Detecting rumors in social media using Emotion based Deep Learning approach

MS(Research) Thesis

By

Drishti Sharma

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

June 2024

Detecting rumors in social media using Emotion based Deep Learning approach

A THESIS

submitted to the

INDIAN INSTITUTE OF TECHNOLOGY INDORE

in partial fulfillment of the requirements for

the award of the degree

of

MS(Research)

By

Drishti Sharma

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

June 2024

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis, titled **Detecting Rumors in social media using Emotion based Deep Learning approach**, is submitted in partial fulfillment of the requirements for the Master of Science (Research) degree in the Department of Computer Science and Engineering at the Indian Institute of Technology Indore. It presents the original work I conducted between August 2022 and April 2024.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

Drishti Sharma (03.06.24)

Signature of the Student with Date

(Drishti Sharma)

This is to certify that the above statement made by the candidate is correct to the best of

Ashish Srivastava

Signature of Thesis Supervisor with Date

(Prof. Abhishek Srivastava)

Drishti Sharma has successfully given his MS(Research) Oral Examination held on 08-July-2024

Signature of Head of Discipline
(Head Representative)
Date: 08-July-2024

Ashish Srivastava

Signature of Thesis Supervisor

Date:

ACKNOWLEDGEMENTS

I want to express my sincere gratitude to everyone who contributed significantly to this journey. First and foremost, I am deeply indebted to my supervisor, **Prof. Abhishek Srivastava**, for his unwavering inspiration throughout my research. This research work could not have been completed without his constant guidance and direction. His continuous support and encouragement have motivated me to remain streamlined in my research.

I am thankful to **Dr. Surya Prakash** and **Prof. Satyajit Chatterjee**, my research committee member, for taking some valuable time to evaluate my progress throughout the course. Their good comments and suggestions helped me to improve my work at various stages. I am also grateful to **Dr. Ranveer Singh**, HOD of Computer Science and Engineering Department and **Dr. Somnath Dey**, former HOD of the Computer Science and Engineering Department, for their help and support.

My sincere acknowledgement and respect to **Prof. Suhas Joshi** , Director, Indian Institute of Technology Indore for providing me the opportunity to explore my research capabilities at Indian Institute of Technology Indore.

I want to express my heartfelt respect to my parents, siblings, and friends for the love, care, and support they have provided to me throughout my life.

Finally, I am thankful to all who directly or indirectly contributed, helped, and supported me.

Drishti Sharma

Abstract

Social media has become a vital platform for information dissemination. However, this ease of sharing can also facilitate the spread of unverified and potentially damaging rumors, negatively affecting society and individuals. Given the vast amount of content generated on social media, there is a critical need for methods to assess information veracity and ensure factual accuracy. Existing research has investigated various approaches for rumor detection, including feature engineering and deep learning techniques, leveraging propagation theory to identify rumors. Our research builds upon this foundation by emphasizing the role of emotions and sentiment analysis in tweets, employing deep learning methods to enhance rumor detection accuracy. Leveraging insights from prior studies, a **S**entiment and **EM**otion driven **T**ransform**E**r **C**lassifier method (SEMTEC) is proposed. Unlike previous models, SEMTEC incorporates the extraction of emotional and sentiment tags alongside content-based information from the main tweet text. This comprehensive semantic analysis allows us to gauge user emotional states, leading to a remarkable improvement in accuracy in rumor detection. The proposed method is tested and compared with existing techniques on standard datasets and shown to be effective. This performance significantly surpasses that of existing state-of-the-art models.

List of Publications

1. **D. Sharma** and A. Srivastava. *Detecting rumors in social media using Emotion based Deep Learning approach*, PeerJ Computer Science (Accepted)

Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

SEMTEC Sentiment and Emotion driven Transformer Classifier

CNN Convolutional Neural Network

GCN Graph Convolutional Network

BERT Bidirectional Encoder Representations from Transformer

LSTM Long Short Term Memory

BiLSTM Bidirectional Long Short Term Memory

SVM Support Vector Machine

Chapter 1

Introduction

Social Media is undergoing a period of remarkable growth. Social networking platforms are deeply integrated into our daily lives now. Their applications extend across diverse sectors, including marketing, business development, entertainment, scientific research, governance, and so on. The influence of social media is indeed remarkable. It impacts important aspects of our lives, including but not limited to making important career decisions, financial behavior, social stature, and fashion trends. Nowadays, people tend to relate and assess everything based on how social media depicts it [1]. With a user population in the billions, as shown in Figure 1.1, these platforms have the potential to influence the conduct and actions of mankind significantly.

With the extent of influence of social media, it unsurprisingly falls prey to the circulation of rumors. A 'rumor' is a common phenomenon in our society and refers to information whose veracity is unconfirmed. Speculated and often unverified information is common on social media and potentially affects the well-being of individuals adversely [2]. Examples of the adverse effects of rumors on social media are many. The 2022 Republic Day celebrations in Delhi stand out as a grim reminder for all of us in India of the damage that rumors on social media can cause. Here, a peaceful procession following democratic norms fell prey to a few mischievous messages on social media and turned into a violent outbreak of street riots in Delhi. It is becoming increasingly common for perpetrators to use social media and foment violence, disorder, and fear in the common citizenry.

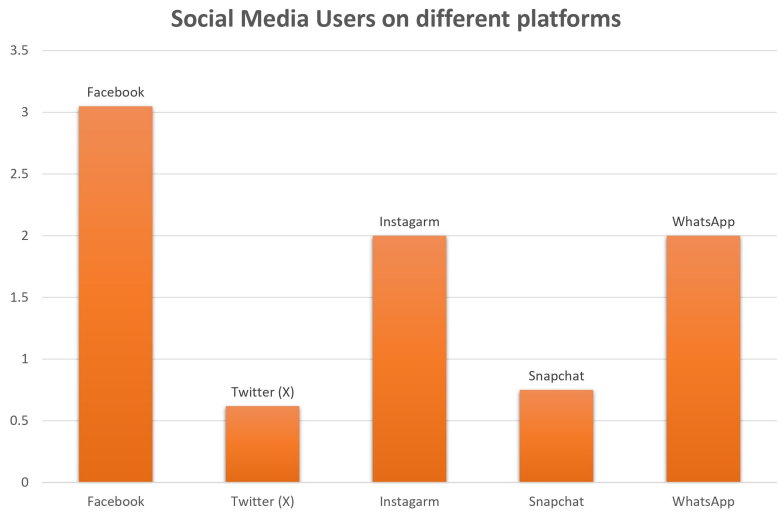

Figure 1.1: A schematic illustration of monthly active social media user around the world. The unit of number on y-axis is 1 unit = 1 billion.

Early detection of rumors on social media is, thus, incredibly crucial. Existing studies demonstrate the employment of diverse machine learning techniques like Naive Bayes, Random Forest, deep learning methodologies like Recurrent Neural Networks(RNN) [3], Recursive Neural Networks (RvNN) [4], and Graph Convolutional Networks(GCN) [5], to identify misinformation in data, over platforms such as Weibo and Twitter. Amongst these, deep learning methods demonstrate superior performance on tasks like classification and translation [6] that ultimately lead to better rumor detection. Of these, the GACL (Graph Adversarial Contrastive Learning) method formulates a loss function for better assessment of text; AFT (Adversarial Feature Transformation) takes an approach wherein it produces conflicting samples that facilitate better rumor detection [7]. RNN-based methods for rumor detection employ three recurrent units and learn the hidden representations that encapsulate variations in contextual information [3]. Feature fusion models with a fusion layer detect rumors by utilizing only a few labeled instances [8]. Further advancement in the direction of rumor detection led to the Bottom-up RvNN and Top-down RvNN methods that take into account the propagation layout of tweets [4] as well. This spurred a host of novel methodologies that place greater emphasis on comprehending the propagation patterns of rumors in addition to the contents of the respective

tweets. Methods that analyze timestamps of tweets carrying potential rumors, such as Credible Early Detection (CED), are also being employed nowadays for the timely identification of rumors [9].

An important aspect in rumor detection that has not received as much attention as it deserves relates to the emotions and sentiments expressed in a tweet. The emotional undertones of a statement provide insights into the writer's state of mind. Tweets with high emotional value tend to spread rapidly and are more likely to be rumors. A recent study shows the interconnectedness of fake news and sentiments [10]. We, therefore, choose to better understand and build upon the SAME (Sentiment-Aware Multimodal Embedding) model, which effectively harnesses latent sentiments in the text to detect fake news [11], in this work.

Our research prioritizes the investigation of the emotional and sentimental dimension of tweets shared online and their potential role in enhancing the accuracy of rumor detection. To the best of our understanding, our **Sentiment and EMotion driven TransformEr Classifier method (SEMTEC)**, assimilates context-based information and leverages emotions and sentiments from the textual modality, is pioneering in its approach to rumor detection by extensively considering these semantic attributes. To establish the validity of our research, we conducted comprehensive experiments over the publicly available "PHEME" dataset. Furthermore, we created a novel dataset named "Twitter24", which contains tweets from the social media platform "Twitter(X)". A manual verification process is employed to ensure the label assignment's accuracy. We utilize Boom Fact Check, a trusted fact-checking website that meticulously verifies the labels. Our **SEMTEC** model demonstrated exceptional performance, yielding an accuracy of approximately 92% on the "PHEME" and exceeds standard methods accuracy by around 2% on the "Twitter24" dataset.

