BRaG: a hybrid multi-feature framework for fake news detection on social media

Razieh Chalehchaleh, Mostafa Salehi, Reza Farahbakhsh, Noel Crespi

▶ **To cite this version:**

Razieh Chalehchaleh, Mostafa Salehi, Reza Farahbakhsh, Noel Crespi. BRaG: a hybrid multi-feature framework for fake news detection on social media. Social Network Analysis and Mining, 2024, 14 (35), pp.50. 10.1007/s13278-023-01185-7 . hal-04533937

HAL Id: hal-04533937
https://hal.science/hal-04533937

Submitted on 5 Apr 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

BRaG: A Hybrid Multi-Feature Framework for Fake News Detection on Social Media

Razieh Chalehchaleh[1,2*], Mostafa Salehi[2], Reza Farahbakhsh[1] and Noel Crespi[1]

[1] *Telecom SudParis, Institut Polytechnique de Paris, Palaiseau, France.*
[2] *Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran.*

*Corresponding author(s). E-mail(s):
razieh.chalehchaleh@telecom-sudparis.eu;
Contributing authors: mostafa_salehi@ut.ac.ir;
reza.farahbakhsh@telecom-sudparis.eu;
noel.crespi@telecom-sudparis.eu;

Abstract

Social media has gained immense popularity for its convenience, affordability, and interactive features. However, the characteristics that make social media platforms appealing also provide a fertile ground for the spread of fake news - deliberately misleading and unverifiable information that can have severe consequences for individuals and society. Previous approaches for detecting fake news have mostly focused on single aspects such as text, but are inadequate as fake news evolves to closely resemble genuine news. To enhance fake news detection, a comprehensive multi-faceted approach is necessary. Various machine-learning techniques have been used to detect fake news. This paper introduces a novel hybrid and multi-feature framework for detecting fake news that considers both the content (e.g., text) and context (e.g., user profiles and propagation graph) of news. Our framework, **BRaG**, leverages a combination of the **B**ERT pre-trained language model, recurrent neural network (**R**NN), and graph neural network (**G**NN) to analyze news text, sequence of engaged users, and the estimated news propagation graph, respectively and form the final news representation vector. Additionally, our approach incorporates text emoji meanings to take into account the contextual information they convey. The proposed

framework is evaluated on two real-world datasets and outperforms existing baselines and state-of-the-art fake news detection models.

Keywords: Fake News Detection, Social Media, Graph Neural Networks, Recurrent Neural Networks, Pre-Trained Language Models, News Content and Context Features

1 Introduction

The widespread use of social media has led to a shift in the way people consume news, with many choosing to follow updates on social media platforms like Twitter[1], Facebook[2], and Instagram[3] rather than traditional news sources such as newspapers or television. According to the Pew Research Center, in 2021, nearly two-thirds of American adults (68%) reported obtaining at least some of their news from social media platforms[4], which is a significant increase from about half (49%) in 2012[5]. Getting news from social media can be both advantageous and problematic. The benefits include low cost, easy accessibility, and quick dissemination of information. However, the downside is that it leaves people vulnerable to exposure to false and deceptive information, commonly known as fake news.

According to a study on the spread of online news Vosoughi et al (2018), false news stories are 70% more likely to be retweeted than true stories, they spread significantly farther, faster, deeper, and more broadly than the truth. Fake news stories were widely shared on social media during the 2016 US presidential election, the average US adult was estimated to have read and retained one or several fake news articles during the period, with a higher frequency of exposure to articles supporting the candidacy of Donald Trump as compared to those supporting Hillary Clinton Allcott and Gentzkow (2017). The COVID-19 pandemic has further highlighted the importance of combating the spread of health misinformation. The World Health Organization (WHO) recently conducted a review that included four studies that analyzed the prevalence of health misinformation on social media. The review revealed that the proportion of such misinformation reached up to 51% in posts related to vaccines, 28.8% in posts related to COVID-19, and 60% in posts related to pandemics. The review also states the spread of health misinformation during such outbreaks not only leads to the misinterpretation of available evidence and impacts mental health but also results in the misallocation of health resources and a rise in vaccine hesitancy Borges do Nascimento et al (2022). These findings serve as a wake-up call to the urgency of false information being spread on social media and the need for continued efforts to address it.

