [36] discuss "**Data Sources**" Social media platforms like Twitter and Instagram serve as rich datasets for training sentiment analysis models, enabling early threat detection. Farheen Khanum et al. [38] discuss the "**Models Accuracies**" where they don't specifically address the use of sentiment analysis for identifying coordinated manipulation campaigns in security applications and they focus on the performance of LSTM networks in sentiment analysis using Twitter data and various machine-learning models. so, their objectives were evaluating LSTM networks for sentiment analysis on Twitter data and improve accuracy, precision, and recall in sentiment analysis tasks. The result Neural networks have shown promising accuracy rates (up to 84%) in classifying sentiments, which is crucial for timely responses to threats. They used LSTM networks for sentiment analysis on Twitter data and Pre-processing techniques: noise removal, tokenization, stop words removal, Feature extraction using GloVe embedding technique. and machine learning models like KNN, SVM, decision tree, random forest. On the occasion of mentioning the accuracy of the models, we must clarify the techniques and methodologies used in cybersecurity and sentiment analysis firstly talking about "**Machine Learning Approaches**" and also called "**Classical Approaches**". These techniques such as Support Vector Machines (SVM) and Long Short-Term Memory (LSTM) networks are employed to enhance sentiment classification accuracy [38] [39]. Followed by "**Deep Learning Frameworks**" where Novel deep learning methods provide efficient processing of large datasets, improving the identification of coordinated campaigns [40].

Monitoring shifts in public opinion that may precede security events: Sentiment analysis has become an important method for monitoring public opinion, specifically in security applications. Through advanced machine learning and natural language processing techniques, Sentiment analysis can detect fluctuations in public sentiment that may occur before security occurrences. This competence enables prompt interventions and informed decision-making. We must discuss "**Real-Time Monitoring of Public Sentiment**" if we wish to discuss "**Monitoring shifts in public opinion**":

- **Automated Analysis:** Traditional techniques for measuring public opinion are frequently labor-intensive and slow. Sentiment analysis, on the other hand, automates the procedure and makes it possible to monitor big datasets from social media and other platforms in real time [41]. Where Lingfeng Yu [41] discusses using sentiment analysis to monitor public opinion, enabling real-time identification of emotional trends. This capability can potentially highlight shifts in sentiment that may precede security events, enhancing proactive measures in security applications. This approach aligns with objectives such as analysing public opinion through text sentiment analysis and identifying emotional tendencies using algorithmic models. However, traditional methods face challenges, like traditional methods are time-consuming and labour-intensive and have difficulty in reflecting real-time sentiment changes and trends. To address these issues, they used machine learning and natural language processing technologies (NLP), and algorithmic models for text sentiment analysis. As a result, public opinion analysis through algorithmic models was achieved and real-time sentiment trends were identified using text data.
- **Deep Learning Models:**
  Real-time public mood monitoring has greatly improved thanks to deep learning algorithms, especially regarding social media data processing. Different approaches use natural language processing (NLP) and neural networks to capture and analyse the fluctuations of public opinion efficiently. Key methods and conclusions from recent studies are outlined in the sections that follow:
  Time-Series Neural Networks: A method utilizing time-series neural networks constructs a commercial map of social hotspots, capturing event dynamics and public opinion evolution through atomic event graph building and unsupervised learning. As Hangyin Mao et al. [42] proposed a method using time-series neural networks for public opinion analysis, capturing event dynamics and sentiment evolution. It employs unsupervised learning for deep event connections, enhancing real-time monitoring of public sentiment during social hot events. This approach aims to construct a commercial map of social hotspots and enhance public opinion monitoring through event analysis.

The results show this approach reduces storage costs and simplifies prediction and exhibits robustness in public opinion monitoring.

BiGura Model for Sentiment Detection: The BiGura model has shown high accuracy in real-time sentiment analysis, where Han Xu [43] proposes a multilayer BiGura modal-based technique for real-time sentiment detection, achieving 92.7% accuracy for text emotions and 86.9% for video emotions, significantly outperforming traditional classifiers like Bayesian and KNN in public sentiment monitoring. This model effectively monitors emotional tendencies and public opinion changes during significant events, enhancing social stability. He used "**Multilayer BiGura Modal-Based Technique**" and deep learning for sentiment analysis in new media. He faced challenges, like traditional machine learning techniques limited by large data volumes and need for real-time sentiment analysis in new media.

Deep Learning Models for Crisis Management: Utilizing LSTM and BiLSTM models, researchers analyse tweet conversations to gauge public sentiment during crises. As Sulochana et al. [44] employed deep learning techniques, specifically LSTM and BiLSTM models, alongside machine learning methods like logistic regression and SVM, to analyse tweet conversations, enabling real-time monitoring of public sentiment during crises and facilitating timely organizational responses. This methodology emphasizes the importance of rapid response strategies based on real-time sentiment analysis. This research resulted in improved organizational response strategies based on tweet analysis and appear automated tools for analysing public sentiment during crises.

BERT for Text Data Analysis: The BERT model has been highlighted for its superior accuracy in sentiment analysis compared to traditional models, enabling real-time processing of diverse data sources. So, Kumar et al. [45] discussed a real-time sentiment analysis system utilizing the BERT model for monitoring public sentiment. They emphasized data collection, preprocessing, and sentiment analysis to identify trends and patterns in text data from sources like social media and news websites. The aim of this research was Real-time sentiment analysis of text data from various sources and utilize BERT model for accurate sentiment analysis performance where this model's effectiveness underscores the potential of transformer-based architectures in understanding public opinion. They used BERT model for sentiment analysis and used real-time processing for sentiment trends and patterns. They faced some of challenges like understanding public opinion and decision-making process in real time and developing real-time sentiment analysis system using BERT model. But they were able to overcome these challenges and show BERT model shows high accuracy compared to traditional models and results demonstrate accuracy of the sentiment analysis system.

Advanced Techniques in Social Media Sentiment Analysis: A comprehensive framework employing advanced deep learning techniques, including BiLSTMs and innovative feature extraction methods, has been developed. This approach enhances data quality and model robustness, setting benchmarks for future sentiment analysis applications. Nguyen [46] discusses advanced deep learning techniques, particularly BiLSTM models, for real-time sentiment analysis on social media. These methods enhance accuracy and efficiency, enabling effective monitoring of public sentiment dynamics through sophisticated data preprocessing and feature extraction techniques. However, he faced challenges, including being diverse and evolving language use on social media and practical challenges in sentiment analysis implementation. To address these issues, he used advanced deep learning techniques for sentiment analysis on social media and data preprocessing, feature extraction, model optimization, and experimental analysis. As a result, outperformed existing methods by more than 3 in performance and established benchmarks in sentiment analysis field.

While deep learning techniques have revolutionized sentiment monitoring, challenges remain in ensuring data privacy and managing misinformation on social media platforms. Addressing these issues is crucial for the responsible application of these technologies in public sentiment analysis.

III. Current Approaches and Methodologies

A. DEEP LEARNING ARCHITECTURES

Recent research has focused on several key architectural approaches:

A.1 TRANSFORMER-BASED MODELS:

BERT and variants: Natural language processing (NLP) and other fields have seen substantial change as a result of transformer-based models, especially BERT and its variations. These models perform better on a variety of tasks,

including text summarization, sentiment analysis, and molecular representation, by utilizing the self-attention mechanism to grasp contextual linkages in data. **BERT (Bidirectional Encoder Representations from Transformers)** Utilizes bidirectional context to enhance understanding of language and excels in understanding contextual meaning for tasks like text classification and sentiment analysis. Salıcı et al. [23] examined the effects of BERT and GPT in NLP and discussed challenges in low-resource languages like Turkish. It turns out that it handles variants like BERTurk to address challenges in low-resource languages, particularly Turkish, enhancing performance through language-specific adaptations and data augmentation techniques. The findings demonstrate the complementary roles that BERT and GPT play in natural language processing (NLP), with BERT's strengths in text categorization and sentiment analysis and GPT's superiority in text generation and creative writing. With an emphasis on particular applications like text summarization and proteomics data processing, respectively, **variant models** like **PEGASUS** and **DIA-BERT** have surfaced. Xiao Fu [47] highlights the effectiveness of these models, leveraging an encoder-decoder framework and pre-training techniques to enhance summary accuracy and fluency. By utilizing dynamic weight allocation and parallel computation, these models have achieved superior performance on summarization benchmarks. The research objectives focus on reviewing advancements in Transformer models for text summarization and analysing their effectiveness and diverse applications. Methods such as graph techniques for topic and sentence identification, latent semantic analysis for semantic representation, Bayesian topic models for probabilistic topic representation, and **ROUGE metrics** for summary quality assessment underline the multifaceted approach taken in this domain. However, challenges persist, particularly regarding insufficient evaluation methods for semantic correctness and credibility, underscoring the need for comprehensive evaluation systems. Despite these challenges, the study highlights the effectiveness of models like **PEGASUS**, **BERT**, and **HETFORMER** in advancing text summarization methodologies. Liu et al. [48] presented the introduction of **DIA-BERT** for DIA proteomics data analysis. where DIA-BERT is a transformer-based model specifically designed for analysing data-independent acquisition mass spectrometry (**DIA-MS**) data, enhancing protein identification and quantification in proteomics. It utilizes pre-trained AI models to improve accuracy and efficiency in data analysis. They used a Transformer-based pre-trained AI model for analysis and utilized high-quality peptide precursors for training. As a result, **DIA-BERT** achieved 54% more protein identifications than **DIA-NN** and It identified 37% more peptide precursors with high accuracy. Transformer models, like BERT and its variations, have proven very adaptable in a variety of industries. They have achieved remarkable accuracy and efficiency in natural language processing (NLP), revolutionizing activities like machine translation and answering questions [23]. Transformer models are being used more and more in cheminformatics to describe chemical SMILES, which helps with reaction forecasting and molecular property prediction [49]. These models also have a lot of potential for use in neuroimaging, where they improve data analysis by doing exceptionally well on regression and classification tasks [50]. Despite their advantages, challenges remain, particularly in adapting these models to low-resource languages and specialized domains. Future research is essential to address these limitations and expand their applicability.

GPT family of models: By improving tasks like text generation, summarization, and comprehension, transformer-based models in particular, the GPT family have profoundly changed natural language processing (NLP). These models demonstrate their adaptability across a range of applications by processing and producing text that is human-like by utilizing deep learning architectures. Important features of GPT models and how they affect NLP are described in the sections that follow. GPT models utilize an autoregressive architecture, predicting the next word in a sequence based on previous words, which excels in text generation tasks [23] where they are pre-trained on vast datasets, allowing them to capture intricate language patterns and contextual nuances, making them effective for conversational AI and content creation [25]. The ability of GPT models to generate text that is coherent and contextually relevant has been demonstrated by their effective application in a variety of NLP applications, such as chatbots, automated content generation, and creative writing [25]. Furthermore, transformer models such as GPT are excellent at dynamically allocating attention weights and comprehending the global context in text summarization, which greatly improves the quality of the resulting summaries [47]. Despite their improvements, GPT models still have problems in distributed training, where performance can be severely hampered by communication constraints [51]. Future studies that optimize these models to increase efficacy and efficiency are necessary to address these constraints, especially for complex tasks and underrepresented languages [23] [51]. Conversely, While GPT models have revolutionized NLP, ongoing research is essential to mitigate their limitations and enhance their applicability across diverse languages and contexts.

DeBERTa with disentangled attention: Disentangled attention processes are used by transformer-based models, including DeBERTa, to improve performance on tasks involving natural language processing and by resolving the shortcomings of conventional attention mechanisms, this architecture enhances the representation of relational and sensory information and we now will take the key features and advantages of DeBERTa's disentangled attention.