Our SEMTEC model's key contributions to significance are summarized below:

- We present a novel dataset named "Twitter24", annotated with rumor and non-rumor labels. It consists of tweets extracted from a social media platform called Twitter, now X. We manually assign the labels after verifying them with a fact-checking website, i.e., Boom Fact Check. This establishes the correctness of

assigning labels to the tweets.

- We propose a novel emotion-based deep learning method named Sentiment and Emotion driven Transformer Classifier (SEMTEC) for rumor detection.
- SEMTEC leverages sentiment tags extracted from the available textual modality.
- The study incorporates an emotional aspect derived from a recurrent neural network (RNN)-based multilayer model, encompassing a diverse range of emotion classes.
- Extensive experimental analysis on the publicly accessible "PHEME" dataset and "Twitter24" dataset demonstrates that our proposed method, SEMTEC, addresses prior limitations and exhibits improved performance compared to existing models.
- We present a novel dataset named "EmoPHEME" annotated with emotion labels specifically designed to facilitate research in emotion extraction. This dataset offers researchers a valuable resource for training and evaluating their emotion extraction models. The original PHEME dataset solely focuses on rumor detection labels. This enriched dataset "EmoPHEME" is a byproduct of our work and opens up new avenues for emotion detection and analysis research.

Chapter 2

Literature Review

Identifying rumors has consistently been a widely studied issue, and researchers are striving to address it due to its direct impact on our society. A number of efforts have been made in understanding and detecting rumors [6, 9]. For the same, a number of methodologies have been employed, including those based on machine learning and deep learning. The researchers first treated rumor detection as a simple classification problem. To resolve the growing issue, machine learning-based approaches, including Naive Bayes, Random Forest, and SVM (Support Vector Machine), were utilized. Further improvement led to the utilization of deep learning-based approaches. As stated by Pattanaik et al.,[6], deep learning models outperform machine learning models. Keeping in mind the different approaches, we have divided this section into three subsections highlighting critical approaches utilized by prior research followed by the available opportunities or gaps in existing research. The subsections are mentioned below :

- Machine Learning Based Approaches
- Deep Learning Based Approaches
- Propagation-based Deep Learning Approaches
- Opportunities or Gaps in Existing Research

2.1 Machine Learning Based Approaches

This section exhibits the approaches based on machine learning for rumor detection. Earlier, when rumor detection was introduced as a threat to society, it was considered a simple classification problem, as Bingol et al.,[12] stated. His work assessed the performance of various supervised machine-learning methods. Bingol et al.,[12] have provided a comprehensive performance evaluation on methods OneR (One Rule), Naive Bayes, ZeroR, JRip, Sequential Minimal Optimization, and Hoeffding Tree. Similarly, Joulin et al.,[13] demonstrated the use of models like Naive Bayes, Logistic Regression, and Random Forest to identify rumor.

Furthermore, Joulin et al.,[13] worked on scaling linear machine learning classifiers to a large corpus with large output space. His work leveraged sentence representation as a bag of words followed by training a linear classifier like logistic regression. Our work demonstrates that scaling machine learning methods significantly improves classification tasks. Additionally, features also influence the performance of rumor detection and identification methods.

The prior research mainly focused on identifying the best algorithm for classifying whether the tweet was a rumor. This led to the classification of rumors using hot topic detection elaborated by Yang et al.,[14].

The hot topic detection technique combines multi-dimensional modeling of sentences with bursty term identification to detect emerging topics for rumor identification automatically. The author Yang et al.,[14] included a term weighting scheme that considers topicality and frequency properties of terms to detect the bursty terms. Named entities to represent the sentence were utilized by a new multidimensional sentence model unit of the architecture. Overall, the binary classifier leverages a set of features to identify sentences with rumor.

The inferences drawn from the prior research can be summarised as follows:

- Machine Learning methods perform well on simple classification tasks.
- Scaling of methods with significant features can significantly improve the machine learning models' performance on a large corpus.

- Rumor detection extends beyond simple classification problems. It requires a more profound understanding of linguistics and propagation features.

2.2 Deep Learning Based Approaches

This section demonstrates the related approaches previously employed utilizing the deep learning models. The emergence of deep learning technologies has significantly impacted the research fields due to their impressive performance. The extensive research done by Kumar et al.,[15] and Bian et al.,[5] establish evidence that deep learning methods are more effective in classifying rumor. Kumar et al.,[15] proposed a new way to represent conversations on social media as binarized constituency trees. The representation allowed feature comparison in the main tweet and its follow-up replies. The researchers utilized three LSTM units for pattern learning. LSTM units learn stance as well as a rumor, making it a multitask model, further followed by propagating essential stance in the signal form up in the tree for effective rumor classification of root node or tweet.

From the prior insights drawn, tree structures were helpful in learning patterns and extracting features out of the tweets. The work done by Ma et al., [3, 16, 4] follows a tree-like structure. The work[3] utilized Recurrent Neural Networks (RNN) for learning hidden features to get contextual information with time and tree-structured Recursive Neural Networks to find similarity in structure, respectively.

Deep Learning models like tree-structured Recursive Neural Networks proposed by Ma et al.,[4] leveraged discriminative features from content available in tweets by following non-sequential structure in order to generate representations helpful in identifying different rumors.

Furthermore, Feng et al.,[17] proposed a BiMGCL model utilizing the bi-directional graphs to structure the rumor events. The BiMGCL performs self-supervised contrastive learning to capture the propagation characteristics of rumor events.

The inferences drawn after studying deep learning approaches are mentioned below :

- The deep learning models achieved superior performance compared to machine

learning models.

- Provides functionality to deduce features like pattern learning and similarity index out of the tweets.

2.3 Propagation based Deep Learning Approaches

Propagation mode utilizes features related to the flow of rumor with deep learning techniques. Ma et al., [16] employed a propagation tree kernel structure to identify patterns between tree structures. Propagation trees provide clues on how a message propagates over time. Adding on a kernel-based method called Propagation Tree Kernel captures patterns that differentiate various rumors by assessing similarities in their tree-type structure.

Whereas researcher Sun et al.,[7] focused on extracting dissimilarity between features of transmitted information to detect rumor. Author Bian et al.,[5] explained that rumor propagation should also include dispersion. The work [5] proposed a BiGCN (Bidirectional Graph Convolutional Network) model that explored dispersion and propagation via top-down and bottom-up representation of rumor.

Furthermore, Wu et.al.,[18] utilized representation learning leveraging the propagation feature. Graphs were constructed following the general thread pattern (replies on the main tweet) on Twitter, followed by a gated graph neural network-based method called PGNN (Propagation Graph Neural Network) to generate powerful representations for each node. The PGNN updates the representation of nodes by information exchange between neighbor nodes within a fixed time. Based on this, Wu et al., [18] proposed GLO-PGNN (Global embedding with Graph Neural Network) and ENS-PGNN (Ensemble Graph Neural Network) for rumor detection.

The inferences drawn from prior research are mentioned below :

- The main post provides crucial content for rumor detection.

- While existing approaches for rumor detection emphasize various features, linguistic and semantic analysis have received comparatively less attention.

Table 2.1 summarizes the literature review, highlighting existing research and the proposed method.

Author	Method Name	Description
Joulin et al.[13]	FastText Classifier	Employs linear classifier following training. Focuses on utilizing machine learning methods for classification.
Bingol et al.[12]	ML Classifier	Considers rumor detection as simple classification problem. Utilizes standard machine learning classifiers.
Ma et al.[19]	GAN-GRU	Employs generator to introduce conflicting and uncertain perspective in original tweet.
Bian et al.[5]	BiGCN	Incorporates propagation by up-down GCN and dispersion via bottom-up GCN for rumor detection.
Lu et al.[20]	GCAN	Generates explanations highlighting the evidence from suspicious retweeters and the concerning words they use.
D. Sharma et al.	**SEMTEC (Proposed)**	**Establishes relationship between the semantic properties of the tweet with its veracity**

Table 2.1: Table summarizing the previous works and proposed method.

2.4 Opportunities in Existing Research/Research Gaps

Prior research indicates the usage of machine learning and deep learning models for rumor classification. The gaps that were identified are mentioned below :

- The emphasis of prior research was to deal with rumor detection as either a simple classification problem or extending it to utilize propagation for classification.
- The availability of relevant datasets was restricted to specific social media platforms and inaccessible to researchers globally.
- The Semantic aspect of the tweet, including emotions, sentiment, and contextual understanding, has not been given as much importance as it should receive.

The proposed work focuses more on analyzing the semantic aspect of the main tweet. The SEMTEC (Sentiment and Emotion driven Transformer Classifier) method utilizes textual features, sentiment, and emotion tags, extending the rumor detection task beyond a simple binary classification. The utilized dataset, comprised of English language tweets extracted from a globally accessible social networking platform Twitter(now X).

Chapter 3

Detecting Rumors in Social Media

This chapter discusses the proposed Sentiment and Emotion driven Transformer Classifier (SEMTEC) method for rumor detection and classification. The problem statement is clearly stated in Section 3.1, followed by the proposed methodology in Section 3.2. Subsequent sections provide detailed step-by-step descriptions of the methodology.

3.1 Problem Statement

In this work, we frame the problem of rumor detection as a binary classification problem Given a dataset D comprising N_t tweet, represented by $T = \{T_i\}_{i=1}^{N_t}$. For each tweet T_i, T_i^t represents the textual features, E_i represents the corresponding extracted emotion features, and S_i represents the sentiment features. From these, we need to predict L_i such that $L_i \in L$, where $L \in \{0, 1\}$ denoting rumor or non-rumor where $E_i \in \{anger, fear, joy, love, sadness, surprise\}$ and $S_i \in \{positive, negative, neutral\}$.