[1] https://twitter.com/
[2] https://facebook.com/
[3] https://www.instagram.com/
[4] https://www.pewresearch.org/journalism/2021/09/20/news-consumption-across-social-media-in-2021/
[5] https://www.pewresearch.org/journalism/2016/05/26/news-use-across-social-media-platforms-2016/

The definition of the term "fake news" is a subject of ongoing debate and has yet to be fully agreed upon. A commonly accepted definition is offered by Allcott and Gentzkow in Allcott and Gentzkow (2017), where fake news is defined as "news articles that are intentionally and verifiably false and could mislead readers." For this paper, we will adhere to this definition. Furthermore, in line with our adopted definition of fake news, Twitter defines misleading content ('misinformation') as "claims that have been confirmed to be false by external, subject-matter experts or include information that is shared in a deceptive or confusing manner."[6]

In the literature generally, two types of features (or a combination of these two types) have been utilized for the task of fake news detection: news content and social context-related features Shu et al (2017); Bondielli and Marcelloni (2019).

- News content-based features: Mainly rely on news text and are widely used in previous works Vishwakarma et al (2023); Prakash and Kumar (2023); Rezaei et al (2022); Nasir et al (2021); Kaliyar et al (2020, 2021); Choudhary et al (2021); Ajao et al (2019). Despite the ease of access and demonstrated effectiveness of content-based features in numerous cases, as fake news becomes increasingly sophisticated in its ability to mimic authentic news content, the enlightenment of these features is constantly being threatened and may eventually decline. Furthermore, the limited text space on social media platforms like Twitter restricts the full functionality of these features.
- Social context-based features: Include a broad range of features such as user characteristics Dou et al (2021); Liu and Wu (2018), user posts Nguyen et al (2020); Yu et al (2017), news propagation, and network structure Dou et al (2021); Bian et al (2020); Lu and Li (2020); Monti et al (2019). Evidence indicates that fake and real news have different propagation patterns which could help us differentiate these two types of news Vosoughi et al (2018). Contextual features are more generalizable than content features since they don't rely on writing language and event keywords. Moreover, these types of features are harder to tamper with and thus more reliable. Perhaps the biggest drawback of these features is their restricted access, which is often due to privacy policies that prevent the public availability of such data on many platforms.

Traditional machine learning fake news detection approaches typically require handcrafted features Wu et al (2015); Chang et al (2016); Granik and Mesyura (2017); Rezaei et al (2022) while deep learning techniques automatically learn hidden representations from simple inputs. Recent studies widely employed deep learning methods such as recurrent neural networks Prakash and Kumar (2023); Ruchansky et al (2017); Ma et al (2016), convolutional neural networks Vishwakarma et al (2023); Yu et al (2017); Yang et al (2018), or a combination of different deep learning methods Nasir et al (2021); Liu and

[6] https://help.twitter.com/en/resources/addressing-misleading-info

Wu (2018); Ajao et al (2018), to obtain representations from sequential data like text or sequence of engaged users. Graph neural networks have extended deep learning techniques to model graphs and their complex relationships. Several studies have exploited these techniques for the task of fake news detection Islam Shovon and Shin (2023); Nguyen et al (2020); Bian et al (2020); Monti et al (2019).

In this paper, we propose the use of both content and context-based features and apply three types of neural networks to generate informative representations from these features for the task of fake news detection. With this approach, our framework aims to be more robust and achieve better performance. As illustrated in Fig. 3 our framework consists of three main components. The first component, the BERT-based news text representation, is responsible for acquiring hidden representations from news text content and uses the BERT pre-trained language model that creates dynamic embeddings considering the context of the text. The second component, the RNN-based sequence of engaged users representation, feeds a sequence of features related to users engaged in news dissemination to a variant of the recurrent neural networks called LSTM, generating representations that capture the local variations of user characteristics effectively. In the third component, the GNN-based propagation graph representation, after estimating and constructing the propagation graph, it is given to a graph neural network (we try three models, namely, GCN, GAT, and GraphSAGE, and pick the one with the best results) so that determining propagation patterns are captured and propagation structure is taken into account. Finally, the generated representations are concatenated with some selected explicit tweet features and fed into a multilayer neural network for classification. Also, in our proposed framework we make use of the clues that text emoji meanings might have and don't simply remove them as a pre-processing step. We aim to enhance the accuracy and robustness of fake news detection by considering the emotional context conveyed by emojis, providing a more thorough analysis of the text.