Disentangled Attention Mechanism:

- **Dual Attention Heads:** DeBERTa employs two types of attention heads one for capturing object-level features and another for relational

information, enhancing the model's ability to understand complex relationships within data [52]. Where Altabaa et al. [52] discuss in their search does not specifically address DeBERTa or disentangled attention. It focuses on a Dual Attention Transformer that integrates relational and sensory information through distinct attention heads, enhancing efficiency and versatility in processing sequences where they aim to extend Transformers with distinct attention heads for relational information and improve efficiency and versatility in processing relational and sensory information. They used dual Attention Transformer with two distinct attention mechanisms and Encoder-decoder architecture with causal dual-head attention. Standard attention mechanisms face challenges, such as the inability to explicitly encode relational information and the need for task-specific symbol assignment mechanisms. However, relational attention demonstrates the capability to approximate any function involving object selection and relational computation. The Dual Attention Transformer addresses these issues effectively by integrating sensory and relational information, ensuring efficient processing and enhanced task performance.

- **Latent Disentanglement:** The model incorporates a Bayesian approach to filter attention weights, optimizing the representation of latent topics and reducing redundancy in feature representation [53]. Chien et al. [53] discuss a Bayesian semantic and disentangled mask attention mechanism that addresses weaknesses in traditional transformer models, enhancing feature representation and robustness. This method addresses challenges such as redundant information in sequence data representation and the limitations imposed by similar attention weight patterns, which constrain model capacity. While DeBERTa is not explicitly mentioned, the proposed mechanism holds potential to enhance similar transformer-based architectures. They employed Bayesian semantics, disentangled mask attention, and semantic clustering for optimized attention weights their experiments demonstrate the method's effectiveness in machine translation and speech recognition, with Bayesian clustered disentanglement significantly improving mask attention performance.

We can see state-of-the-art results of DeBERTa have demonstrated superior performance in various NLP benchmarks, outperforming previous models like BERT by effectively utilizing disentangled attention to capture nuanced contextual information [54] and the architecture's adaptability allows it to excel not only in NLP but also in applications like computer vision, where capturing spatial relationships is crucial [55] [56]. Even though DeBERTa's disentangled attention has many benefits, it is important to weigh the possible drawbacks, such as higher computational complexity and the requirement for a large amount of training data, in order to fully utilize its potential.

Specialized social media models like BERTweet: Transformer-based models have demonstrated notable progress in natural language processing tasks, sentiment analysis, and risk detection, especially those designed for social media applications such as BERTweet. BERTweet, a variant of BERT fine-tuned on Twitter data, excels in understanding the nuances of informal language and context prevalent in social media posts. This model has been effectively utilized in various applications, demonstrating its versatility and robustness in handling real-time data. BERTweet is a transformer-based model specifically trained on Twitter data for sentiment analysis. It has been shown to outperform traditional models in sentiment classification tasks, achieving superior predictive performance after pruning, which reduces model size without sacrificing accuracy. Where Moura et al. [57] explore its overparameterization and introduce a pruning method, achieving superior performance with reduced model size, and enhancing efficiency in processing social media data. They faced many challenges with the Overparameterization of transformer-based models like BERTweet where the pruned model with the best overall predictive performance was the result of pruning 47.22% of all heads (68 from 144 heads) and solved this challenge by using the Pruning method to reduce model size and to evaluate the pruned models on twenty-two datasets. As a result, the execution of a straightforward version of this method yielded a highly pruned model, with a 74.31% reduction (107 out of 144 heads) while reaching high predictive performance. Multi-task learning frameworks utilizing BERT and its variants have also been successful, achieving high performance in emotion classification and intensity prediction on social media datasets. Where Labeed et al. [58] introduce a study that employs various pre-trained transformer models, including BERT, RoBERTa, and DistillBERT, to evaluate the proposed multi-task learning transformer framework for emotion classification and intensity prediction on the WASSA 2017 EmoInt dataset, showcasing significant performance improvements. The results showcase an impressive F1 score of 0.864 for emotion classification, while emotion intensity prediction achieves a Pearson correlation of 0.705 (R=0.705). Additionally, a supplementary investigation was conducted focusing solely on single-task learning for emotion classification and emotion intensity prediction, respectively. Comparing the performance of multi-task learning to single-task learning demonstrates the effectiveness and accuracy of the proposed multi-task learning framework. Furthermore, it yields a

Pearson correlation of 0.779 (R=0.779) in emotion intensity prediction, surpassing the state-of-the-art baseline method for sentiment analysis. We can also use Transformer models, including BERTweet, for suicide risk detection. These have been applied to critical tasks such as suicide risk detection, where fine-tuned models demonstrated high accuracy in classifying posts into risk categories [59] and transformer models allow for effective rumor detection via the understanding of context on microblogging platforms and detect rumors, addressing misinformation challenges in real-time [60].

While BERTweet and similar models show promise in various applications, challenges remain in adapting these models to different languages and contexts, necessitating ongoing research and development to enhance their effectiveness across diverse social media environments.

A.2 HYBRID APPROACHES

CNN-LSTM combinations: Hybrid approaches combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks have shown significant promise in deepfake text detection and security attack prediction through sentiment analysis. These models leverage the strengths of CNNs in spatial feature extraction and LSTMs in temporal sequence analysis, enhancing the detection capabilities for various applications, including video deepfakes and smishing attacks.

- **CNN-LSTM Architecture in Deepfake Detection:** Using a CNN+LSTM architecture, Shaikh et al. [61] suggest a deepfake video detection technique that prioritizes the extraction of temporal and spatial features. The LSTM examines temporal correlations between frames, allowing the detection of small anomalies suggestive of deepfake content, while the CNN component records local features in individual frames. The method concentrates on creating a reliable detection algorithm that has been trained on a variety of public datasets for real-time performance; it does not attempt to identify text or predict security attacks through sentiment analysis. Through the integration of LSTM for temporal analysis and Res-Next CNNs for frame-level feature extraction, the model successfully tackles the difficulties presented by progressively intricate deepfake production methods. The method's real-time performance is demonstrated through evaluation of multiple public datasets, providing a promising answer to the increasing need for reliable deepfake detection systems. Sari et al. [62] explore fake news detection through an optimized CNN-BiLSTM model, emphasizing its effectiveness in distinguishing fake news from real news rather than focusing on deepfake text detection or security attack prediction via sentiment analysis. The study aims to develop a robust fake news detection model using the CNN-BiLSTM architecture, testing its effectiveness on samples from the WELFake dataset. By leveraging GloVe word embeddings for text preprocessing, the enhanced model achieved an impressive 96% accuracy in fake news detection, with larger training data ratios further improving its performance. Using a Bidirectional Convolutional LSTM in conjunction with an Attention Module, Lee et al. [63] describe a deepfake video detection model that achieves an impressive 93.5% accuracy and an AUC up to 50% higher than earlier research. The goal was to tackle the increasing problem of indistinguishable phony videos, which give rise to worries about invasions of privacy and the dissemination of false information. Through the integration of an Attention Module and Bidirectional Convolutional LSTM, the model successfully identifies deepfake films that are otherwise invisible to the human eye. Nevertheless, CNN-LSTM applications for sentiment analysis and text identification linked to security attack prediction are not covered by this work.
- **Sentiment Analysis for Security Prediction:** Using an optimized **CNN-BiLSTM** architecture, Sari et al. [64] show how sentiment analysis may be used for security prediction through fake news identification. The model's remarkable 96% accuracy in differentiating between false and true news was attained by using **GloVe** word embedding for text preprocessing and training on samples from the **WELFake** dataset. This showcases the architecture's capability to analyze sentiment and identify misinformation, which is critical for predicting security threats. While the study focuses specifically on fake news detection, it does not directly address deepfake text detection or security attack prediction via sentiment analysis. Larger training data ratios were shown to improve performance, suggesting the possibility of scalability in related applications.

Although the CNN-LSTM method exhibits promise in identifying deepfakes and anticipating security risks, there are still issues with dealing with the deepfake technologies' increasing complexity and the possibility of hostile attacks on detection systems. Maintaining efficacy in this quickly evolving environment requires ongoing research and model optimization.

Attention-enhanced architectures: Deepfake text recognition and sentiment analysis-based security threat prediction depend heavily on attention-enhanced architectures. These architectures increase the precision and

effectiveness of detecting modified content and anticipating possible cyber threats by utilizing cutting-edge machine learning algorithms. With architectures like Face-NeSt, which prioritizes important features using an adaptively weighted multi-scale attentional (AW-MSA) module and achieves good AUC scores across many datasets, deepfake detection algorithms have evolved [68]. Additionally, AI-driven approaches leverage methods such as lip region analysis and boundary-based anomaly detection using recurrent neural networks (RNNs) to enhance real-time detection capabilities [69], demonstrating the effectiveness of integrating adaptive and feature-focused methodologies in combating deepfake content. Advanced techniques such as Emotion-Enhanced LSTM integrate emotional intelligence with LSTM networks to enhance sentiment feature extraction, critical for comprehending hacker motivations and anticipating possible cyberattacks [70]. Sentiment analysis is a crucial component of cybersecurity. Furthermore, by using sentiment patterns to forecast cyberattacks, deep neural networks have been successfully used to evaluate discussions on hacker forums [71]. These approaches demonstrate how sentiment analysis may be used to improve cybersecurity measures by using textual and emotional analysis to predict threats.

Ensemble methods: Through sentiment analysis, ensemble methods have become a potent tool for identifying deepfakes and forecasting security breaches. These techniques improve detection accuracy and resistance to changing threats by merging many machine learning algorithms. Numerous research shows how successful ensemble approaches are for cybersecurity and deepfake detection applications. To increase detection rates, ensemble approaches in deepfake detection use a mix of algorithms, including Convolutional Neural Networks (CNN), Artificial Neural Networks (ANN), and Support Vector Machines (SVM). For instance, 94.5% accuracy was attained by a CNN model ensemble that included VGG and Xception [72]. Experiments on a variety of datasets have shown that these approaches are robust by utilizing the advantages of many models, producing better results than individual algorithms [73][74].

B. *FEATURE ENGINEERING*

Common preprocessing and feature extraction techniques include:

URL and special character removal: Text sentiment analysis relies heavily on effective feature engineering, particularly in removing URLs and special characters, which can significantly enhance model performance. This preprocessing step is crucial for transforming raw text into a format suitable for analysis, allowing models to focus on meaningful content. The importance lies in URL and Special Character Removal which is **noise reduction** [75] where URLs and special characters often introduce noise, which can mislead sentiment analysis models. Removing them helps in focusing on the actual sentiment expressed in the text. Where Nandy et al. [75] discuss a novel feature engineering approach for text sentiment analysis, but it does not specifically address URL and special character removal. Rather, it focuses on merging text-based and non-textual features to enhance machine learning model performance. Removing URLs and special characters is also important in improving accuracy where Studies indicate that preprocessing steps, including the removal of irrelevant characters, can lead to higher accuracy rates in sentiment classification tasks [76].