3.2 Methodology : SEMTEC Method

The proposed methodology primarily focuses on analyzing the semantic characteristics of a tweet and demonstrates how the implied emotions and sentiments contribute to classifying the tweet into specific categories. Ajao et al.[10] demonstrate that a text's emotional tone and sentiment play a significant role in determining its

truthfulness. For example, highly emotional text, such as that conveying fear or anger, is more readily accepted as true. In conformance with this, the proposed SEMTEC method leverages a tweet's emotional and sentiment aspects for rumor detection.
This section provides a comprehensive overview of the proposed model's architecture, its components, and the necessary steps to achieve the final label of the tweets. The

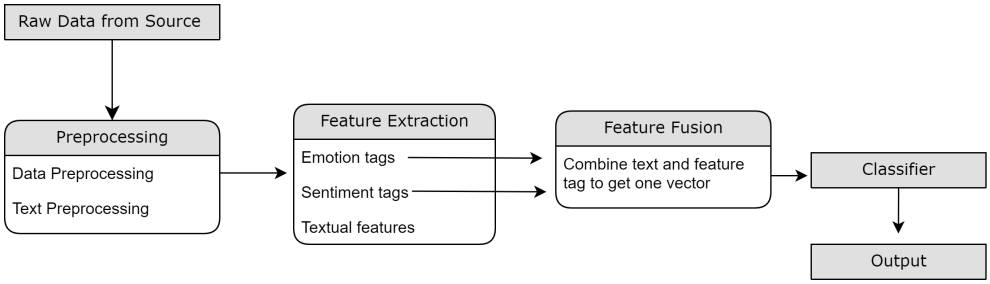

Figure 3.1: Illustration of flow of proposed methodology

subsequent sections include a component-wise explanation of the flow of the proposed method, as shown in Figure 3.1.

The overview of the proposed methodology is discussed below:

- The proposed methodology utilizes the textual modality of a tweet (message) on social media. To prepare the text for feature extraction, the textual modality needs to go through a pre-processing step, which involves cleaning the text and pre-processing.

- This process begins with pre-processing, followed by a feature extraction stage for emotion and sentiment assessment of the message. Section 3.2.2 discusses the semantic feature extraction stage.

- After acquiring the features mentioned above, their role in classifying a tweet as either a rumor or not is determined. The emotion and sentiment tags are concatenated with the textual modality to create a comprehensive feature representation.

- Finally, an encoder transformer module is utilized to extract the contextual information from the tweet, followed by a classifier to get the final label (rumor or non-rumor) corresponding to the tweet.

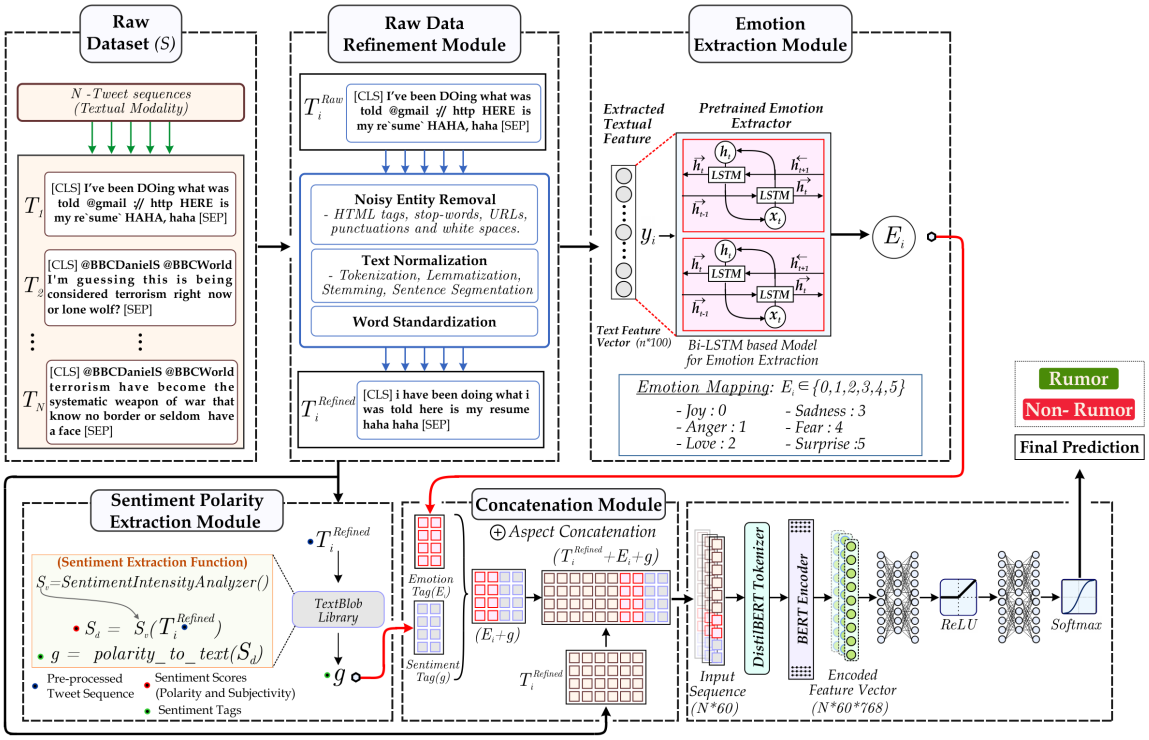

Figure 3.2: Image illustrates the proposed architecture. The method follows order as Raw Dataset Module, containing tweets fetched from source (Twitter) followed by Data Refinement Module, which involves cleaning and pre-processing. Furthermore, feature extraction utilizing Emotion Extraction Module and Sentiment Polarity Extraction module, is done. Finally, the textual modality is combined with features and provided as input to the classifier, to get final label.

Figure 3.2 illustrates the overall architecture of the Sentiment and Emotion driven Transformer Classifier (SEMTEC) model. The proposed method operates on textual modality, extracting data from social networking platforms like Twitter. The extracted data comprises irregularities that undergo processing before being passed on as input to the model. The data refinement module, as depicted in Figure 3.2, processes the data by removing ambiguities and inconsistencies and is explained in Section 3.2.1

3.2.1 Data Refinement Module

The rumor detection task involves working with textual data that largely represents the way people typically talk informally. The language of informal talk is full of inconsistencies and errors that need to be rectified. To do this, we employ the text

preprocessing toolkit "text_hammer" alongside a custom function to handle the text processing. The exercise of data refinement involves the following steps.

- Expansion of word contractions: Contractions are abbreviations or shortened forms of usually two words (or two parts of a word) that involve an apostrophe. To provide a consistent meaning of a statement to the model, these need to be expanded.

- Removal of emails, HTML tags, and special characters: Emails, HTML tags, and special characters increase the length of the text and can hinder the extraction of necessary information from textual data; hence, these are removed.

- Handling accented characters: As our model follows contextual-based learning, removal of accented characters, i.e., special symbols used to show a specific dialect or accent, helps in maintaining qualitative vocabulary corpus. Two examples of accented characters are résumé and naïve.

- Handling irregular capitalization: Proper capitalization facilitates the recognition of sentence tags such as nouns and pronouns, which leads to an easy analysis flow.

- Lowercasing: We perform lowercasing to maintain similarity and avoid additional vocabulary space for words with the same spelling. For example: 'Travel' and 'travel' have the same meaning but have different values when converted to vectors.

Figure 3.3 demonstrates the steps in the Data Refinement Module. The raw text passed as input undergoes expansion of contracted words, removal of HTTP and email tags, and conversion of accented characters to their original form. Finally, the text undergoes lower-casing to avoid unnecessary increases in the vocabulary corpus.

Subsequent to the preprocessing of the textual modality in the Data Refinement Module, the essential features are extracted to aid the rumor classification exercise as shown in Figure 3.2. This study primarily focuses on semantic features, i.e., emotion,

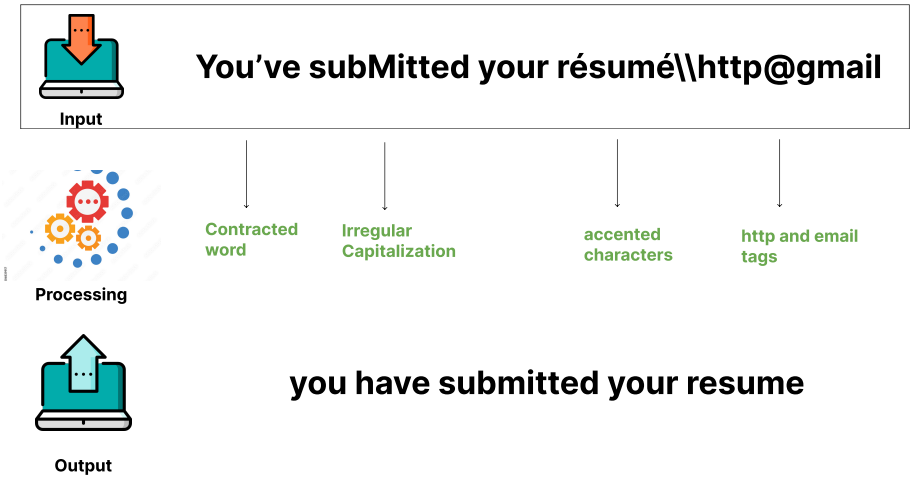

Figure 3.3: Image illustrates an example of the processing done by data refinement module.

sentiment tags, and textual features. The sentiment tags give an overall polarity to the text on whether the latter conveys a positive, negative, or neutral meaning. In contrast, emotions are more specific and complex behavioral aspects of the text, including specific classes for joy, sadness, surprise, love, anger, and fear. The feature extraction module is explained in detail in Section 3.2.2.