This approach takes into account multiple aspects of news dissemination and recognizes the importance of considering a range of criteria in detecting fake news. By combining these three feature sets, the framework aims to provide a comprehensive and robust solution for detecting fake news, ensuring that no single feature set is relied upon too heavily. This multi-faceted approach is expected to consider a more nuanced and accurate understanding of the news and its origin, improving the overall accuracy and reliability of the fake news detection process.

In the evaluation phase, our proposed framework undergoes testing on two real-world datasets, FANG Nguyen et al (2020) and TWITTER15 Ma et al (2017). The comparative analysis involves models, such as GCN Kipf and Welling (2016), GAT Veličković et al (2018), GraphSAGE Hamilton et al (2017), BERT Devlin et al (2018), LSTM Hochreiter and Schmidhuber (1997), Hybrid CNN-RNN Nasir et al (2021), and GDP Monti et al (2019). The results demonstrate that our proposed framework outperforms both the usage of its

composing components individually and state-of-the-art fake news detection methods, affirming its superior performance in detecting fake news.

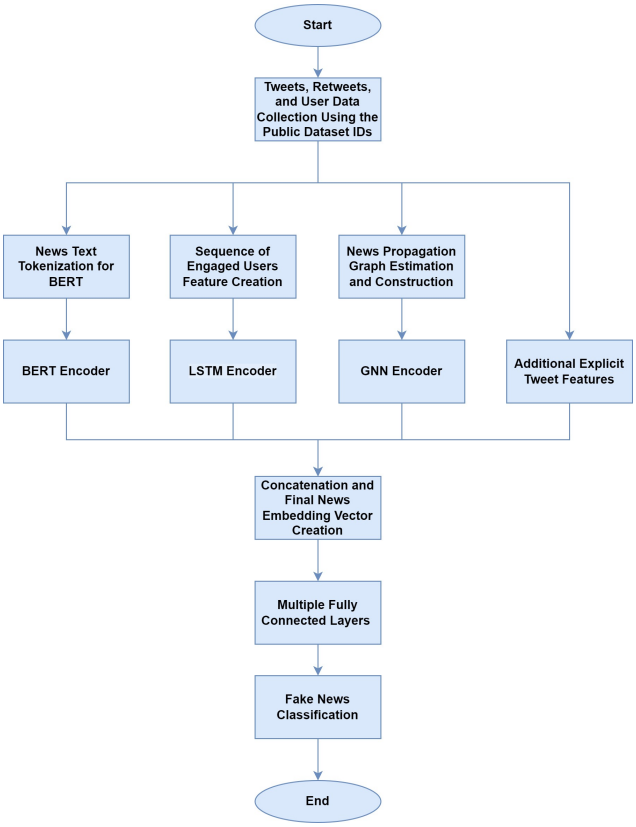


Fig. 1 Flowchart of the Proposed Fake News Detection Framework.

We illustrate the sequential steps of our fake news detection framework in Fig. 1. This flowchart visually represents the progression through the major components of our framework, including BERT-Based News Text Representation, RNN-Based Sequence of Engaged Users Representation, and GNN-Based Propagation Graph Representation, followed by the concatenation of features and prediction using fully connected layers. More details will be provided in Section 3. Our main contributions can be summarized as follows:

- We propose a novel hybrid fake news detection framework that considers multiple aspects of news dissemination by effectively utilizing three key sets of features associated with news articles on Twitter: the news text, the sequence of engaged users, and the news propagation structure.
- The proposed approach goes beyond traditional methods by embedding the meaning of emojis within the input, providing a deeper understanding of the sentiment and intent behind the text.

- The proposed framework is evaluated on two real-world datasets, demonstrating its superior performance compared to using its individual components and state-of-the-art fake news detection methods.

The remainder of the paper is organized as follows: In Section 2, we provide an overview of the related work in the field of fake news detection. Section 3 presents our proposed hybrid multi-feature framework, BRaG, which includes detailed descriptions of the framework's components. Following that, Section 4 describes the experiments conducted to evaluate the effectiveness of our framework including datasets, evaluation setting, and the results. In Section 5, we discuss the results and limitations of our approach. Finally, in Section 6, we conclude the paper by summarizing our contributions providing insights gained from this research, and discussing potential future directions for further exploration.