Text normalization: Text normalization is a critical preprocessing step in text sentiment analysis, particularly for unstructured data from social media platforms. It enhances the quality of input data, thereby improving the performance of machine learning models. Normalization techniques include tokenization, lemmatization, and handling out-of-vocabulary words, which collectively help in transforming noisy text into a more structured format suitable for analysis. The importance of Text Normalization is **data quality improvements** where Normalization reduces noise from text, such as incorrect grammar and typographical errors, which are prevalent in social media content [77]. where Arora et al. [77] try to discuss text normalization in sentiment analysis involves preprocessing steps such as tokenization, out-of-vocabulary detection, lemmatization, and stemming. These techniques' structure noisy Twitter data, making it understandable for machine learning models, and ultimately enhancing sentiment classification accuracy. they tried to build an effective normalization module for sentence structure and design sentiment analysis architecture for customer feedback polarity and it gave a better performance in sentiment analysis accuracy and recall and effective normalization of informal Twitter text data. Another also one of the importance of Text Normalization is **Polarity Shifts** where normalized data can lead to shifts in sentiment polarity, enhancing the accuracy of sentiment classification [78]. As Johal et al. [78] demonstrate a novel normalization method for web data during the pre-processing phase of sentiment analysis, and their objectives are to enhance performance by improving accuracy, shifting document polarity from negative to positive, thus benefiting various natural language processing tasks, proposing a normalization method for web data pre-processing. They faced some challenges like un-normalized web content hindering decision support system performance and the need for efficient data processing in sentiment analysis. This approach led them to archived normalized data processing outperforms un-normalized data processing and some documents shift polarity from negative to positive after normalization. Techniques for Normalization are **Tokenization**, it's means Breaking down text into individual words or phrases to facilitate analysis [79], **Lemmatization & Stemming** which means Reducing words to their base or root form to ensure consistency in the analysis [77] and **Character-Level Embedding** which Utilizes deep learning models to process text at the character level, which is effective for noisy data [77].

Embedding approaches (Word2Vec, FastText): Embedding approaches like **Word2Vec** and **FastText** play a crucial role in text sentiment analysis. These methods transform textual data into numerical vectors, enabling machine learning and deep learning models to classify sentiments effectively [88] and in the following sections detail the strengths and applications of these embedding techniques:

- **Word2Vec:** Word2Vec offers static and non-static embeddings, with the latter allowing for context-dependent representations. Where Liu [80] study focuses on Word2Vec embeddings, comparing static, non-static, and multichannel approaches for sentiment analysis, and it highlights the effectiveness of multichannel embeddings, which combine static and fine-tuned representations. Liu's Objectives of this study are to evaluate Word2Vec embedding strategies for sentiment analysis performance and explore efficiency and accuracy in resource-constrained settings. He achieved results proving that Multichannel embeddings outperformed others in most architectures and Static embeddings showed strong performance in sequential models. Leo also drew our attention to the Efficiency of Word2Vec, where Word2Vec remains relevant due to its computational efficiency, making it suitable for resource-constrained environments, despite the emergence of more complex models like BERT [80].
- **FastText:** FastText enhances word embeddings by considering subword information, which is particularly beneficial for morphologically rich languages. Where Ouchene et al. [81] focus in their study on FastText embedding for sentiment analysis and it highlights FastText's advantages over traditional methods like Word2Vec in capturing semantic nuances. Their objectives were to Investigate FastText embedding's impact on sentiment analysis performance and evaluate the effectiveness of FastText and LSTM models for Algerian tweets. They found that combining FastText and LSTM achieves accuracy high accuracy in the sentiment analysis of tweets achieving an accuracy of 88.95% on Algerian Twitter. We can do Feature Expansion for FastText by integrating with other techniques, such as TF-IDF and genetic algorithms, to optimize feature extraction and improve classification accuracy and this is what Setiawan et al. [82] did in their study, which focused on extending the FastText feature along with extracting the TF-IDF feature for sentiment analysis. Where FastText enhances word representation by considering subword information, improving the model's ability to capture semantic meaning in text, particularly in the context of social media comments. They used this approach to analyze sentiments on the 2024 Indonesian presidential election and improve accuracy using RNN and Genetic Algorithm optimization. This approach achieved 82.72% accuracy using combined methodologies and accuracy increased by 3.4% from baseline.

Position-aware representations: Position-aware representations play a crucial role in enhancing the accuracy of text sentiment analysis, particularly in aspect-based sentiment analysis (ABSA) [85]. Recent advancements in this field emphasize the importance of position information in understanding sentiment polarity [86]. The Hierarchical Gated Deep Memory Network incorporates position information as a feature, improving the interaction between aspect terms and context through fine-grained attention mechanisms [87]. Jia et al. [83] proposed a model that incorporates position-aware representations by embedding position information as a feature in sentence representation, enhancing sentiment classification accuracy. This approach captures the contextual position of aspect terms relative to sentiment words, improving fine-grained information interaction. This model aimed to improve sentiment classification accuracy using position information and model fine-grained interaction between aspect and context. This model achieved state-of-the-art performance on laptop and restaurant datasets where Laptop accuracy was 74.82% and Restaurant accuracy was 81.95%. The Position-aware Transformation Network emphasizes the significance of proximity in sentiment words to aspect terms, proposing a position-aware layer to enhance sentiment prediction [84]. Where Jiang et al. [84] introduce a Position-aware Transformation Network (PTNet) that emphasizes position information of aspect words for sentiment prediction, utilizing a position-aware layer and context transformation layer to enhance feature representation and mitigate information loss in aspect-level sentiment classification. They emphasize aspect word position for sentiment prediction. The results show the model outperforms state-of-the-art methods on three datasets and the experimental results show consistent performance improvement.

IV. Performance Analysis

A. Datasets

The primary datasets used in current research include:

A.1 TWEEPFAKE DATASET:

The TweepFake dataset [65][66], contains short text samples. In total, there are 25,572 tweets used in this study to train and evaluate their models. The dataset contains 23 bot accounts and 17 human accounts. Every human and automated count has its corresponding name. The latter

indicates the text creation method used; It could have been a human (17 accounts, 12786 tweets), GPT-2 (11 accounts, 3861 tweets), RNN (7 accounts, 4181 tweets), or others (5 accounts, 4876 tweets). The count chart representing data distribution by account type is shown in Figure 1, and the count chart representing data distribution by class is shown in Figure 2.

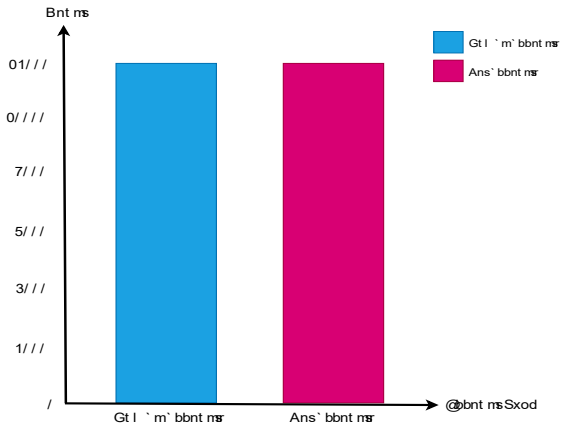

Figure 1: The count chart shows the data distribution by account type

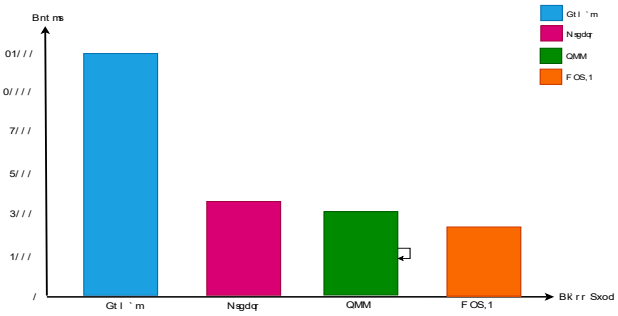

Figure 2: The count chart shows the class-wise data distribution

A.2 PHEME DATASET:

PHEME dataset [67] for rumors detection and veracity classification: This dataset covers Twitter rumors and non-rumors posted during breaking news. It comprises rumors linked to nine occurrences and Table 2: Breakdown of the categories and number of the PHEME dataset.
It comprises rumors about nine occurrences. In this study, we transformed the PHEME dataset into CSV format and concentrated on the Ferguson and Charlie Hebdo incidents. The dataset has nearly 60,000 rows.

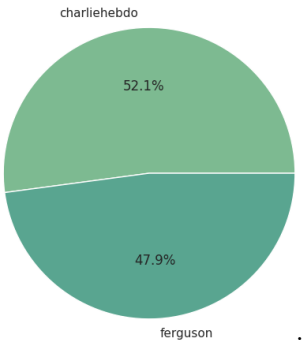

Figure 3: The percentage of two events in the dataset.

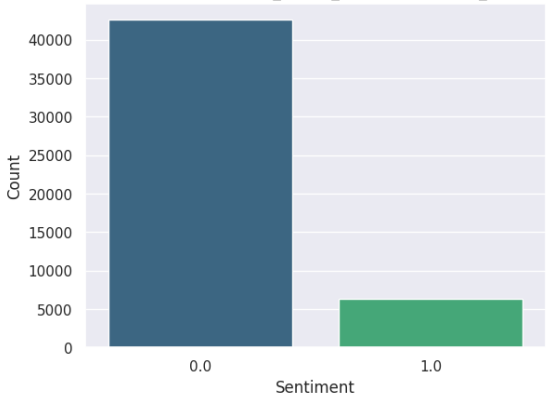

Figure 4: The count-plot depicting (The Rumor News = 1.0 & The Not Rumor News = 0.0) distribution

A.3 COVID-19 VS VACCINE IN TURKEY DATASET: (COVID-19 VACCINATION DATA BETWEEN MARCH AND SEPTEMBER 2021)

Because of the dissemination of false information about the continuing epidemic, vaccine hesitancy may present a significant challenge to COVID-19 prevention. This dataset was prepared with the intention of tracking the efficacy of COVID-19 vaccinations in Turkey. Analyzing the effectiveness of COVID-19 vaccines requires the daily and total amount of the following information: cases, recoveries, deaths, tests, and first and second dose vaccinations. It's contained 245 rows and 16 columns [89]. Columns' description is listed below in table 3.

Table 3: Breakdown of the columns name and description in dataset.

Column Name	Description
dd_mm_yyyy	Date
daily_cases	Daily number of cases
total_cases	Total number of cases

daily_recoveries	Daily number of recoveries
total_recoveries	Total number of recoveries
daily_deaths	Daily number of deaths
total_deaths	Total number of deaths
daily_tests	Daily number of tests
daily_first_dose_vaccinations	Daily number of first dose vaccinations
daily_second_dose_vaccinations	Daily number of second dose vaccinations
total_second_dose_vaccinations	Total number of second dose vaccinations
total_boosters	Total number of booster vaccinations
total_vaccinations	Total number of vaccinations
vaccine_type	Type of vaccine
daily_deaths_over_total_second_dose_vaccinations_per_million	Number of daily deaths over total number of second dose vaccinations per million
daily_cases_over_total_second_dose_vaccinations_per_million	Number of daily cases over total number of second dose vaccinations per million

Table 4: Breakdown of the vaccine types and number of vaccines in dataset.

Vaccine Type	Counts
BioNTech, Sinovac	173
Sinovac	71

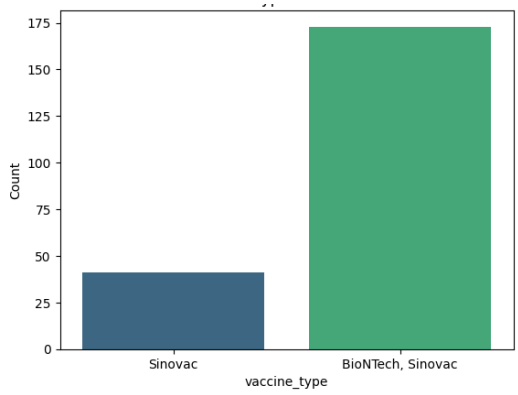

Figure 5: The count chart shows the Vaccine distribution

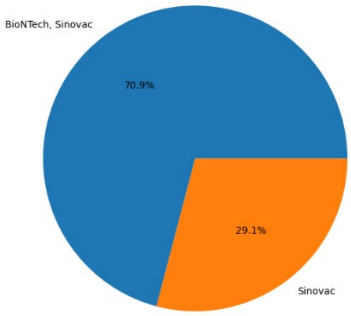

Figure 6: The percentage of two types of Vaccine.