3.2.2 Feature Extraction

Features play a crucial role in enhancing the working of deep learning models for classification tasks as stated by Ma et. al.,[4]. The proposed SEMTEC method prioritizes the use of semantic features given their effectiveness in capturing the underlying meaning of tweets. Furthermore, the textual features leverage the contextual meaning from the textual modality aiding in rumor classification. This section details the feature extraction modules i.e. Sentiment Extraction Module and Emotion Extraction Module followed by textual feature extraction depicted in Figure 3.2. We employ various deep learning techniques to extract features from the textual data.

3.2.2.1 Sentiment Feature Extraction

Sentiments depict the overall undertone of the textual modality. Via sentiment tags, the acceptability of text by people can be determined. Ajao et. al.,[10] explain how sentiments are helpful in discriminating fake news. The extraction of sentiment features from textual modality provides significant contextual insights into a tweet. For extracting sentiment tags, we implement a module from the natural language processing toolkit, *TextBlob*. This module is pre-trained on a variety of datasets.
TextBlob leverages a lexicon-based sentiment analysis approach and initially determines the intensity (positive or negative orientation) of individual words in a sentence. Lexicon-based approaches involve the use of a pre-built dictionary that categorizes words as positive or negative. Having determined the intensity, TextBlob utilises two sentiment scores: polarity and subjectivity. The polarity is calculated by summing the polarity scores of each word in the text. This score ranges from -1 (completely negative) to $+1$ (completely positive). We do not use the subjectivity factor in our work. The generation of sentiment tags involves the utilization of a sentiment intensity analyzer as mentioned in Equation 3.1.

$$S_v = SentimentIntensityAnalyzer() \tag{3.1}$$

Subsequently, polarity scores are generated as shown in Equation 3.2 followed by estimation of tags from the calculated scores.

$$S_d = S_v.polarity\_scores(T_i) \tag{3.2}$$

The algorithm below outlines the procedure for sentiment tag extraction and subsequent incorporation of the same into the original textual features.

As per the algorithm, the Sentiment Feature Extraction Module takes processed text as input. Line 1 to 3 of the algorithm explain the process of analysing the sentiment labels via the Sentiment Intensity Analyser of the Textblob module, which is followed by the polarity score generation S_d. Finally, the polarity scores are analysed

Algorithm 3.1 Sentiment Feature Extraction Module

Input: $e_x : TextualModalityT_i$
Output: S_i: Sentiment Label
function Sentiment(e_x)

1: $S_v = IntensityAnalyzer()$
2: $S_d = S_v.polarity\_scores(T_i)$
3: $S_i = polarity\_to\_text(S_d)$
4: **return** S_i

to generate labels S_i where $S_i \in \{positive, negative, neutral\}$.

3.2.2.2 Emotion Feature Extraction

Emotions depict the psychological aspect of a writer's state of mind in a more refined way as compared to sentiments. Deb et.al.,[21] state in their work that false stories inspire fear, disgust, and surprise, while true stories inspire anticipation, sadness, joy, and trust.

Following the same, the novel framework proposed in this thesis invests in extracting the emotion tags from a textual modality. The emotion extraction module considers the following six emotion tags namely: joy, sadness, anger, fear, love, and surprise. To associate each tweet with an emotion, we employ a recurrent neural network (RNN)-based deep learning model. Leveraging the 'Emotions dataset for NLP', we classify each tweet into one of six pre-defined emotion categories based on its textual content.

For the extraction of emotion tags, we utilize the RNN based module namely Bi-directional LSTM. The Long Short-Term Memory (LSTM) network architecture leverages recurrently connected sub-networks, termed memory blocks. The memory block is designed to remember its state over time and control information flow using non-linear gating mechanisms. It consists of cell, input gate, output gate and forget gate [22]. Mathematically explaining, the recurrently connected blocks can be represented through functions. Below defined equations represent functions corresponding to the gates utilized in the working of LSTM units. The Equation 3.3 describes the forward pass limiting the input in the RNN network where the current input is say

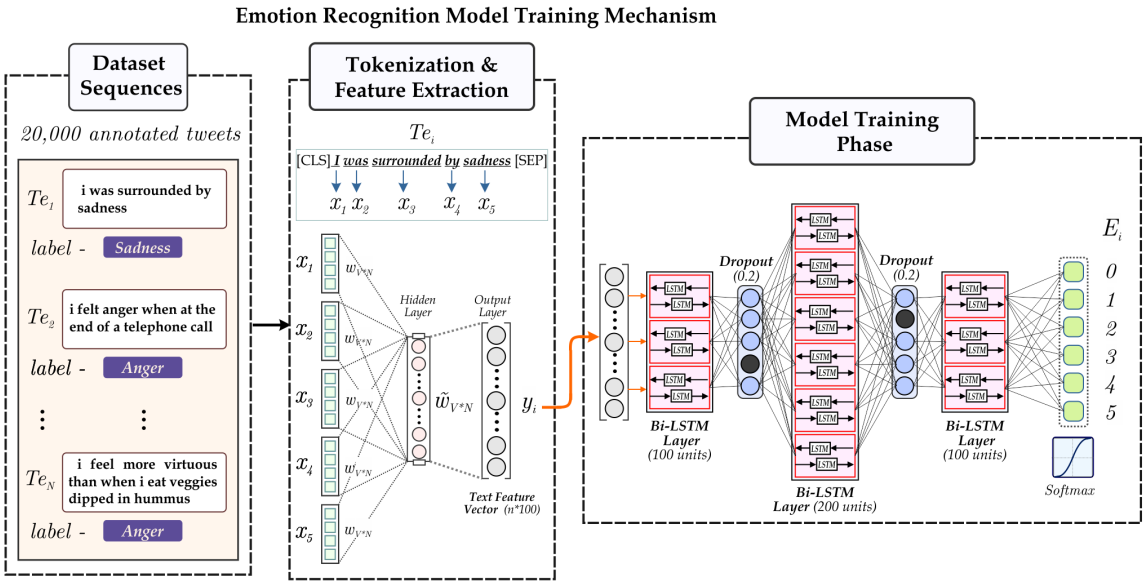

Figure 3.4: Illustration the RNN based emotion extraction module

x^t and z^{t-1} is output of LSTM in the last iteration. It represents the functionality of block input which focuses on updating the block input component.

$$y^{(t)} = \mathcal{F}(W_y x^t + R_y z^{t-1} + b_y) \tag{3.3}$$

The $\mathcal{F}$ in Equation 3.3 usually is *tanh* whereas W_y and R_y are weights associated with x^t and z^{t-1}. For the next connected block, the current input is combined with previous layer's output, as shown in Equation 3.4 followed by removal of information from the previous cell, i.e. $g^{(t)}$, following the same procedure as input gate on current input, previous cell output and the state c^{t-1}. The τ is always *sigmoid* and p, W, R are weights at respective stages. Equation 3.3 represents the working of input gate of recurrent unit.

$$i^{(t)} = \tau(W_i x^t + R_i z^{t-1} + p_i \odot c^{t-1} + b_i) \tag{3.4}$$

By combining the $y^{(t)}$, $i^{(t)}$ and $g^{(t)}$ we can calculate the cell value as $c^{(t)} = y^{(t)} \odot i^{(t)} + c^{(t-1)} \odot g^{(t)}$. Finally the output of the recurrent model i.e., the output gate can be

described as in Equation 3.5.

$$O^{(t)} = \tau(W_o x^t + R_o z^{t-1} + p_o \odot c^t + b_o) \tag{3.5}$$

The Equations 3.3, 3.4, 3.5 explains the mathematical working of the recurrent blocks utilized in our proposed work. Every equation represents the working of subunits of RNN blocks involving input gate, output gate. Van et.al [22] in his work states the in-depth working of recurrent blocks utilized in the proposed work.

Our emotion extraction module leverages two LSTM networks one in forward direction and other in backward direction, to capture the contextual insights of tweet in order to generate the output label. Figure 3.4 accordingly depicts the process employed for emotion extraction from the textual modality. The process initiates with passing the raw text T (representing individual tweet) as input to the model. T when tokenized gives T_1 which further provides us embedding vector by utilizing the GloVe model of gensim library.

GloVe stands for Global Vectors for word embedding. It is a pretrained model trained on large text data, utilizing this we get our embedding vectors for available textual modality[23].

Furthermore, the embeddding pass through bidirectional recurrent blocks of dimension $100, 200, 100$ as depicted in lines $3, 5, 7$ of the algorithm. Finally, the output of last bidirectional LSTM layer is passed through fully connected dense layer of dimension 6 followed by softmax activation function to get label E_i where $E_i \in joy, sadness, surprise, love, anger, fear$. The algorithm discussed below shows flow the model follows.