2 Related Work

Existing fake news detection approaches mainly fall into two categories based on the information they use: news content-based and social context-based Shu et al (2017); Bondielli and Marcelloni (2019). A simple classification of the

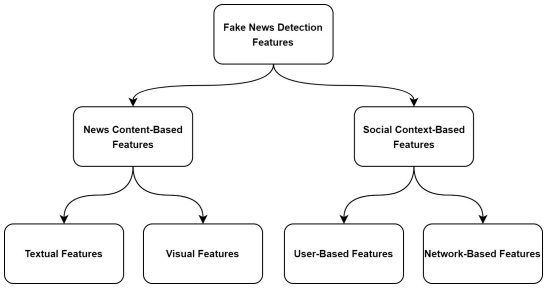


Fig. 2 Categorization of features used for fake news detection.

different kinds of features extracted and used for the task of fake news detection is illustrated in Fig. 2. Content-based features are the most commonly used type of features, they typically rely on the textual and visual content of the news Rezaei et al (2022); Nasir et al (2021); Kaliyar et al (2020, 2021); Choudhary et al (2021); Ajao et al (2019). Although these features are most available and have proven beneficial in many cases, they have several drawbacks. As fake news is constantly improving and becoming more and more similar to real news, these features will eventually lose their descriptiveness. In addition, in some social media platforms such as Twitter, where the statements are brief, these features don't work as effectively. Contextual features include a wide range of features considering various aspects like characteristics of users engaged in the news dissemination Dou et al (2021); Liu and Wu (2018), user posts and comments regarding the news Nguyen et al (2020); Yu

et al (2017), and news propagation structure Monti et al (2019); Wu et al (2015). Studies have shown that fake news and real news follow different propagation patterns, which can be used to distinguish between the two types of news Vosoughi et al (2018). Contextual features are more generalizable than content features because they are not dependent on writing style, language, event keywords, etc. Additionally, manipulating news propagation patterns is a difficult task that cannot be achieved by individual profiteers making these types of features potentially more reliable. The biggest challenge regarding these features is that due to privacy policies, in many social platforms user data are not available publicly.

Traditional machine learning fake news detection methods typically rely on handcrafted features that are designed by domain experts to capture relevant characteristics of the data. These features may include linguistic features such as sentiment, lexical diversity, and readability, as well as social features such as user engagement and network characteristics. However, extracting these features can be time-consuming and computationally expensive, especially when dealing with large datasets. In addition, the selection of handcrafted features may be subjective and may lead to biased features that are not representative of the underlying data distribution. In Granik and Mesyura (2017) authors used a naïve Bayes classifier with bag of words features. A graph-kernel-based hybrid SVM classifier that uses a set of handcrafted propagation and semantic features is proposed in Wu et al (2015). Several classifiers namely logistic regression, naïve Bayes, k-nearest neighbors, random forest, decision tree, support vector machine, and XGBoost have been used on content and social features of COVID-19-related news in Zhou et al (2020a). Rezaei et al. Rezaei et al (2022) proposed an ensemble-based approach that extracts content features such as sentiment and semantic features from the news and employs five primary classifiers - Random Forest, Support Vector Machine, Decision Tree, LightGBM, and XGBoost. A meta-learning algorithm, AdaBoost, is used to develop a stacking generalization model that combines the results from all primary classifiers.

Recent advancements in fake news detection have witnessed a shift towards deep learning techniques. These methods have gained popularity due to their ability to automatically learn hidden representations from input data without the need for extensive domain expertise. This enables them to learn to detect patterns and features that may be challenging or impossible to identify through handcrafted features.

Recurrent neural networks (RNNs) are commonly used for processing sequential or temporal data in natural language processing (NLP) tasks such as text classification, machine translation, and text generation. RNNs are particularly effective in modeling sequence data because they have a memory of past inputs and can use this information to inform their predictions. They work by processing input sequences one element at a time and maintaining an internal state that captures information from the previous elements in the sequence. This makes them well-suited for tasks that require a contextual understanding

of the input. By modeling the sequence of words and phrases in a text, RNNs can potentially identify patterns that are indicative of fake news. In addition to being used for processing sequential or temporal data like text, recurrent neural networks can also be used to obtain representations from sequences of user features, such as user engagement with news articles. In Ma et al (2016) Tanh-RNN, LSTM, and GRU which are different types of recurrent neural network (RNN) architectures have been used for rumor detection in microblogs. Ruchansky et al. Ruchansky et al (2017) proposed a model composed of three modules: capture, score, and integrate, in the capture module LSTM is used to capture the temporal pattern of user responses on a given news. In Yu et al (2017) authors use convolutional neural networks on all correlative microblog posts of an event to extract high-level features from the input sequence and identify misinformation.