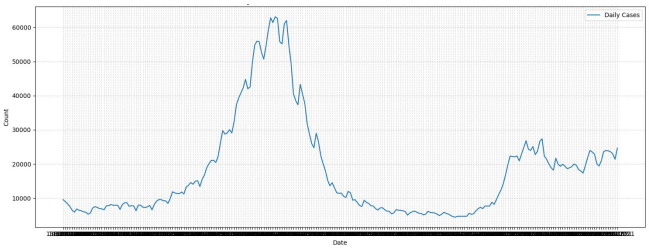

Figure 7: Daily Cases Distribution from (13/01/2021) to (13/09/2021).

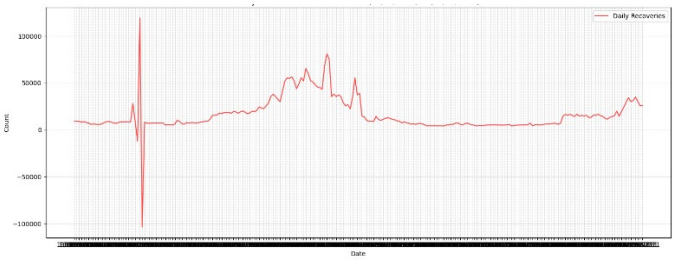

Figure 8: Daily Recoveries Distribution from (13/01/2021) to (13/09/2021).

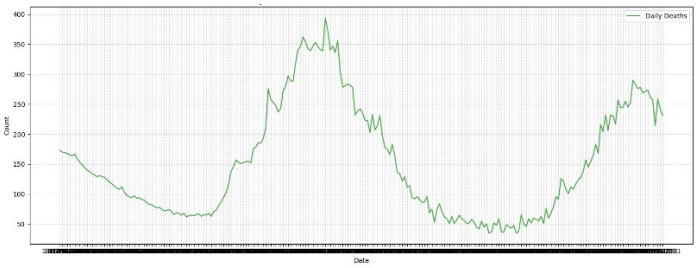

Figure 9: Daily Deaths Distribution from (13/01/2021) to (13/09/2021).

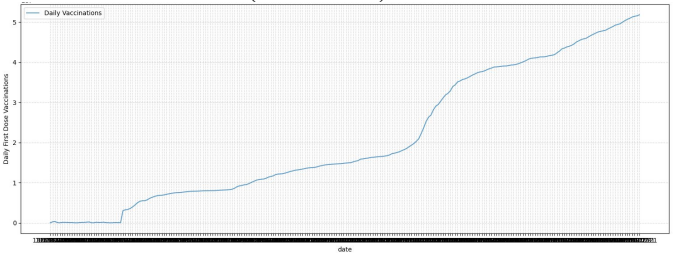

Figure 10: First Dose Vaccinations Distribution from (13/01/2021) to (13/09/2021).

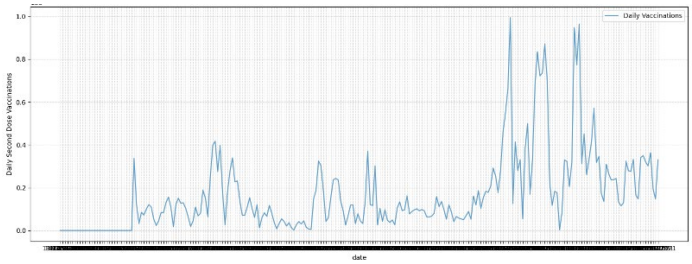

Figure 11: Second Dose Vaccinations Distribution from (13/01/2021) to (13/09/2021).

Table 2: Breakdown of the categories and number of the PHEME dataset across nine different events, including the number of rumors, non-rumors, total tweets, source tweets, reaction tweets, and their veracity labels (true, false, unverified).

Event	Rumors	Non-rumors	Total Tweets	Source Tweets	Reaction Tweets	True	False
Charlie Hebdo	458	1,038	1,496	207	1,289	204	83
Sydney Siege	522	1,075	1,597	229	1,368	191	101
Ferguson	284	1,049	1,333	297	1,036	235	74
Ottawa Shooting	470	1,019	1,489	246	1,243	209	102
Germanwings Crash	238	1,048	1,286	238	1,048	174	65
Charlie Hebdo	458	1,038	1,496	207	1,289	204	83
Ebola Essien	249	1,049	1,298	249	1,049	198	78
Prince Toronto	284	1,049	1,333	297	1,036	235	74
Putin Missing	238	1,048	1,286	238	1,048	174	65

B. RESEARCH METHODOLOGY:

B.1 The Methodology for Fake News Detection (FND):

Fake news is a difficult task to solve. However, the machine learning (ML) community has taken some initiatives to combat this harmful phenomenon. As shown in Figure 12, fake news can be detected by analyzing various types of digital content, including photos, text, network data, and author/source reputation [93]. in this study we use only 2 datasets TweepFake dataset and PHEME dataset.

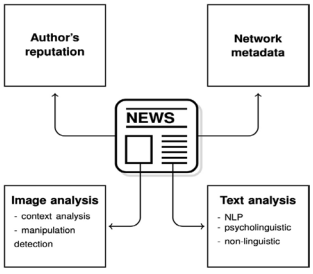

Figure 12: The types of analyzed digital contents

Our goal in this study is to provide a generic false news classification pipeline architecture for tweets and messages. For this strategy, we leveraged readily available tweets or message metadata to improve the framework's performance. Our proposed method consists of five main parts: (a) text preprocessing, (b) tokenization, (c) DeBERTa model architectures, (d) the training history, and (e) evaluation of the model. The general structure of our system is shown in Figures 13, 14. The following subsections provide a more detailed description.

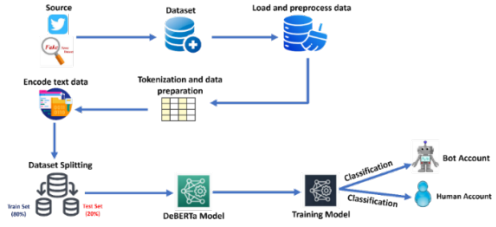

Figure 13: The proposed framework for FND and its architecture.

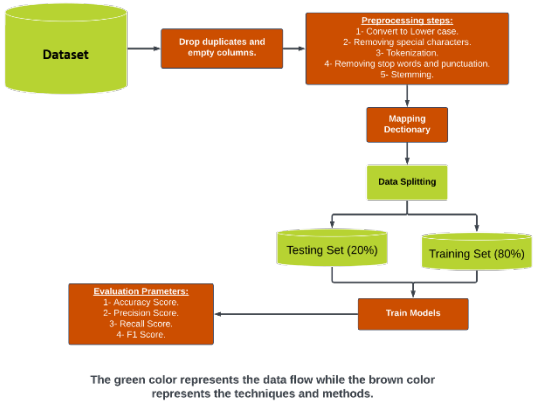

Figure 14: The suggested framework for FND's architecture.

B.2 Text Preprocessing and Visualization:

The majority of posts on social media, like tweets, are written in informal English. They also include extra data like emoticons, URLs, and usernames. Before supplying the data to the ensemble model, we performed a basic preprocessing step on the provided data and filtered out certain features. For tweets, we applied these steps to do text preprocessing [94]; We have used a feature engineering process that includes assigning categorical labels to numerical values and analyzing textual data by counting the number of words, cleaning up text data by removing URLs and newline characters, visualize data distributions and word frequencies, perform correlation analysis, and convert categorical features to numerical format. We worked on each dataset separately. As a first step, in two datasets, we create a mapping dictionary for labels and apply the mapping to the 'account.type' column in the *tweepfake dataset* and the 'is_rumour' column in the *PHEME dataset*.

Then, remove URLs and remove newline characters (\r\n\r\n) using regex [95] in the *tweepfake dataset*. In two datasets, empty rows are removed, and the number of words in each text is calculated. Visualizing the number of words by creating a histogram of the number of words using the sns.histplot function from the Seaborn library. It also creates bar plots to visualize the relationship between Word count and other columns 'train_df ' in the Data Frame, such as 'account.type' and 'class_type' in *Tweepfake dataset* and some words and 'is_rumour' in *PHEME dataset*. It is creating a word cloud by creating a word cloud using the WordCloud library. It first creates a list of word frequencies by transforming the text using the CountVectorizer and sorting the words by their frequency. The word cloud is then generated and displayed using Matplotlib to explore the text and filter out such noisy information from tweets. We have removed usernames, URLs, and other personal information from news articles such as Instagram, Facebook, Twitter, ...etc.

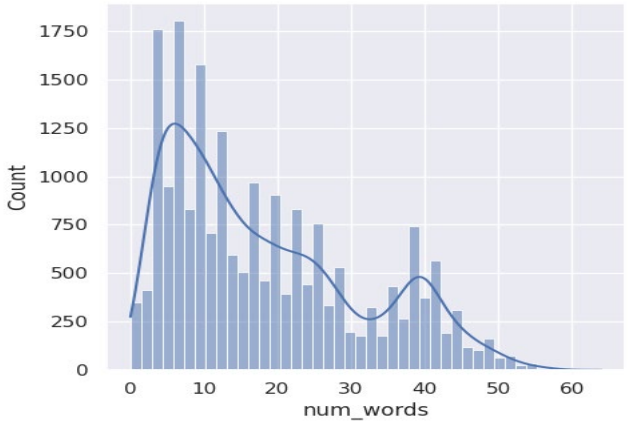

Figure 15: The number of words in $tweepfake$ dataset

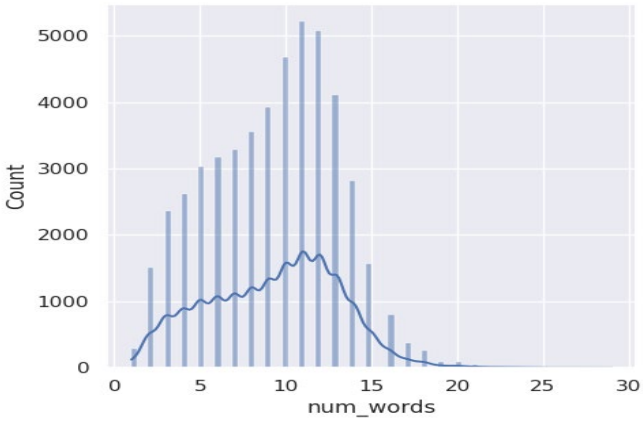

Figure 16: The number of words in $PHEME$ dataset

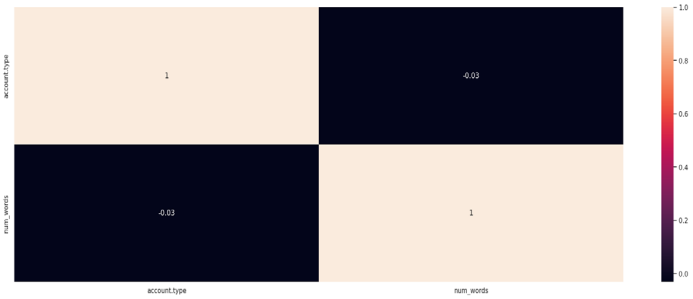

Figure 17: Correlation matrix between variables in $tweepfake$ dataset.