After extraction of semantic features i.e. emotion tags and sentiment tags, the textual modality is concatenated with the extracted features. Furthermore, textual features are extracted from modified textual modality as depicted in Section 3.2.2.3

Algorithm 3.2 Emotion extraction module

Input: T: Tweet from Emotion dataset
Output: E_{i} : Emotion label
function Emotion(T)

$T_1 \leftarrow clean\_text(T)$
$T_1 \leftarrow TokenizeEmbed(T)$
$g_1 \leftarrow Bi(LSTM(T_1, 100))$
$g_1 \leftarrow dropout(0.2)$
$g_2 \leftarrow Bi(LSTM(g_1, 200))$
$g_2 \leftarrow dropout(0.2)$
$C \leftarrow Bi(LSTM(g_2, 100))$
$E_{\text{i}} \leftarrow Dense(6, activation = \text{“}softmax\text{”})$
return E_{i}

3.2.2.3 Textual Feature Extraction

The proposed work focuses on textual data, acknowledging its supremacy in conveying meaning and context within social media posts. Therefore, to extract meaningful features from the textual content of tweets, we leverage the power of transformer-based deep learning models, specifically employing the well-established BERT (Bidirectional Encoder Representations from Transformers) architecture [24].

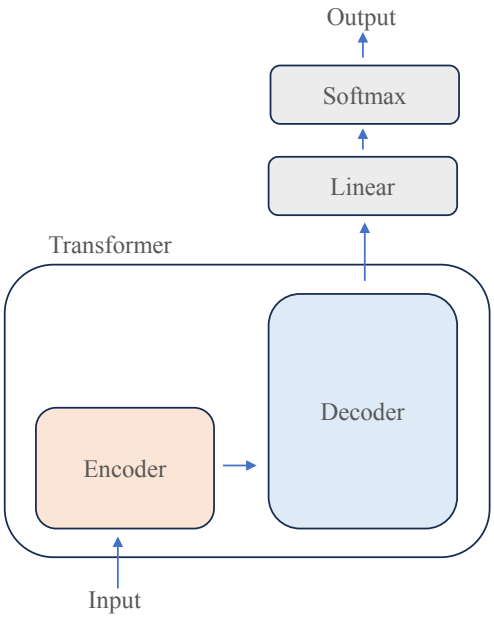

Figure 3.5: Architecture of standard transformer

The basic architecture of transformer is described in Figure 3.5. The transformer architecture consists of encoder and decoder components. The encoder generates distinct continuous vector representations by processing the text of a tweet. These vectorized embeddings are then utilized by the decoder to predict the desired outputs. The proposed work utilises BERT because it is pretrained on large data which sums

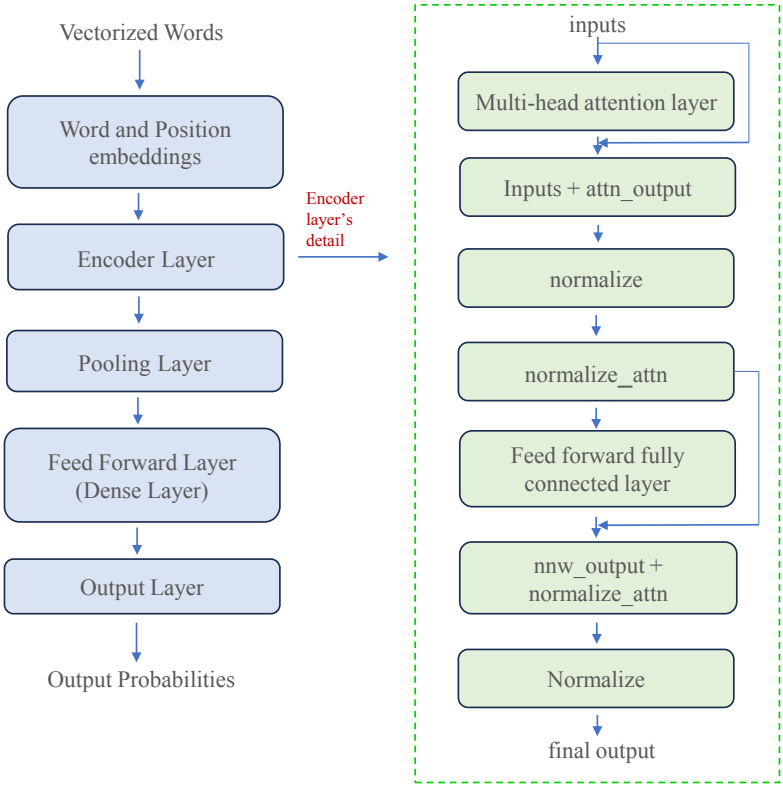

Figure 3.6: Architecture of transformer-based deep learning model for embedding generation

upto around 3.3 billion words from Wikipedia and BooksCorpus. The architecture of BERT is depicted in Figure 3.6. The model consists large number of encoder layers, feed forward network and attention heads.

The BERT language representation paradigm employs a deep architecture of stacked transformer encoder layers to generate contextualized word embeddings for each input token. Contextualized word embeddings are the numeric vectors generated after taking in account the context of text as well. As depicted in Figure 3.6, BERT captures the contextual meaning using the multi-head attention mechanism called heads[25]. It

consists of multiple heads executing in parallel to get broader range of relationship. Attention is simply the weighted average as stated by Jesse[1]. The encoder model provides vectors that can be utilized for specific task. Our proposed SEMTEC method leverages the vectors for classification task.

The mathematical explanation of encoder model i.e., BERT in our proposed SEMTEC method is explained in following manner. Our study represents each tweet as a sequence of words denoted as $Wd_i = \{w_i^x\}_{x=1}^{Z}$, where Z represents the word count in the tweet. These words sequence (Wd_i), forms the textual modality (T_i^t) for a tweet (T_i) . To create the embedding representation, we employ the distilBert-TokenizerFast model from the pre-trained transformer architecture. This tokenizer adds two special tokens, CLS (Class) at the beginning and SEP (Separator) at the end of each tweet's sequence. We use the distilBertTokenizerFast from the pre-trained transformer model.
For a given tweet(T_i), we provide the input (T_i^t). This input is then processed to generate a sequence of integer-based tokens D_t shown in Equation 3.6.

$$D_t = distilBertTokenizerFast(T_i^t) \tag{3.6}$$

For any tweet (T_i), the output is tokens and can be demonstrated as $D_t = \{d_i^x\}_{x=1}^{l}$, where l denotes the length of sequence. In our work, we are taking a fixed sequence length of 60 for each tweet, i.e., $l = 60$. Padding will be done for the tweets having length less than 60. Further, the tokens will be passed from encoder model to get the embedding vector for each token as shown in Equation 3.7.

$$E_m = BERT\,\{h_i^x\}_{x=1}^{l} \tag{3.7}$$

Here, $E_m = \{e_x\}_{x=1}^{d}$, where d is the dimension of size 768. The demonstrated process was textual feature extraction.

After the extraction of essential features, the final vector representation leveraging

[1]https://towardsdatascience.com/deconstructing-bert-part-2-visualizing-the-inner-workings-of-attention-60a16d86b5c1

the semantic features and textual features is passed through the classification module of the architecture as explained in Section 3.2.3, to get the required label as rumor or non-rumor.

3.2.3 Classification

This section illustrates the final step of our proposed SEMTEC method. The classification step is essential to get a final label corresponding to any tweet. The final module employs a neural network architecture as illustrated in Figure 3.2. The concatenated representation from the previous module serves as the input to the classifier, having text appended with emotion and sentiment tags, which subsequently generates the desired label. The encoded feature vector is of dimension 768 for each tweet. Furthermore, the vectors are passed through dense layers with ReLU between the layers to add non-linearity. The Equation 3.8 demonstrates the dense layer process.

$$z = Wx + b \tag{3.8}$$

Equation 3.8 represents the working of a dense layer, performing a linear transformation on the input data. The weight matrix W, depicts the significance of each input element, while the bias vector b, introduces activation among neurons. Within a neural network architecture, dense layers leverage a weighted linear combination of their inputs, augmented by a bias term, to generate a new representation of the incoming data, potentially enabling the network to extract more complex features or relationships. Lastly, the features (y_j) are passed through final dense layer of dimension 2, followed by a Softmax activation function to get the probabilities of the label of the tweet as demonstrated in Equation 3.9.

$$P = softmax_j(y_j) \tag{3.9}$$

Finally, the label can be procured by calculating the maximum of the probabilities denoted as P. The algorithm given below demonstrates the working of the classifier module.

Algorithm 3.3 Classification module

Input: T: Tweet with emotion and sentiment tag
Output: L_{i} : Rumor or Non-rumor label
function Classifier(T)

$T_1 \leftarrow clean\_text(T)$
$T_2 \leftarrow TokenizeEmbed(T_1)$
Define hidden layer activation function: $f(x) = \text{ReLU}(x)$
Function: Forward pass (x)
$\mathbf{z} = \mathbf{W}_1\mathbf{x} + \mathbf{b}_1$
$\mathbf{h} = f(\mathbf{z})$
$\mathbf{y} = \mathbf{W}_2\mathbf{h} + \mathbf{b}_2$
return y
Function: Predict class (T_2)
$\mathbf{y} = \text{Forward pass}(T_2)$
$i = \text{softmax}_j(\mathbf{y}_j)$
$L_i = \text{L}(\mathbf{i})$
return L_i

This way the proposed SEMTEC method achieves the task of rumor detection and classification.

Chapter 4

Experimentation and Result

In this section, we assess the effectiveness and precision of our model by conducting experiments on various features incorporated within it. Additionally, we demonstrate the model's efficacy using real-world data. This section encompasses the necessary setup details, parameter analysis, feature explanations, and a conclusive comparison. Online IDE's including Kaggle and Google Colab, were utilized for this work.