In addition to recurrent neural networks, convolutional neural networks (CNNs) can also be used for feature extraction in fake news detection. CNNs are commonly used in computer vision tasks, but they have also been adapted for natural language processing tasks. Multi-modal feature extraction is done using Text-CNN to take into account the similarity across modalities of the news Zhou et al (2020b). Liu et al. Liu and Wu (2018) treat news propagation paths consisting of users engaged in news dissemination as a multivariate time series and embeddings are obtained using both recurrent and convolutional networks to capture both global and local variations of user characteristics along the propagation path.

Graph neural networks (GNNs) are a type of neural network that can operate on graph-structured data. They have gained increasing attention in recent years due to their ability to handle complex, non-linear relationships between graph nodes and to perform tasks such as node classification, link prediction, and graph classification. GNNs operate by iteratively aggregating information from a node's neighbors to update the node's hidden representation. This process is typically repeated multiple times to allow for the representation to incorporate information from nodes at varying distances in the graph. For the first time, Monti et al. Monti et al (2019) encoded news diffusion paths utilizing graph convolutional networks. In each news, graph nodes represent tweets/retweets and their authors and edges represent diffusion paths plus social relations. Authors used Bi-Directional Graph Convolutional Networks to consider both top-down and bottom-up propagation of rumors in Bian et al (2020). The news content, user historical posts, and news propagation graph are jointly taken into account using text representation methods and graph neural networks in Dou et al (2021).

Word embeddings are fixed-length, dense, and distributed representations for words Almeida and Xexéo (2019). Pre-trained word embeddings are an essential component of modern natural language processing (NLP) systems. These embeddings are created by training deep neural networks on large text corpora, enabling them to encode contextual and semantic information about words. GloVe (Global Vectors for Word Representation) Pennington

Table 1 A summary of approaches, features, and data in previous works.

Reference	Approach				Features			Data
	Traditional ML	Deep Learning			Content-based	Context-based		
		RNN	CNN	GNN	Word Embedding Technique	User	Network	
Wu et al (2015)	✓				Explicit features	✓	✓	Weibo
Granik and Mesyura (2017)	✓				BoW			Facebook
Rezaei et al (2022)	✓				Explicit features			Long articles
Chang et al (2016)	✓				Explicit features			Twitter
Ruchansky et al (2017)		✓			word2vec	✓		Twitter and Weibo
Yu et al (2017)			✓		word2vec	✓		Twitter and Weibo
Yang et al (2018)			✓		word2vec	✓		Long articles
Zhou et al (2020a)			✓		word2vec			Long articles
Liu and Wu (2018)		✓	✓		-			Twitter and Weibo
Kaliyar et al (2020)		✓			GloVe	✓		Long articles
Nasir et al (2021)		✓	✓		GloVe			Long articles
Kaliyar et al (2021)			✓		BERT			Long articles
Choudhary et al (2021)			✓		BERT			Long articles
Monti et al (2019)				✓	Not clear	✓	✓	Twitter
Liu and Wu (2018)				✓	GloVe	✓	✓	Twitter
Dou et al (2021)				✓	BERT	✓	✓	Long articles with related tweets

et al (2014) is one such pre-trained word embedding model that has been widely used for various NLP tasks. A hybrid approach for fake news detection framework combining convolutional and recurrent neural networks that extract high-level features from GloVe word embeddings of news text is proposed in Nasir et al (2021). In recent years, more advanced models like BERT (Bidirectional Encoder Representations from Transformers) Devlin et al (2018) have emerged, and have become the de facto standard for pre-trained language models. BERT is a powerful transformer-based model that has been pre-trained on massive amounts of text data, enabling it to capture deep contextual relationships between words. Kaliyar et al. proposed a fake news detection model that uses BERT pre-trained embeddings with different CNN layers for feature extraction Kaliyar et al (2021).