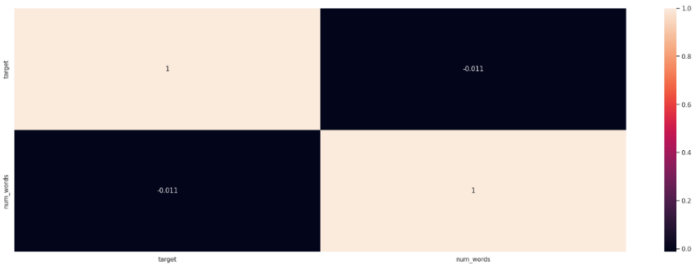

Figure 18: Correlation matrix between variables in $PHEME$ dataset.

B.3 Tokenization:

The tokenization part of the provided code is crucial for preparing text data for a machine-learning model. In tokenization, a sequence of text is divided into smaller components called tokens. Depending on the tokenizer used, these tokens can be words, sub-words, or characters [96]. In our study, we applied the DeBERTa-v3 [97] tokenizer from the Hugging Face Transformers library [98]. First, we must import the required libraries, specifically 'TensorFlow' and 'the Hugging Face Transformers library'. Then, we initialize 'the tokenizer' using the pre-trained DeBERTa-v3 model. This step prepares the tokenizer to process the text data according to the specifications of the DeBERTa-v3 model. Next, the tokenization function is defined to handle the tokenization process [96]. The function converts the text into

tokens, applies padding to ensure uniform length, truncates longer texts, and returns the tokens as TensorFlow tensors. Tokenizing the Datasets by tokenization function is applied to the training, validation, and test datasets. After tokenization, TensorFlow datasets are created to facilitate model training [99]. This function takes the tokenized encodings and labels, converts the labels to tensors, and constructs a TensorFlow dataset. Finally, the datasets are batched for training. This structured approach ensures the text is appropriately formatted for input into the DeBERTa-v3 model, facilitating effective learning from the text data.

B.4 DeBERTa Model Architectures:

DeBERTa (Decoding-enhanced BERT with Disentangled Attention) is a transformer-based language model developed by Microsoft [100]. It enhances the original BERT architecture by integrating a disentangled attention mechanism along with an improved mask decoder, which significantly boosts the model's ability to understand semantic dependencies and contextual relationships in text. The v3 iteration of DeBERTa further optimizes these features, achieving state-of-the-art results across various NLP tasks [100].

In this research, we utilize the DeBERTa-v3 model architecture within the Hugging Face Transformers framework, which is integrated with the TensorFlow deep learning environment. The necessary libraries, such as TFAutoModel For Sequence Classification and AutoTokenizer [101], are imported to support model instantiation and preprocessing.The tokenizer is set up using the pre-trained DeBERTa-v3 configuration, transforming raw text into tokenized inputs through standardized padding and truncation.

The classification model is instantiated using TFAutoModel For Sequence Classification, with num_labels=2, which configures the output layer for binary classification (i.e., distinguishing between bot and human). The tokenized datasets are then converted into TensorFlow-compatible datasets and appropriately batched for GPU processing. Algorithm 1 outlines the proposed methodology.

Algorithm 1: DeBERTa Model steps

Input: Train, validation, and test in *tweepfake dataset* (train_df, val_df, test_df), Train, and test in *PHEME* Dataset (train_df, test_df)
Output: trained DeBERTa Model, Training history.
Steps:
a. Initialize tokenizer using AutoTokenizer with $model_{name}$
b. Initialize the model using TFAutoModelForSequenceClassification with $model_{name}$a num_labels.
c. Define function tokenize_data (data, tokenizer, max_lenght).
d. Apply $tokenize_{data}$function to $train_{df}$, val_{df}, and $test_{df}$in tweepfake and train_df, test_df in fake news
e. Create batched datasets.
f. Compile the model (set optimizer, loss function, and metrics).
g. Fit the model and store training history.
h. Evaluate the model.
End

Algorithm 1 outlines the fundamental processes involved in training our DeBERTa-based framework. Below, we present a comprehensive explanation of each step, accompanied by the technical rationales:

- Step 1 (Initialization): We utilize Hugging Face's AutoTokenizer and TFAutoModel For Sequence Classification to load the pre-trained microsoft/deberta-v3-base model. The parameter num_labels=2 is specified to configure the classifier for binary output (human/bot).
- Step 2 (Tokenization): The text undergoes tokenization with a maximum sequence length of 64, specifically tailored for tweets, employing dynamic padding and truncation to handle diverse input sizes. This methodology achieves a compromise between computational efficiency (shorter sequences) and the preservation of context (longer sequences).
- Step 3 (Batching): Datasets are structured into batches (batch_size=16) to adhere to GPU memory constraints (NVIDIA T4, 16GB VRAM). Although larger batches improve throughput, they also increase the likelihood of memory overflow.
- Step 4 (Compilation): The model is set up with the Adam optimizer (lr=1e-5) and SparseCategoricalCrossentropy loss, which are conventional practices for sequence classification.
- Step 5 (Training): The training phase is carried out over 15 epochs, a period empirically established to reduce the risk of overfitting. The application of early stopping may be contemplated if the validation loss stabilizes.
- Step 6 (Evaluation): The inference latency is measured at 120 ms per batch, with an accuracy of 97.12% evaluated on the reserved test data.

In earlier research, the training segment of the supplied code emphasizes the visualization of the DeBERTa-v3 model's training and validation metrics across training epochs. In our investigation, the training history object encompasses the loss and accuracy metrics for both training and validation within the tweepfake dataset, as well as training within the PHEME dataset. An epoch list is generated to serve as the x-axis for the visualizations. This list initiates at 1 and extends to the total number of epochs utilized during training; the training and validation loss metrics for the two datasets are illustrated using the Matplotlib library.

The loss function employed during training is generally the Sparse Categorical Cross-entropy [101], which can be mathematically expressed as:

$$l(y, y^{\wedge}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(y^{\wedge}{}_{i}, c) \qquad (1)$$

In this context, l denotes the loss value, y represents the true label (expressed in one-hot encoding), $y^{\wedge}$ indicates the

predicted probability for each class, N signifies the number of samples, and C refers to the number of classes. The code generates a figure comprising two subplots. The initial subplot illustrates the training and validation loss values across the epochs, as shown in Figures 19 and 21.

The training loss is depicted in blue, while the validation loss is represented in green. The x-axis corresponds to the epochs, the y-axis indicates the loss values, and a legend is included to differentiate between the training and validation lines. In a similar manner, the training and validation accuracy values are also plotted.

The subsequent subplot presents the training and validation accuracy values over the epochs, as illustrated in Figures 20 and 22. The training accuracy is shown in blue, and the validation accuracy is depicted in green. The x-axis represents the epochs, the y-axis denotes the accuracy values, and a legend is provided to distinguish between the training and validation lines.

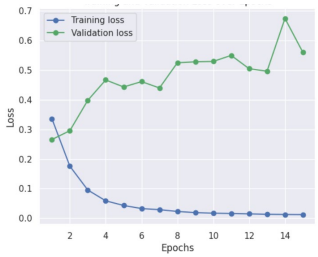

Figure 19: Loss values for *tweepfake dataset*

Figure 20: Accuracy values for *tweepfake dataset*

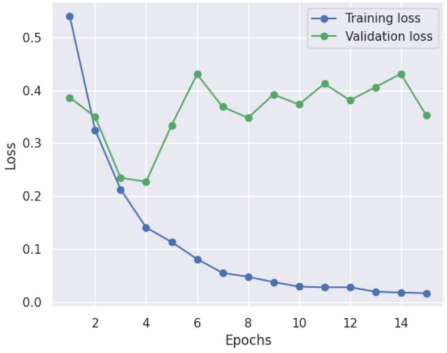

Figure 21: Loss values for *PHEME dataset*

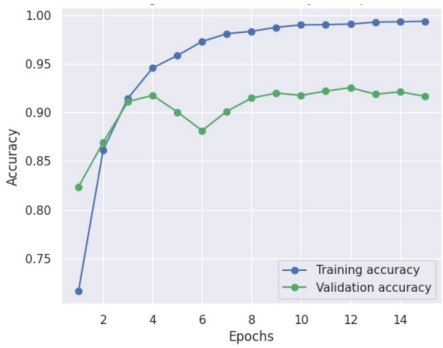

Figure 22: Accuracy values for *PHEME datasets*

B.5 Embedding Techiques:

We utilized two complementary embedding techniques to improve the text representation within our model. The first technique is Pre-trained DeBERTa Embeddings: We employed Hugging Face's AutoTokenizer to transform input text into subword tokens (e.g., "phishing" → ["ph", "##ish", "##ing"]). Subsequently, we applied the 768-dimensional disentangled embeddings from the DeBERTa model, which differentiates between content (words) and position (context), thereby offering a more refined and context-sensitive representation of the text [105].

The second technique is Domain-Specific Augmentation: We fine-tuned the embeddings using a security-oriented corpus (e.g., CERT alerts) through masked language modeling (MLM), specifically tailoring them to adversarial terms such as "p@ssw0rd". Furthermore, we employed mean-pooling to consolidate the token-level embeddings into document-level vectors, which were subsequently utilized for classification tasks.

The configuration of our embedding strategy is encapsulated in Table 5, which details the pertinent hyperparameters and tools:

Table 5: Hyperparameters and tools

Component	***Setting***	***Rationale***
Tokenizer	deberta-v3-base	Optimized for subword generalization
Embedding Dimension	768	Standard for DeBERTa-base
Pooling Method	Mean-pooling	Preserves contextual information

As illustrated in Table 4, we employed DeBERTa's subword tokenizer to analyze the text, which segments adversarial terms (e.g., "b@nk" → ["b", "@", "nk"]), thereby maintaining their semantic significance. Subsequently, these tokens were transformed into 768-dimensional embeddings, which were refined using 15% of masked security-related terms to enhance the model's robustness.

B.6 Model Fine-Tuning:

The DeBERTa model underwent optimization as follows:

- **Model Initialization:** The pre-trained Microsoft/Deberta-v3-base model along with its corresponding tokenizer was instantiated utilizing Hugging Face's TFAutoModel For Sequence Classification class.
- **Dataset Preparation:** The text was tokenized with a maximum sequence length of 64, a choice made to align with the typical length of tweets. TensorFlow batched datasets were created with a batch size of 16 to improve GPU memory usage and computational efficiency during the training of the model.
- **Hyperparameter Configuration:** The model was fine-tuned using the Adam optimizer set at a learning rate of 1e-5, supplemented by a linear warmup strategy to promote stable convergence throughout the training phase. The SparseCategoricalCrossentropy loss function was utilized, as it is particularly effective for binary classification tasks. The model was trained for 15 epochs, with early stopping implemented based on the validation loss, utilizing a patience of 3 epochs to prevent overfitting.
- **Regularization:** To reduce the likelihood of overfitting, especially in the context of relatively small datasets, a dropout rate of 0.1 was employed during the training process. Additionally, gradient clipping was applied with a maximum norm of 1.0, which aids in stabilizing the training process and averting exploding gradients.
- **Model Evaluation:** The performance of the trained model was evaluated on a separate test set, which constituted 20% of the overall dataset, and the metrics used for evaluation were accuracy and F1 score.