In this work, the data experimented with is textual data only. An overview of the nature of the utilized dataset is demostrated through illustrative examples in Table 4.1.

Tweet	**Label**
Now 10 dead in a shooting there today	Non-rumor
Charlie Hebdo became well known for publishing the Muhammed cartoons	Non-rumor

Table 4.1: The table illustrates the overview of the textual data and corresponding labels

4.0.1 Aggregation of Textual Data

We have utilized the publicly available "PHEME" dataset and a novel dataset named "Twitter24" in our work. Following subsections depicts the datasets description.

4.0.1.1 PHEME

The dataset is based on actual life incidents that happened around the world; the events are defined as hashtags, namely #charliehebdo, the incident of firing in France, and #ferguson, an incident of killing a black person in the USA. The dataset was formed consisting of a total of nine events. The tweets were taken from around 25,691 Twitter(X) users. This work utilizes the events mentioned above.

4.0.1.2 Twitter24

The novel dataset named "Twitter24" has been curated from the real-time tweets extracted manually from the social media platform Twitter(X). The dataset consists of only textual modality. It consists of tweets from popular user accounts like "Narendra Modi", "Virat Kohli", focusing more on information circulating in India. The labels are assigned manually and the correctness is established by utilizing fact checking website i.e., "Boom Fact Check". The purpose of this dataset is to validate the performance of SEMTEC model on real-time data.

Parameters	**PHEME**	**Twitter24**
Count of users	25,691	4,200
Count of tweets	62,445	4,829
Count of rumors	13,824	2,782
Count of non-rumors	48,619	2,043

Table 4.2: Parameter Statistics for PHEME and Twitter24 datasets

As referred in Table 4.2, the "PHEME" dataset consists of a total of 62,445 tweets and "Twitter24" consists around 4,829 tweets which are distributed in two labels, i.e., rumor and non-rumor. Three mutually exclusive training, testing, and validation sets are created from the tweets with tweet share as 70%, 20%, and 10%, respectively.

Tweet	**Label**
i was feeling a little vain when i did this one	sadness
i felt anger when at the end of a telephone call	anger

Table 4.3: Overview of "Emotion dataset for NLP" with textual data and labels

For training our RNN based deep learning module, "Emotion dataset for NLP" is

utilized. Table 4.3 illustrates an overview of the dataset. The dataset aggregates a total of $20,000$ tweets, categorized into six different classes namely joy, sadness, anger, fear, love, and surprise with a tweet count of $6,761$, $5,797$, $2,709$, $2,373$, $1,641$ and 719 respectively for each category.

4.0.2 Pre-processing the Dataset

This section presents the data pre-processing steps undertaken to address inconsistencies within the dataset and reduce the potential for erroneous outcomes in subsequent analyses.
The raw data from the dataset consists of redundancy and inconsistencies that need to be addressed. Equation 4.1 illustrates the removal of undefined values from the dataset denoted as D.

$$D_f = drop\_na(D) \tag{4.1}$$

Furthermore, duplicate redundancy can be removed using $drop\_duplicates()$.

4.0.3 Setup Requirements for Comparitive Analysis

This section includes a detailed description of the system and software requirements required to reproduce the results provided in this work. We present the specifications clearly and concisely using tables for easy reference. Leveraging the given parameters, the reproducibility of the mentioned results can be achieved.

4.0.3.1 Software Requirements

This section details the computational environment that facilitated the research and enabled the achievement of the presented results. Table 4.4 illustrates all the software parameters used for setting up the running environment of the proposed SEMTEC method.

Parameter	Value
IDE	Kaggle
Disk Space	73.1 GB
RAM (CPU)	30 GB
RAM (GPU)	15GB
GPU Type	Nvidia Tesla T4
Accelarator Count	02
Total RAM (with accelerator)	15 + 15 + 30
CPU	Intel Skylake/AMD/Broadwell
Count of CPU Cores	04

Table 4.4: Software Specifications for SEMTEC Method

4.0.3.2 Hardware Requirements

The following section details the hardware requirements used in this study to ensure the reproducibility of the presented results. We focus on the critical hardware components that significantly impact the performance of our experiments. Additionally, we acknowledge that similar configurations with comparable capabilities might achieve similar outcomes, aiming to broaden accessibility for researchers with varying resource constraints.

Parameter	Value
Device	Lenovo IdeaPad L340
Processor	Intel Core i7
Generation	9th Gen
Installed RAM	8.0 GB
Operating System	Windows
Edition	Windows 11
Disk Space	1 TB
SSD	256 GB

Table 4.5: Hardware Specifications for SEMTEC method

Table 4.5 demonstrates the specifications of the local system utilized in fulfillment of the proposed SEMTEC method.

4.0.4 Compared Methods

To evaluate the effectiveness of our proposed model, we compare its performance with existing methods. This section details the various methods employed in our experimentation. We compare our proposed SEMTEC method with different Deep Learning-based models such as FastText Classifier [13], GAN-GRU [19], TDRD [26], BiGCN [5], GCAN [20] and GACL [7].

4.0.4.1 FastText Classifier

FastText Classifier [13] represents text data as a bag of words and employs a linear classifier following training. This approach aligns with the principle of establishing simple machine learning models as strong baselines for text classification tasks.

4.0.4.2 GAN-GRU

The GAN-GRU [19] method is based on Generative Adversial Network(GAN). It employs a generator to introduce conflicting and uncertain perspectives into the original tweet thread, leading the discriminator to learn from more complicated examples.

4.0.4.3 TDRD

TDRD (Topic Driven Rumor Detection) method, extracts the topic of the post to derive the label for the tweet. Xu et.al.,[26] first automatically perform topic classification on source microblogs, and then they successfully incorporated the predicted topic vector of the source microblogs into rumor detection.

4.0.4.4 BiGCN

BiGCN (Bi-directional Graph Convolutional Network) [5] method utilizes both propagation and dispersion for rumor detection. The model incorporates both features by operating from bottom to top and top to bottom propagation of rumors. The up-down GCN (UD-GCN) incorporates propagation feature whereas bottom-up (BU-GCN) deals with dispersion of rumor.

4.0.4.5 GCAN

The GCAN (Graph Aware Co-attention Networks) [20], a neural network based method, predicts whether the tweet is true or not and simultaneously generates explanations that highlight the evidence from suspicious retweeters and the concerning words they use.

4.0.4.6 GACL

GACL (Graph Adversial Contrastive Learning) [7], deals with issues of poor generalization in conventional models, where the module of contrastive learning extracts similarities and differences among tweet threads. Furthermore, the AFT (Adversial Feature Transformation) module generates conflicting samples to extract event-invariant features.

Table 4.6 illustrates the comparison of features among the exiting research and the proposed SEMTEC method. The "X" indicates that feature is not utilized while "Y" indicates that corresponding feature is used in the mentioned work.

Method	**Contextual Analysis**	**Sentiment Tags**	**Emotion Tags**	**Propagation Feature**	**Text Classifier**
FastText	X	X	X	X	Y
GAN-GRU	Y	X	X	X	Y
TDRD	Y	X	X	X	Y
UDGCN	X	X	X	Y	Y
GCAN	Y	X	X	Y	Y
BiGCN	X	X	X	Y	Y
GACL	Y	X	X	X	Y
SEMTEC	**Y**	**Y**	**Y**	**X**	**Y**

Table 4.6: Illustrating feature breakdown in existing methods and proposed method for rumor detection task.

Existing rumor detection methods primarily rely on classification approaches, focusing on features extracted from follow-up comments to the initial tweet as indicated in Table 4.6. However, these methods often neglect the potential value of the main tweet itself for early rumor detection, particularly in real-time scenarios. This paper

introduces SEMTEC, a novel approach that moves beyond classification and emphasizes the importance of the main tweet. SEMTEC leverages a comprehensive feature set that incorporates functionalities employed in prior work, while also introducing additional features to enhance real-time detection accuracy.

4.0.5 Evaluation Metrics

This study evaluates the performance of compared approaches to measure their efficiency. We quantify the efficacy using specific metrics: Accuracy, F1-score, Recall and Precision correspondingly .

We define the precision with respect to a particular class where, $label \in \{rumor, non-rumor\}$, as the quotient of the number of correctly predicted instances of that label divided by the total number of predictions made for that label. This is mathematically represented in Equation 4.2.

$$Precision_{label} = \frac{True\_Predicted_{label}}{Total\_Predicted_{label}} \tag{4.2}$$

In the context of classification tasks, recall, serves as a crucial metric to assess the sensitivity of a classifier. We specifically measure the effectiveness of the classifier in identifying true positives. Recall ultimately quantifies the proportion of actual positive (rumor) instances that the classifier successfully classified correctly. We further formalize this in Equation 4.3.

$$Recall_{label} = \frac{True\_Predicted_{label}}{Total_{label}} \tag{4.3}$$

We leverage the F1-score metric for a combining precision and recall into a single, balanced measure. The F1-score is formulated as the harmonic mean of these two metrics in Equation 4.4. Through this we aim to provide a comprehensive evaluation of our classifier's performance, considering both its ability to correctly identify positive instances (precision) and its ability to avoid false negatives (recall) against the

considered exisitng approaches.

$$F1 - score_{label} = \frac{2 \times Precision_{label} \times Recall_{label}}{Precision_{label} + Recall_{label}} \tag{4.4}$$

Accuracy is a metric to state the overall performance of the mode. In our work, accuracy can be stated as the average precision calculated for available labels.