A summary of the literature regarding the approaches, features, and used data is presented in Table 1. In many existing approaches to fake news detection, a primary focus is placed on analyzing the text of news articles. However, this approach may be insufficient, especially when the text is short. To address this limitation, some approaches incorporate content along with contextual features. Nevertheless, with recent advancements in deep learning and the wide variety of contextual data associated with news, there is ample opportunity to exploit. Our proposed method takes a more holistic view of the fake news generation and propagation process by considering three key sets of features associated with news articles. These sets include the news text, the sequence of engaged users, and the news propagation structure. By analyzing these features, our framework aims to capture as many relations, patterns, and insights as possible. We use powerful deep learning methods, such as graph neural networks and BERT, to ensure that we extract the maximum amount of valuable information from the features. In addition to the text, we also take into account the meanings of emojis to better understand the intent and emotions behind an article. Our aim is that by doing so, we can find patterns that differentiate fake news from real news. We also incorporate potentially useful explicit contextual features, such as the number of likes and retweets, to add additional insights. By combining all of these features, we aim to have a more robust and accurate model.

3 Proposed Hybrid Framework: BRaG

Our novel fake news detection framework, named **BRaG**, is designed with three key components: **B**ERT-Based, **R**NN-Based, and **G**NN-Based. Each of these components represents a distinct aspect of news propagation and user engagement:

- **BERT-Based News Text Representation:** This component employs a BERT pre-trained model to encode the tokenized news text, considering emoji meanings, to extract the semantic meaning of the news article.
- **RNN-Based Sequence of Engaged Users Representation:** This component uses a variant of Recurrent Neural Networks (RNNs), namely LSTM,

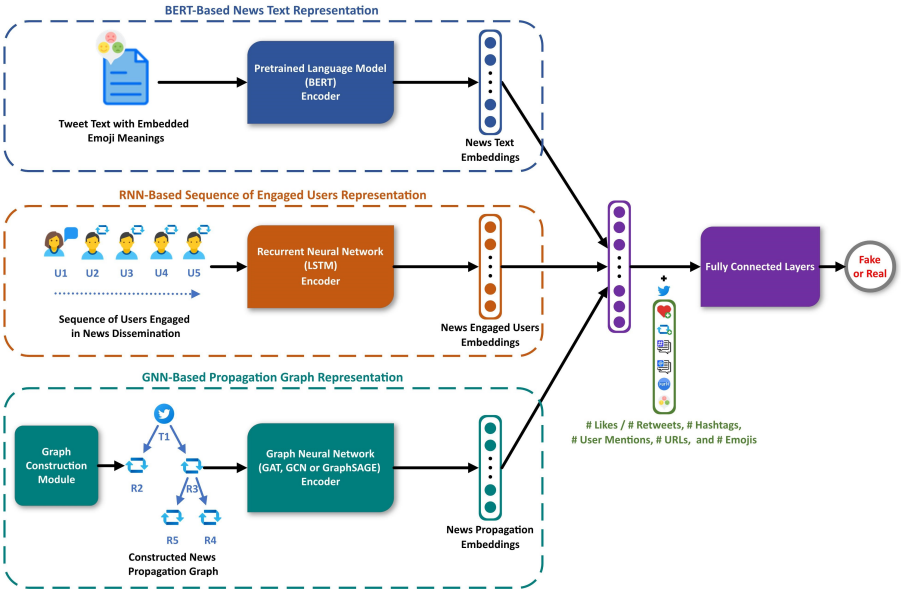


Fig. 3 The architecture of the BRaG framework, includes three components: BERT-based News Text Representation, RNN-Based Sequence of Engaged Users Representation, and GNN-Based Propagation Graph Representation. The hidden representations obtained from these components are combined with additional tweet features to construct the final news embedding vector, which is fed to multiple fully connected layers to make predictions.

to capture the characteristics of the sequence of users engaged in news dissemination.

- **GNN-Based Propagation Graph Representation:** This component utilizes a chosen Graph Neural Network (GNN) to represent the news propagation graph that we estimate and construct and capture the hidden and valuable patterns behind the news propagation structure. We try GCN, GAT, and GraphSAGE models.

The hidden representations obtained from these components are then concatenated with a few additional potentially useful explicit tweet features to construct the final news embedding vector. The news embedding vector is fed to multiple fully connected layers, and prediction is made. The architecture of the proposed model is illustrated in Fig. 3. Next, we will introduce the aforementioned components in detail.