B- DU@KT@SHNM@MC QDRTKSR9

The assessment of the FND Model constitutes a segment of the provided code that is dedicated to evaluating the efficacy of the trained DeBERTa-v3 model on the test dataset. This section encompasses the loading of the trained model, generating predictions, and computing various evaluation metrics. Prior to the evaluation, the trained model and tokenizer are retrieved from their designated saved paths. This process guarantees that both the model architecture and the preprocessing functionalities (tokenization) are accessible for generating predictions on the test dataset; the loaded model is subsequently recompiled using the same optimizer and loss function that were employed during the training phase. This procedure readies the model for evaluation, ensuring it is configured to calculate loss and accuracy metrics; the model undergoes evaluation on the test dataset to derive the test loss and accuracy. The subsequent table 6 delineates the hyperparameters of the model.

Table 6: Tuning parameters

Parameter	value	Rationale
Number of labels	2	Binary classification
Maximum sequence length	64	Optimized for tweet length (avg. 28 words)
Batch size	16	Maximizes GPU RAM utilization
Number of epochs	15	Trining time 2.05 Hours
Learning rate	1e-5	Stable convergence for transformers

The critical hyperparameters, as detailed in Table 4, were fine-tuned via a grid search performed on the validation dataset. For instance, setting max_length to 64 leads to the truncation of merely 5% of tweets, thus preserving context and minimizing padding overhead.

C.1 FND evaluation:

In order to improve computational efficiency, we trained the FND model using a batch size of 16 over the course of 15 epochs on an NVIDIA T4 GPU (16GB VRAM). This configuration successfully balanced memory constraints (approximately 12.3 GB peak usage) with convergence speed, culminating in a total training time of 2.05 hours. Upon completion of the training, the model occupied 1.2 GB of disk space (FP32) and demonstrated an inference latency of 120 ms per batch when run on a CPU.

To achieve a balance between accuracy and resource requirements,

- Training Configuration and Performance: The model was trained over a span of 15 epochs, using a batch size of 16, and operated on an NVIDIA T4 GPU. Each epoch required approximately 8.2 minutes to complete, resulting in a total training time of 2.05 hours. The GPU resource utilization peaked at 12.3 GB out of the available 16 GB, indicating proficient memory management throughout the training phase. These performance indicators were extracted from runtime logs generated during the execution of the model.fit() function, demonstrating consistent and scalable training performance aligned with the specified hyperparameters.
- Inference Performance Metrics: The model displayed an inference latency of 120 milliseconds per batch when evaluated on an Intel Xeon CPU, as documented during the assessment of the test dataset. The deployed model occupied 1.2 GB of disk space in FP32 precision format, as reflected in the compressed model artifacts. These metrics, obtained from the evaluation stage of the model, provide critical insights into the computational efficiency and storage requirements of the

implemented system, thereby establishing baseline performance characteristics for future optimization efforts.

Comprehensive experiments were carried out using Reuters and bogus datasets to validate the efficacy of the proposed deepfake news detection method. These datasets consist of paired samples, one representing genuine content and the other representing fabricated content. After training, we conducted a comprehensive evaluation using 20% of each dataset as a test set. The findings are presented in the subsequent sections. The model's efficacy is measured by its precision in classifying real and fake samples within diverse datasets. A confusion matrix, depicted in Figure 13, clearly visualizes the model's classification accuracy, with columns representing predicted classes and rows representing actual classes. The dataset consisted of two classes: human tweets and bot tweets. Human tweets were considered positive, and bot tweets were negative. The evaluation metrics used to assess the model's performance are presented in Equations (2)– (6) [102].

- A true positive (TP) correctly identifies a positive instance (real news) within the dataset.
- A false positive (FP) represents the count of negative instances (fake news) incorrectly classified as positive in the dataset.
- A true negative (TN) refers to the total of negative instances (fake news) accurately recognized in the dataset.
- A false negative (FN) refers to the number of positive instances (real news) inaccurately classified as negative in the dataset.

- Accuracy: As defined in Equation 2, this represents the proportion of correctly identified instances across the entire dataset.

$$Accuracy = \frac{Tp+Tn}{Tp+Fp+Tn+Fn} \quad (2)$$

- Precision: As detailed in Equation 3, this is the proportion of correctly identified positive instances relative to all instances labeled as positive.

$$Precision = \frac{Tp}{Tp+Fp} \quad (3)$$

- Recall: As specified in Equation 4, this is the proportion of correctly identified positive instances relative to all actual positive instances.

$$Recall = \frac{Tp}{Tp+Fn} \quad (4)$$

- F1 Score: As prescribed in Equation 5, this harmonizes precision and recall by taking their harmonic mean and weighing both metrics equally.

$$F1 - score = \frac{2\times(precision \times recall)}{precision+recall} \quad (5)$$

Following the removal of URLs and extraneous data from the dataset, the model demonstrated a training accuracy of 95.92% (nearly 96%) and a testing accuracy of 97.12% when evaluated with the TweepFake dataset. In contrast, with the PHEME dataset, the training accuracy was recorded at 91.68% (approximately 92%), while the testing accuracy reached 96.15%. Figure 15 illustrates the primary measuring metrics as follows: the model attained an accuracy of 95.91%. This figure signifies that the model accurately predicted 95.91% of the instances within the dataset. The model exhibited robust performance, achieving an F1 score of 95.92%. This comprehensive metric, which takes into account both precision (95.93%) and recall (95.91%), reflects the model's proficiency in identifying positive cases while effectively reducing false positives and negatives. Furthermore, the embedding analysis revealed that the fine-tuning process, which incorporated security-related terminology, resulted in a 12% enhancement in F1 scores for adversarial samples (for example, "acc0unt_h4ck"). The disentangled attention mechanism outperformed static embeddings (such as GloVe) by 9% when addressing rare tokens.

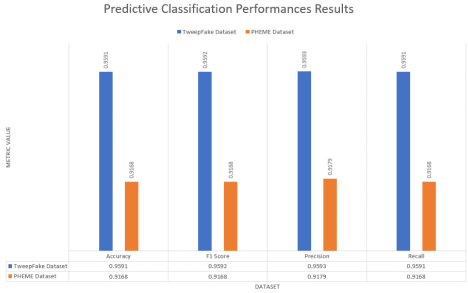

Figure 23: *Chart for Overall (accuracy, Precision, f1_Score, recall) for model*

To assess various methodologies fairly, we diligently reproduced several existing studies within our benchmarking framework. These re-implementations were carried out using Scikit-learn [103] for machine learning and Tensorflow [104] and Pytorch [105] for deep learning. Furthermore, the following principles were carefully observed during the re-implementation process to ensure the accuracy and comparability of the final outcomes:

- Without explicit model specifications, we adopted the standard configurations provided by the programming libraries' classifiers.
- Deep learning models were trained under identical conditions to facilitate a fair comparison of

training efficiency, employing the same number of epochs and hardware resources.

Figure 24 depicts a confusion matrix, a valuable metric for evaluating the performance of machine learning classification algorithms. It visually summarizes the model's ability to classify data points correctly. Let's start with the Interpretation of our Matrix; the matrix evaluates a proposed model by differentiating between "bot" and "human" categories. As we mentioned before, the model performs well, with many correct predictions (TP and TN) compared to incorrect predictions (FP and FN). True Positive (TP): 1008 - The model correctly predicted 1008 instances as "bot" when they were actually "bot". True Negative (TN): 1128 - The model accurately forecasts 1128 instances as "human" when they were actually "human". False Positive (FP): 54 - The model incorrectly predicted 54 instances as "human" when they were actually "bot". False Negative (FN): 37 - The model incorrectly predicted 37 instances as "bot" when they were actually "human".

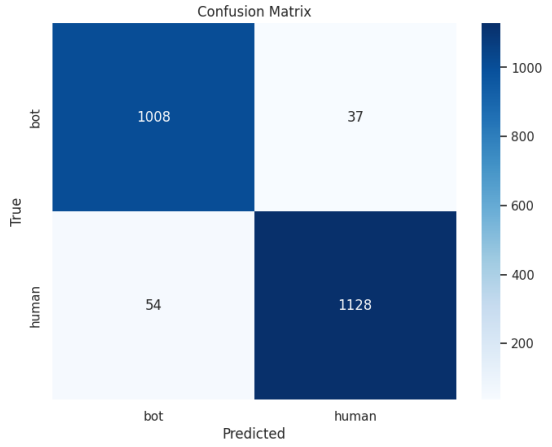

Figure 24: Confusion matrix visualization for *Tweepfake dataset*

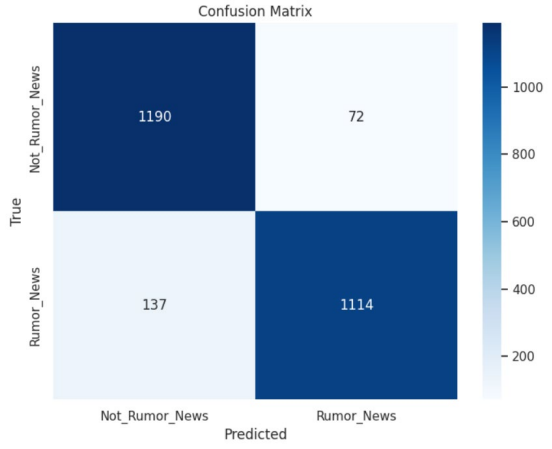

Figure 25: Confusion matrix visualization for *PHEME dataset*

D. Discussion

This research is founded on a systematic and reproducible tokenization pipeline specifically designed for the detection of fake news on short-text platforms, such as Twitter. In contrast to earlier studies that predominantly utilized basic tokenizers or general-purpose models, our methodology employs the sophisticated DeBERTa-v3 tokenizer, which has yet to see extensive application in this field. Importantly, we enhance the tokenization process for informal, noisy text by integrating dynamic padding and sequence truncation techniques that are specifically tailored for tweet-length inputs. Additionally, we organize the output as TensorFlow datasets, which allows for smooth integration with GPU-based training frameworks and improves training efficiency. These implementation decisions significantly impact both model performance and scalability, establishing our tokenization pipeline as a novel and practical advancement in real-world fake news classification systems.

Impact of Preprocessing Variability:

The influence of preprocessing on model performance was evaluated across multiple scenarios, with accuracy assessed under each condition:

Table 7: Accuracy Drop Under Inconsistent Preprocessing

Preprocessing Scenario	Accuracy	vs. Full Preprocessing
Full preprocessing (baseline)	97.12%	–
Partial preprocessing	94.1%	-3.02%
Noisy input	89.7%	-7.42%
Raw text	82.3%	-14.82%

As demonstrated in Table 7, the complete preprocessing (baseline) achieves the highest accuracy of 97.12%, thereby setting the standard for comparison.

In cases where only partial preprocessing is utilized, the accuracy decreases to 94.1%, indicating a reduction of 3.02%.

When dealing with noisy input—which may include misspellings, special characters, or distortions—the accuracy further diminishes to 89.7%, reflecting a 7.42% decline.

The most pronounced drop in performance is observed when raw text is employed without any preprocessing, leading to an accuracy of 82.3%, which represents a decrease of 14.82%.

E. COMPARATIVE RESULTS

The performance of existing literature in identifying AI-generated content varies, according to a comparative investigation. While BERTweet-based identification obtains a higher 94.7% on human-written text but drops to 80.2% on GPT-2 generated content, DeBERTa-based techniques show a strong accuracy of 91%. Particularly for GPT-2 detection, ensemble approaches provide a moderate accuracy of up to

84.4%. It's interesting to note that conventional machine learning methods that use sentiment features attain a competitive 86% accuracy, demonstrating the promise of less complex methods.