4.0.6 Results

This section discusses the results of the experiments conducted to evaluate the proposed model. We compare the performance of the proposed model against existing methods to assess its effectiveness. The comparison is further depicted to illustrate the relative performance of each method. The values presented are in the range of 0 to 1 and the parameters calculated have results as per every class of categorization, namely Rumor(R) and Non-rumor(N).

Model	Precision		Recall		F1-score		Accuracy
	R	N	R	N	R	N	
FastText	0.00	0.66	0.00	**1.00**	0.00	0.79	0.66
GAN-GRU	0.77	0.79	0.79	0.76	0.78	0.77	0.78
TDRD	0.81	0.83	0.63	0.92	0.71	0.87	0.82
UDGCN	0.75	0.83	0.67	0.87	0.70	0.85	0.80
GCAN	0.76	0.87	0.75	0.87	0.76	0.87	0.83
BiGCN	0.75	0.86	0.73	0.87	0.74	0.86	0.82
GACL	0.80	0.87	0.75	0.90	0.77	0.88	0.85
SEMTEC	**0.91**	**0.92**	**0.94**	0.90	**0.93**	**0.91**	**0.92**

Table 4.7: Effectiveness Comparision results from exiting methods on "PHEME" dataset

4.0.6.1 Effectiveness Comparisions

To evaluate the efficacy of our proposed approach for rumor detection, we compare its performance with existing techniques. We investigate the performance of various techniques by evaluating them based on established metrics like precision, recall, F1-score, and accuracy. Leveraging the publicly available "PHEME" dataset, we illustrate

the effectiveness of our model by comparing its results to those obtained using previously employed techniques. Table 4.7 shows how significantly better our SEMTEC

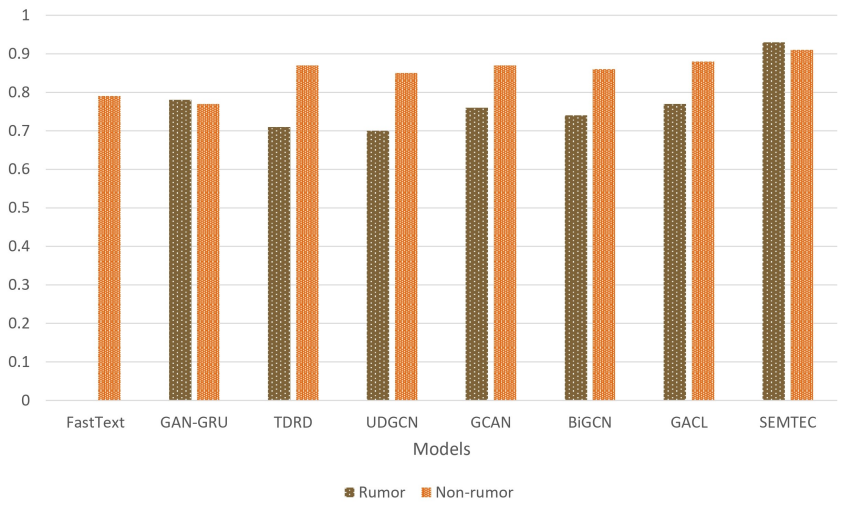

Figure 4.1: Illustration of Accuracy Comparison for Diverse Models

model performs on the aforementioned dataset than existing techniques. The performance of our SEMTEC model on different metric parameters namely Precision, Recall and F1-score is 0.91, 0.92 and 0.92 repectively. In terms of accuracy, we achieve a surge of around 0.7 when compared with the best performing existing method. Figure 4.1 illustrates a comprehensive visualization of the variations in F1-score across different models.

Model	Precision		Recall		F1-score		Accuracy
	R	N	R	N	R	N	
Naive Bayes (NB)	0.72	0.75	0.66	0.80	0.68	0.77	0.74
Random Forest (RF)	0.68	0.74	0.65	0.77	0.67	0.75	0.72
Support Vector (SVM)	0.74	0.76	0.66	0.82	0.69	0.79	0.75
BiLSTM	0.68	0.75	0.67	0.76	0.66	0.75	0.72
Transformer	0.67	0.74	0.66	0.76	0.67	0.73	0.71
SEMTEC	**0.91**	**0.92**	**0.94**	**0.90**	**0.93**	**0.91**	**0.92**

Table 4.8: Effectiveness Comparision of SEMTEC on PHEME dataset with standard classifiers

Our model achieves superior performance due to its incorporation of both emo-

tion and sentiment features alongside a contextual analysis of textual modalities, as opposed to current techniques, which rely on the textual content of social media posts.

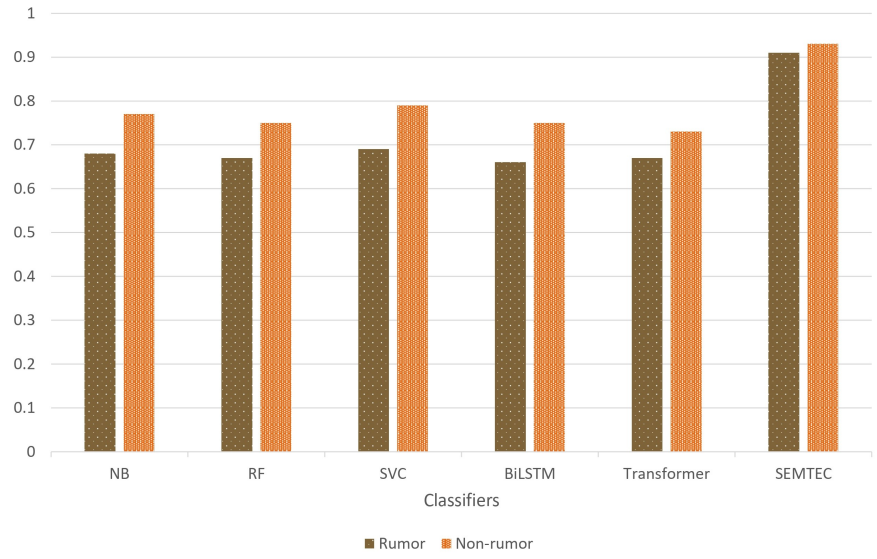

Figure 4.2: Illustration of Accuracy Comparison for Standard Classifiers on PHEME dataset

Furthermore, we compare our work with existing classifiers. The classifiers can be divided into two classes, Machine Learning based classifiers like Support Vector Machine (SVM), Random Forest (RF) and Deep Learning based classifiers like Transformers. As illustrated in Table 4.8, the SEMTEC method outperforms the standard classifiers by a significant margin.

Figure 4.2 visually illustrates the performances of the standard classifiers on the "PHEME" dataset. Our findings suggest that rumor detection extends beyond a simple classification task. To validate the performance of our proposed SEMTEC method, we further experimented with the novel "Twitter24" dataset. The experimentation demonstrates that the proposed SEMTEC method surpasses the existing standard methods used for classification by around 2%. Table 4.9 illustrates the findings highlighting the superior performances.

Model	Precision		Recall		F1-score		Accuracy
	R	N	R	N	R	N	
Naive Bayes (NB)	0.90	0.84	0.87	0.87	0.89	0.85	0.87
Random Forest (RF)	0.88	0.87	0.90	0.84	0.89	0.86	0.88
Support Vector (SVC)	0.90	0.92	0.95	0.86	0.92	0.89	0.91
BiLSTM	0.86	0.91	0.94	0.80	0.90	0.85	0.88
Transformer	0.88	0.84	0.80	0.85	0.88	0.85	0.87
SEMTEC	**0.92**	0.92	0.95	**0.89**	**0.94**	**0.91**	**0.93**

Table 4.9: Effectiveness Comparision of SEMTEC on Twitter24 dataset with standard classifiers.

4.0.6.2 Performance Gain Analysis

This section analyzes the effectiveness of our SEMTEC method with combinations of various features utilized in our work. The performance of our model is the outcome of integration of emotion features, sentiment features and contextual analysis of text. Ajao et.al., [10] depicted the interconnectedness of semantic features with the detection of fake news. Figure 4.3 demonstrates the performance of our method with regard to inclusion of various features. In Figure 4.3, we discuss the performance of the proposed model regarding the inclusion of emotion tags, sentiment tags, and textual modality. The emotion feature directly conveys the tweet's objective. This variant is illustrated

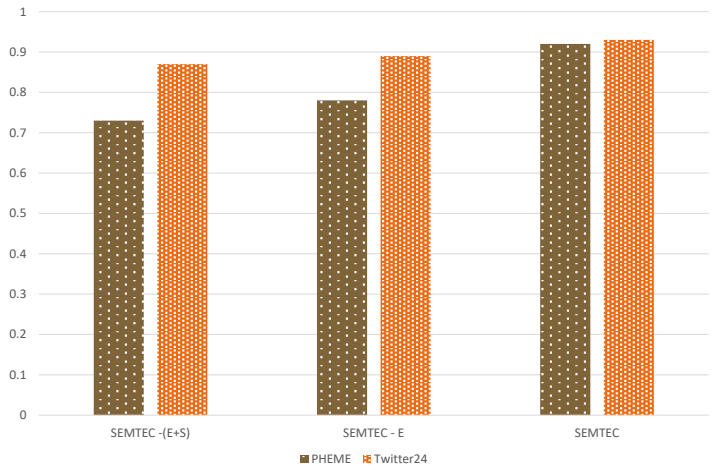

Figure 4.3: Illustration of performance of SEMTEC for various variants on PHEME and Twitter24 datasets

as SEMTEC. This model performs better than the prior SEMTEC - (E), where only

text along with sentiment tag, was used. SEMTEC - (E+S) illustrates the textual modality without any features. This work presents the results achieved in terms of accuracy. This enhancement can be attributed to the incorporation of emotion and sentiment tags as they facilitate a deeper understanding of the sentence semantics, which further prove significant in predicting the label of the post.

4.0.6.3 Analysis of Curated Dataset with Emotion Labels

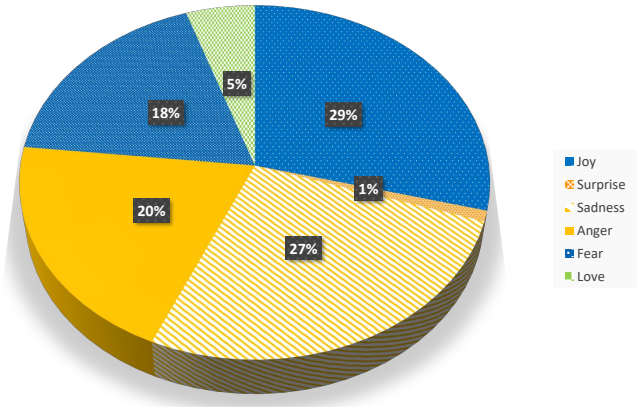

Figure 4.4: Distribution of Emotion lables on curated "EmoPHEME" dataset

In this section, we discuss the curated "EmoPHEME" dataset. The proposed SEMTEC method utilizes emotion tags, extracted using the RNN based deep learning emotion module. This module, trained on the "Emotion dataset for NLP", enables the generation of emotion labels for the "PHEME" dataset, capturing the emotional aspect of the tweets. The emotion extraction module facilitates the exploration of new information dimensions within the "EmoPHEME" dataset. This enhanced dataset offers potential applications in both training and testing models designed for emotion-related sentiment analysis tasks.

Figure 4.4 visualizes the distribution of the emotion labels across the "EmoPHEME" dataset. The labels, namely joy, anger, sadness, love, surprise and fear have percentage divisions as 29%, 20%, 27%, 5%, 1% and 18%, of the total tweets respectively.

Chapter 5

Conclusion and Future Work

This chapter details the observations and outcomes that conclude from the presented work, the aspects of extension of the proposed SEMTEC method, and concluding remarks. Subsequent sections encapsulate detailed descriptions.

5.1 Observations and Outcomes

This study introduces the SEMTEC approach, which investigates the relationship between semantic variables and their efficacy in rumor identification. It underlines the possibility of using emotion and sentiment analysis to improve the performance of existing approaches. This approach recognizes the emotional component of social media conversation, with sentiment analysis capturing the overall tone of textual data. Sentiment tags can help determine the public's receptivity to information. Emotions depict the psychological aspect of a writer's state of mind more refinedly than sentiments.

This distinction is crucial, as emotions provide a more precise representation of the writer's state, directly impacting how the audience perceives the message. Notably, the findings suggest a more significant influence of emotional features than sentiment features in rumor detection, as emotions map more directly to audience perception.

5.2 Future Work

This section focuses on the possible extensions of the proposed rumor detection endeavour. Subsequent subsections illustrate the aspects of further research.

5.2.0.1 Semantic Web approach for Rumor Detection

The proposed SEMTEC method for detecting rumors can be extended to utilize semantic web technologies to identify rumors in real-time. Semantic Web is an ontology-based approach that utilizes queries to address problems. "Semantic" refers to machine-readable data, while "web" signifies interconnected objects mapped to resources using URIs. In simple terms, the semantic web represents an extension of the World Wide Web (WWW) that furnishes software with metadata pertaining to published information.
We plan to utilize the knowledge graph representation of platforms like DBpedia and Wikidata via the semantic web to extend the proposed work. We can access the articles in real-time through SPARQL queries and help validate the tweet's label.

5.2.0.2 Rumor Detection on Low Resource Language

The extension of this proposed work acknowledges another limitation of the current work i.e., prioritizing high resource language. This research focuses on the development of a dedicated rumor detection model for the Hindi language, a widely spoken language in India despite the nation's multilingual landscape. To facilitate this, we are in the process of constructing a dataset by leveraging Hindi tweets retrieved from the social media platform X (formerly Twitter). The dataset is being annotated using established fact-checking websites. Furthermore, we are trying to build models leveraging emojis and essential features for rumor detection.

5.3 Conclusion

This study introduces SEMTEC, a novel deep learning-based method for rumor detection in social media. SEMTEC leverages sentiment and emotion analysis to enhance rumor classification accuracy.

The proposed approach utilizes a recurrent neural network (RNN) for extracting emotional features from tweets. Sentiment analysis is performed using the pre-trained TextBlob library. The extracted sentiment and emotion tags are then concatenated with the original tweet content. Following pre-processing of the textual data from the dataset, an encoder module extracts contextual features from the cleaned text. These contextual features, along with the sentiment and emotional features, are subsequently fed into a deep learning model for rumor classification.

The effectiveness of the proposed method SEMTEC is assessed using a social media dataset consisting of English tweets acquired from Twitter. The experiment results demonstrate that SEMTEC outperforms existing methods in terms of rumour detection efficacy. While the approach does not currently analyze the propagation patterns of rumors, it presents an opportunity for future exploration.

Future work includes investigating the integration of the semantic web and knowledge graphs from platforms like Wikidata and DBpedia. This ontology-based approach could enable real-time access to articles for rumor verification. Additionally, we aim to extend our work to encompass rumor detection in languages predominantly used in India, such as Hindi.

Bibliography

[1] J. H. Heinrichs, J.-S. Lim, and K.-S. Lim, "Influence of social networking site and user access method on social media evaluation," *Journal of Consumer Behaviour*, vol. 10, no. 6, pp. 347–355, 2011.

[2] G. W. Allport and L. Postman, "The psychology of rumor." 1947.

[3] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," 2016.

[4] J. Ma, W. Gao, and K.-F. Wong, "Rumor detection on twitter with tree-structured recursive neural networks." Association for Computational Linguistics, 2018.

[5] T. Bian, X. Xiao, T. Xu, P. Zhao, W. Huang, Y. Rong, and J. Huang, "Rumor detection on social media with bi-directional graph convolutional networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 549–556.

[6] B. Pattanaik, S. Mandal, and R. M. Tripathy, "A survey on rumor detection and prevention in social media using deep learning," *Knowledge and Information Systems*, pp. 1–42, 2023.

[7] T. Sun, Z. Qian, S. Dong, P. Li, and Q. Zhu, "Rumor detection on social media with graph adversarial contrastive learning," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 2789–2797.

[8] H.-y. Lu, C. Fan, X. Song, and W. Fang, "A novel few-shot learning based multi-modality fusion model for covid-19 rumor detection from online social media," *PeerJ Computer Science*, vol. 7, p. e688, 2021.

[9] C. Song, C. Yang, H. Chen, C. Tu, Z. Liu, and M. Sun, "Ced: credible early detection of social media rumors," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 8, pp. 3035–3047, 2019.

[10] O. Ajao, D. Bhowmik, and S. Zargari, "Sentiment aware fake news detection on online social networks," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2507–2511.

[11] L. Cui, S. Wang, and D. Lee, "Same: sentiment-aware multi-modal embedding for detecting fake news," in *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*, 2019, pp. 41–48.

[12] H. Bingol and B. Alatas, "Rumor detection in social media using machine learning methods," in *2019 1st International Informatics and Software Engineering Conference (UBMYK)*. IEEE, 2019, pp. 1–4.

[13] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016.

[14] Z. Yang, C. Wang, F. Zhang, Y. Zhang, and H. Zhang, "Emerging rumor identification for social media with hot topic detection," in *2015 12th web information system and application conference (WISA)*. IEEE, 2015, pp. 53–58.

[15] S. Kumar and K. M. Carley, "Tree lstms with convolution units to predict stance and rumor veracity in social media conversations," in *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2019, pp. 5047–5058.

[16] J. Ma, W. Gao, and K.-F. Wong, "Detect rumors in microblog posts using propagation structure via kernel learning." Association for Computational Linguistics, 2017.

[17] W. Feng, Y. Li, B. Li, Z. Jia, and Z. Chu, "Bimgcl: rumor detection via bi-directional multi-level graph contrastive learning," *PeerJ Computer Science*, vol. 9, p. e1659, 2023.

[18] Z. Wu, D. Pi, J. Chen, M. Xie, and J. Cao, "Rumor detection based on propagation graph neural network with attention mechanism," *Expert systems with applications*, vol. 158, p. 113595, 2020.

[19] J. Ma, W. Gao, and K.-F. Wong, "Detect rumors on twitter by promoting information campaigns with generative adversarial learning," in *The world wide web conference*, 2019, pp. 3049–3055.

[20] Y.-J. Lu and C.-T. Li, "GCAN: Graph-aware co-attention networks for explainable fake news detection on social media," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 505–514. [Online]. Available: https://aclanthology.org/2020.acl-main.48

[21] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aap9559

[22] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artificial Intelligence Review*, vol. 53, pp. 5929–5955, 2020.

[23] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[24] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[26] F. Xu, V. S. Sheng, and M. Wang, "Near real-time topic-driven rumor detection in source microblogs," *Knowledge-Based Systems*, vol. 207, p. 106391, 2020.