V. Future Research Directions

The future of sentiment analysis-based security threat prediction and deepfake text detection presents numerous study avenues. Researchers can improve the precision and efficiency of these systems by utilizing cutting-edge machine learning and deep learning methodologies. The following sections outline key areas for future exploration:

- **Integration of Advanced Models:**
  Deep Learning Architectures: Predicting cyberattacks and detecting deepfake content depend on the ability to recognize subtle feelings in text, which can be enhanced by using models like BERT and LSTM. [90] [34].
  Emotion Detection: Sentiment analysis and emotion detection together can improve prediction skills by offering deeper insights into the driving forces behind hostile messages [91].
- **Adversarial Robustness:**
  Vulnerability Assessment: Investigating the susceptibility of sentiment analysis models to adversarial attacks is essential. Understanding how these models can be misled will inform the development of more robust detection systems [92].
  Countermeasures: The dependability of sentiment analysis in security environments might be increased by investigating practical ways to lessen adversary attacks [92].
- **Cross-Platform Analysis:**
  Multi-Source Data Integration: Examining information from several sources, such as hacker forums and social media, can improve detection accuracy and offer a thorough picture of possible threats [34] [90].

Although these approaches show promise for improving security prediction and deepfake detection, there are still obstacles to overcome, especially in guaranteeing that models are resilient to hostile attacks and the moral ramifications of tracking online sentiment. To create safe and efficient systems, more research in these areas is essential.

VI. Conclusion

The field of deepfake text detection and security attack prediction continues to evolve rapidly. While current approaches show promising results, significant challenges remain in creating robust, scalable solutions. Future research should focus on improving model efficiency, reducing false positives, and developing more sophisticated ensemble approaches.

REFERENCES

[1] Alghamdi, a. M., pileggi, s. F., & sohaib, o. (2023). Social media analysis to enhance sustainable knowledge management: a concise literature review. Sustainability, 15(13), 9957.

[2] Osadola, o., amuta, e., somefun, c., somefun, t., ongbali, s., mene, j. (2024) deciphering disinformation: strategies for identifying and addressing fake news in today's information landscape 2024 international conference on science, engineering and business for driving sustainable development goals (seb4sdg), 1-7

[3] Aïmeur, e., amri, s., & brassard, g. (2023). Fake news, disinformation and misinformation in social media: a review. Social network analysis and mining, 13(1), 30.

[4] Dame adjin-tettey, t. (2022). Combating fake news, disinformation, and misinformation: experimental evidence for media literacy education. Cogent arts & humanities, 9(1), 2037229.

[5] Millière, r. (2022). Deep learning and synthetic media. *Synthese*, *200*(3), 231.

[6] Iparraguirre-villanueva, o., guevara-ponce, v., ruiz-alvarado, d., beltozar-clemente, s., sierra-liñan, f., zapata-paulini, j., & cabanillas-carbonell, m. (2023). Text prediction recurrent neural networks using long shortterm memory-dropout.

[7] Perez, a. R., & rivas, p. (2023). Combatting human trafficking in the cyberspace: a natural language processing-based methodology to analyze the language in online advertisements. Arxiv preprint arxiv:2311.13118.

[8] Hu, z., dychka, i., potapova, k., & meliukh, v. (2024). Augmenting sentiment analysis prediction in binary text classification through advanced natural language processing models and classifiers. Int. J. Inf. Technol. Comput. Sci, 16, 16-31.

[9] Lipton, z. C., berkowitz, j., & elkan, c. (2015). A critical review of recurrent neural networks for sequence learning. *Arxiv preprint arxiv:1506.00019*.

[10] Dumitru, r. G., peteleaza, d., & surdeanu, m. (2024). Enhancing transformer rnns with multiple temporal perspectives. Arxiv preprint arxiv:2402.02625.

[11] Baskaran, s., alagarsamy, s., selcia, s., & shivam, s. (2024, march). Text generation using long short-term memory. In *2024 third international conference on intelligent techniques in control, optimization and signal processing (incos)* (pp. 1-6). Ieee.

[12] Iparraguirre-villanueva, o., guevara-ponce, v., ruiz-alvarado, d., beltozar-clemente, s., sierra-liñan, f., zapata-paulini, j., & cabanillas-carbonell, m. (2023). Text prediction recurrent neural networks using long shortterm memory-dropout.

[13] Singh, b., kumar, a., kaur, s., shekhar, s., & singh, g. (2023, november). Exploring the effectiveness of various deep learning techniques for text generation in natural language processing. In *2023 international conference on advances in computation, communication and information technology (icaiccit)* (pp. 70-75). Ieee.

[14] Jiancong, zhu. (2024). Comparative study of sequence-to-sequence models: from rnns to transformers. Applied and computational engineering, 42(1):67-75. Doi: 10.54254/2755-2721/42/20230687.

[15] Naik, d., naik, i., & naik, n. (2024, july). Decoder-only transformers: the brains behind generative ai, large language models and large multimodal models. In *the international conference on computing, communication, cybersecurity & ai* (pp. 315-331). Cham: springer nature switzerland.

[16] Xu, h., bi, z., tseng, h. M., song, x., & feng, p. From transformers to the future: an in-depth exploration of modern language model architectures.

[17] Sajun, a. R., zualkernan, i., & sankalpa, d. (2024). A historical survey of advances in transformer architectures. *Applied sciences*, *14*(10), 4316.

[18] Alomari, e. A. (2024). Unlocking the potential: a comprehensive systematic review of chatgpt in natural language processing tasks. *Cmes-computer modeling in engineering & sciences*, *141*(1).

[19] Han, xu., zhuming, bi., hsien-cheng, tseng., xinyuan, song., peiyong, feng. (2024). From transformers to the future: an in-depth exploration of modern language model architectures. Doi: 10.31219/osf.io/n8r5j

[20] Raghuraj, singh. (2024). Advancements in natural language processing: an in-depth review of language transformer models. International journal for science technology and engineering, 12(6):1719-1732. Doi: 10.22214/ijraset.2024.63408

[21] Deldjoo, y., he, z., mcauley, j., korikov, a., sanner, s., ramisa, a., ... & ricci, f. (2024). Recommendation with generative models. *Arxiv preprint arxiv:2409.15173*.

[22] Sharkey, e., & treleaven, p. (2024). Bert vs gpt for financial engineering. *Arxiv preprint arxiv:2405.12990*.

[23] Salıcı, m., & ölçer, ü. E. (2024, september). Impact of transformer-based models in nlp: an in-depth study on bert and gpt. In *2024 8th international artificial intelligence and data processing symposium (idap)* (pp. 1-6). Ieee.

[24] Charpentier, l. G. G., & samuel, d. (2024). Gpt or bert: why not both?. *Arxiv preprint arxiv:2410.24159*.

[25] Alshannaq, f. B., shehab, m. M., al-assaf, a. H., alhenawi, e. A., & awawdeh, s. (2025). An exploration into the mechanisms and evolution of gpt models. In *impacts of generative ai on the future of research and education* (pp. 477-498). Igi global.

[26] Chen, j., wang, s., qi, z., zhang, z., wang, c., & zheng, h. (2024). A combined encoder and transformer approach for coherent and high-quality text generation. *Arxiv preprint arxiv:2411.12157*.

[27] Zhang, m. (2024, march). A comparative study on pre-trained models based on bert. In *2024 6th international conference on natural language processing (icnlp)* (pp. 326-330). Ieee.

[28] Pandey, r., waghela, h., rakshit, s., rangari, a., singh, a., kumar, r., ... & sen, j. (2024). Generative ai-based text generation methods using pre-trained gpt-2 model. *Arxiv preprint arxiv:2404.01786*.

[29] Blackburn, m. (2022, december). Multilingual social media text generation and evaluation with few-shot prompting. In *proceedings of the 2nd workshop on natural language generation, evaluation, and metrics (gem)* (pp. 417-427).

[30] Wang, z., wang, j., gu, h., su, f., & zhuang, b. (2018). Automatic conditional generation of personalized social media short texts. In *pricai 2018: trends in artificial intelligence: 15th pacific rim international conference on artificial intelligence, nanjing, china, august 28–31, 2018, proceedings, part ii 15* (pp. 56-63). Springer international publishing.

[31] Guo, y., & sarker, a. (2023, may). Socbert: a pretrained model for social media text. In *proceedings of the fourth workshop on insights from negative results in nlp* (pp. 45-52).

[32] Ma, r., gao, y., li, x., & yang, l. (2023, april). Research on automatic generation of social short text based on backtracking pattern. In *2023 asia-pacific conference on image processing, electronics and computers (ipec)* (pp. 336-347). Ieee.

[33] Gagandeep, & verma, j. (2024). Natural language processing for sentiment analysis in social media posts to identify suspicious behaviour. *Abhigyan*, 09702385241284879.

[34] Mardassa, b., beza, a., al madhan, a., & aldwairi, m. (2024, april). Sentiment analysis of hacker forums with deep learning to predict potential cyberattacks. In *2024 15th annual undergraduate research conference on applied computing (urc)* (pp. 1-6). Ieee.

[35] Liu, m., zhao, y., han, h., & zhang, j. (2024, may). Detecting potential malicious insiders based on sentiment profile. In *2024 3rd international joint conference on information and communication engineering (jcice)* (pp. 156-162). Ieee.

[36] Svitlana, lehominova., yurii, shchavinsky., tetiana, muzhanova., dmytro, rabchun., m., m., zaporozhchenko. (2023). Application of sentiment analysis to prevent cyberattacks on objects of critical information infrastructure. International journal of computing, doi: 10.47839/ijc.22.4.3362

[37] Olaoluwa, f., & potter, k. (2024). Natural language processing (nlp) for social media threat intelligence.

[38] Khanum, f., & lakshmi, p. S. (2024, october). Sentiment analysis using natural language processing, machine learning and deep learning. In *2024 5th international conference on circuits, control, communication and computing (i4c)* (pp. 113-118). Ieee.

[39] Dr., bhuvana, j. (2024). A study and development of application on sentiment analysis. Doi: 10.55041/isjem01354

[40] Kumar, n., agarwal, p., bansal, s., yadav, v. K., & bhowmik, d. (2024). Sentiment analysis using novel deep learning methods. *Proceedings on engineering sciences*, *6*(2), 853-862.

[41] Lingfeng, yu. (2024). Public opinion monitoring of sports stars based on text sentiment analysis. International journal of computer science and information technology, doi: 10.62051/ijcsit.v4n2.02

[42] Mao, h., jiang, y., lai, x., zhang, y., & huang, x. (2024). Enhancing public opinion monitoring for social hot events with a time series neural network-based logic map. *Measurement and control*, 00202940241284017.

[43] Xu, h. (2024). A bigura-based real time sentiment analysis of new media. *Peerj computer science*, *10*, e2069.

[44] Sulochana, b. C., pragada, b. S., kiran, b. C., reddy, g. A., & belwal, m. (2024). *Real-time crisis management: deep learning analysis on tweets from x*. 1–8. Https://doi.org/10.1109/icccnt61001.2024.10726184

[45] Kumar, a., & r, k. (2024). Real-time sentiment analysis system using the bert model. *International journal of innovative research in computer and communication engineering*. Https://doi.org/10.15680/ijircce.2024.1205114

[46] Nguyen, h. H. (2024). Enhancing sentiment analysis on social media data with advanced deep learning techniques. *International journal of advanced computer science & applications*, *15*(5).

[47] Fu, x. (2024). Transformer models in text summarization. *Applied and computational engineering*, *101*(1), 35–41. Https://doi.org/10.54254/2755-2721/101/20240946

[48] Liu, z., liu, p., sun, y., nie, z., zhang, x., zhang, y., ... & guo, t. (2024). Dia-bert: pre-trained end-to-end transformer models for enhanced dia proteomics data analysis. *Biorxiv*, 2024-11.

[49] Mswahili, m. E., & jeong, y. S. (2024). Transformer-based models for chemical smiles representation: a comprehensive literature review. *Heliyon*.

[50] Zhu, x., sun, s., lin, l., wu, y., & ma, x. (2024). Transformer-based approaches for neuroimaging: an in-depth review of their role in classification and regression tasks. *Reviews in the neurosciences*, (0).

[51] Anthony, q., michalowicz, b., hatef, j., xu, l., abduljabbai, m., shafi, a., ... & panda, d. K. (2024, august). Demystifying the communication characteristics for distributed transformer models. In *2024 ieee symposium on high-performance interconnects (hoti)* (pp. 57-65). Ieee.

[52] Altabaa, a., & lafferty, j. (2024). Disentangling and integrating relational and sensory information in transformer architectures. *Arxiv preprint arxiv:2405.16727*.

[53] Chien, j. T., & huang, y. H. (2024). Latent semantic and disentangled attention. *Ieee transactions on pattern analysis and machine intelligence*.

[54] Greco, c. M., & tagarelli, a. (2023). Bringing order into the realm of transformer-based language models for artificial intelligence and law. *Artificial intelligence and law*, 1-148.

[55] Pereira, g. A., & hussain, m. (2024). A review of transformer-based models for computer vision tasks: capturing global context and spatial relationships. *Arxiv preprint arxiv:2408.15178*.

[56] Jin, m., wen, q., liang, y., zhang, c., xue, s., wang, x., ... & xiong, h. (2023). Large models for time series and spatio-temporal data: a survey and outlook. *Arxiv preprint arxiv:2310.10196*.

[57] Moura, r., carvalho, j., plastino, a., & paes, a. (2024). Less is more: pruning bertweet architecture in twitter sentiment analysis. *Information processing & management*, *61*(4), 103688.

[58] Labeed, q., & liang, x. (2024, july). Multi-task learning transformers: comparative analysis for emotion classification and intensity prediction in social media. In *2024 14th international conference on pattern recognition systems (icprs)* (pp. 1-7). Ieee.

[59] Pokrywka, j., kaczmarek, j. I., & gorzelańczyk, e. J. (2024, december). Evaluating transformer models for suicide risk detection on social media. In *2024 ieee international conference on big data (bigdata)* (pp. 8566-8573). Ieee.

[60] Anggrainingsih, r., hassan, g. M., & datta, a. (2024). Transformer-based models for combating rumours on microblogging platforms: a review. *Artificial intelligence review*, *57*(8), 212.

[61] Shaikh, m. S., nirankari, l., pardeshi, v., sharma, r., & kale, prof. S. (2023). Deepfake detection using deep learning (cnn+lstm). *Indian scientific journal of research in engineering and management*. Https://doi.org/10.55041/ijsrem26808

[62] Sari, w. K., azhar, i. S. B., yamani, z., & florensia, y. (2024). Fake news detection using optimized convolutional neural network and bidirectional long short-term memory. *Computer engineering and applications journal*, *13*(03), 25-33.

[63] Lee, d., & moon, j. (2020). A method of detection of deepfake using bidirectional convolutional lstm. *Information security and cryptology*, *30*, 1053–1065. Https://doi.org/10.13089/jkiisc.2020.30.6.1053

[64] Sari, w. K., azhar, i. S. B., yamani, z., & florensia, y. (2024). Fake news detection using optimized convolutional neural network and bidirectional long short-term memory. *Computer engineering and applications journal*, *13*(03), 25-33.

[65] Tweepfake: about detecting deepfake tweets. (n.d.). *Arxiv*. Retrieved from https://arxiv.org/abs/2008.00036

[66] Tweepfake: about detecting deepfake tweets. (2021). *Plos one*. Retrieved from https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0251415

[67] pheme dataset for rumour detection and veracity classification. (2018). *Figshare*. Retrieved from https://figshare.com/articles/dataset/pheme_dataset_for_rumour_detection_and_veracity_classification/6392078

[68] Yadav, a., & vishwakarma, d. K. (2024). Aw-msa: adaptively weighted multi-scale attentional features for deepfake detection. *Engineering applications of artificial intelligence*, *127*, 107443.

[69] Hariprasad, y. (2024). *Enhancing cybersecurity and deepfake detection with advanced techniques*. Https://doi.org/10.31219/osf.io/jp6zv

[70] *Attention-emotion-enhanced convolutional lstm for sentiment analysis*. (2022). *33*(9), 4332–4345. Https://doi.org/10.1109/tnnls.2021.3056664

[71] Mardassa, b., beza, a., al madhan, a., & aldwairi, m. (2024, april). Sentiment analysis of hacker forums with deep learning to predict potential cyberattacks. In *2024 15th annual undergraduate research conference on applied computing (urc)* (pp. 1-6). Ieee.

[72] Kumar, a. A., priyangha, s. D., meghana, p., dheeraj, m., & aarthi, r. (2024, june). Xai–empowered ensemble deep learning for deepfake detection. In *2024 15th international conference on computing communication and networking technologies (icccnt)* (pp. 1-7). Ieee.

[73] Wagh, k., hindka, m., gopi, t., & ahmed, s. A. (2024). Ensemble machine learning method for detecting deep fakes in social platform. *Ictact journal on image & video processing*, *14*(3).

[74] Atas, s., & karakose, m. (2023, october). A new approach to in ensemble method for deepfake detection. In *2023 4th international conference on data analytics for business and industry (icdabi)* (pp. 201-204). Ieee.

[75] Nandy, h., & sridhar, r. (2021). A novel feature engineering approach for twitter-based text sentiment analysis. In *evolving technologies for computing, communication and smart world: proceedings of etccs 2020* (pp. 299-315). Springer singapore.

[76] Sıngh, s., kumar, k., & kumar, b. (2024). Analysis of feature extraction techniques for sentiment analysis of tweets. *Turkish journal of engineering*, *8*(4), 741-753.

[77] Arora, m., & kansal, v. (2019). Character level embedding with deep convolutional neural network for text normalization of unstructured data for twitter sentiment analysis. *Social network analysis and mining*, *9*(1), 12.

[78] Johal, s. K., & mohana, r. (2020). Effectiveness of normalization over processing of textual data using hybrid approach sentiment analysis. *International journal of grid and high performance computing (ijghpc)*, *12*(3), 43-56.

[79] Nazir, s., asif, m., rehman, m., & ahmad, s. (2024). Machine learning based framework for fine-grained word segmentation and enhanced text normalization for low resourced language. *Peerj computer science*, *10*, e1704.

[80] Liu, r. (2024). Exploring the impact of word2vec embeddings across neural network architectures for sentiment analysis. *Applied and computational engineering*, *97*(1), 93–98. Https://doi.org/10.54254/2755-2721/97/2024melb0085

[81] Ouchene, l., & bessou, s. (2023). *Fasttext embedding and lstm for sentiment analysis: an empirical study on algerian tweets*. 51–55. Https://doi.org/10.1109/icit58056.2023.10226060

[82] Illahi, i. R., & setiawan, e. B. (2024). Sentiment analysis on social media using fasttext feature expansion and recurrent neural network (rnn) with genetic algorithm optimization. *International journal on information and communication technology (ijoict)*, *10*(1), 78-89.

[83] Jia, z., bai, x., & pang, s. (2020). Hierarchical gated deep memory network with position-aware for aspect-based sentiment analysis. *Ieee access*, *8*, 136340–136347. Https://doi.org/10.1109/access.2020.3011318

[84] Jiang, t., wang, j., song, y., & rao, y. (2019). A position-aware transformation network for aspect-level sentiment classification. *International joint conference on neural network*, 1–8. Https://doi.org/10.1109/ijcnn.2019.8852474

[85] Chauhan, a., sharma, a., & mohana, r.m. (2023). A transformer model for end-to-end image and text aspect-based sentiment analysis. *2023 seventh international conference on image information processing (iciip)*, 277-282.

[86] Su, h., wang, x., li, j., xie, s., & luo, x. (2024). Enhanced implicit sentiment understanding with prototype learning and demonstration for aspect-based sentiment analysis. *Ieee transactions on computational social systems*.

[87] Chen, y., zhuang, t., & guo, k. (2021). Memory network with hierarchical multi-head attention for aspect-based sentiment analysis. *Applied intelligence, 51*, 4287 - 4304.

[88] Rajalakshmi, n.r., saravanan, s., & singha, a. (2023). Surplus data prediction and classification of textual-data using machine and deep learning comparative analysis. *2023 international conference on communication, security and artificial intelligence (iccsai)*, 329-334.

[89] https://www.kaggle.com/datasets/vivovinco/covid19-vs-vaccine-in-turkey

[90] Awasthi, a., bdair, m., kumar, a. N., thapa, s., & kumar, b. R. (2024, august). Nlp for sentiment analysis in social media posts to detect suspicious behaviour. In *2024 international conference on intelligent algorithms for computational intelligence systems (iacis)* (pp. 1-6). Ieee.

[91] Ghosh, a., pandey, n., & ashokkumar, c. (2024, june). Integrating emotion detection with sentiment analysis for enhanced text interpretation. In *2024 second international conference on inventive computing and informatics (icici)* (pp. 562-568). Ieee.

[92] Bajaj, a., & vishwakarma, d. K. (2023). Evading text based emotion detection mechanism via adversarial attacks. *Neurocomputing*, *558*, 126787.

[93] https://www.techtarget.com/whatis/definition/deepfake

[94] naseem, u., razzak, i., & eklund, p. W. (2021). A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter. *Multimedia tools and applications*, *80*, 35239-35266.

[95] rashid, s. M., mccusker, j. P., pinheiro, p., bax, m. P., santos, h. O., stingone, j. A., ... & mcguinness, d. L. (2020). The semantic data dictionary–an approach for describing and annotating data. *Data intelligence*, *2*(4), 443-486.

[96] he, p., liu, x., gao, j., & chen, w. (2020). Deberta: decoding-enhanced bert with disentangled attention. *Arxiv preprint arxiv:2006.03654*.

[97] he, p., gao, j., & chen, w. (2021). Debertav3: improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *Arxiv preprint arxiv:2111.09543*.

[98] hugging face transformers library documentation for deberta: deberta-v2

[99] tensorflow datasets documentation for creating and managing datasets: tensorflow datasets

[100] MICROSOFT RESEARCH. (2021). DEBERTA: DECODING-ENHANCED BERT WITH DISENTANGLED ATTENTION. HTTPS://ARXIV.ORG/ABS/2006.03654

[101] JADON, SHRUTI. "A SURVEY OF LOSS FUNCTIONS FOR SEMANTIC SEGMENTATION." 2020 IEEE CONFERENCE ON COMPUTATIONAL INTELLIGENCE IN BIOINFORMATICS AND COMPUTATIONAL BIOLOGY (CIBCB). IEEE, 2020.

[102] YACOUBY, REDA, AND DUSTIN AXMAN. "PROBABILISTIC EXTENSION OF PRECISION, RECALL, AND F1 SCORE FOR MORE THOROUGH EVALUATION OF CLASSIFICATION MODELS." PROCEEDINGS OF THE FIRST WORKSHOP ON EVALUATION AND COMPARISON OF NLP SYSTEMS. 2020.

[103] scikit-learn: scikit-learn: machine learning in python — scikit-learn 1.5.2 documentation

[104] tensorflow: introduction to tensorflow.
[105] pytorch: start locally | pytorch.
[106] li, c., xu, b., wu, g., zhuang, t., wang, x., & ge, w. (2014, may). improving word embeddings via combining with complementary languages. in *canadian conference on artificial intelligence* (pp. 313-318). cham: springer international publishing.