3.1 BERT-Based News Text Representation

BERT Devlin et al (2018) is a highly advanced, pre-trained language model that relies on transformer encoder architecture. Unlike earlier models, BERT produces bidirectional representations that incorporate both the left and right contexts of words within a sentence. BERT has been trained in an unsupervised manner on masked language modeling and next-sentence prediction

tasks using large-scale data corpora, including BooksCorpus Zhu et al (2015) and English Wikipedia, with approximately 800 million and 2,500 million words, respectively, resulting in highly adaptable embeddings that capture word meanings with respect to their underlying concepts. For our approach, we utilize the BERT (base form) architecture, which consists of 12 layers and has fewer parameters than BERT Large, making it faster and more efficient for our purposes. In this section, we will provide a detailed explanation of our methodology for tokenizing the news text and leveraging BERT to encode the text, resulting in the extraction of informative news text embeddings.

3.1.1 Text Tokenization for BERT

The tokenization process in BERT is responsible for transforming raw text into input suitable for the model. BERT utilizes the WordPiece Wu et al (2016) method with a 30000 token vocabulary to accomplish tokenization. This approach involves dividing the text into words and subwords using the predefined vocabulary. Additionally, the tokenizer inserts special tokens like [CLS] to denote the sequence start, [SEP] to separate segments, [PAD] for padding, and [UNK] for unknown words. BERT's input token sequences are required to have the same length. To meet this requirement, during the tokenization process, if the resulting tokenized sequence exceeds the specified maximum length, it is truncated, while if the sequence length is shorter, padding tokens are added until the desired length is achieved.

To illustrate, given a news text, we construct a sequence of tokens denoted as $W_i = w_1, ..., w_X$. As mentioned earlier, these sequences must have a uniform length X. If the text contains more than X tokens, only the first X tokens are retained, and if there are fewer tokens, padding tokens are added until the final count reaches X. For each token, BERT's final input embeddings are generated by summing the corresponding token, segmentation, and position embeddings. This process ensures that the final embeddings capture both the semantic information of the text and its contextual information.

3.1.2 BERT Encoder

We use BERT with T transformer layers. Hidden representations of tokens at transformer layer t ($0 \leq t \leq T$) are denoted as $R^t = \{r_1^t, ..., r_X^t\}$ and computed by (1):

$$R^t = Transformer_t(R^{t-1}) \tag{1}$$

Finally, to obtain the news text embedding BE_i, we extract the hidden token representations from the second to the last layer, which is denoted as R^{T-1}. Then, we apply mean pooling to these hidden representations. The mean pooling operation calculates the average of all the token representations r_j^{T-1} in the sequence as shown in (2) where X represents the number of tokens, and r_j^{T-1} represents the hidden representation of token j from the second to the last layer. By summing up all the token representations and dividing the sum by X, we obtain the mean value. This mean value represents the final news

text embedding BE_i, which captures the contextual information of the news text based on BERT's pre-trained representations.

$$BE_i = \frac{1}{X}\sum_{j=1}^{X} r_j^{T-1} \tag{2}$$

3.2 RNN-Based Sequence of Engaged Users Representation

The RNN-based sequence of engaged users representation component utilizes Long Short-Term Memory (LSTM) Hochreiter and Schmidhuber (1997), a variant of Recurrent Neural Networks (RNNs), to capture the characteristics of the users engaged in news dissemination. In the upcoming subsections, we will describe the process of creating a fixed-length sequence of user features and elaborate on how this sequence is used as input to the LSTM encoder, and how the final representation of the sequence of users engaged in news dissemination is obtained.

3.2.1 Sequence of Engaged Users Creation

For each news article a_i, first, we need to identify users that tweet/retweeted the news and construct the corresponding propagation path $P(a_i)$. As the number of engaged users is not the same for every news article, in a manner similar to Liu and Wu (2018) we first denote users who engaged in the propagation of a_i as a variable-length multivariate time series $P(a_i) = \langle ..., (x_j, t), ... \rangle$ in which x_j is the feature vector of each user engaged in the dissemination of a_i at time t. Since the input sequences of our recurrent neural network need to have the same length, we will go through two steps, if there are more than n tuples in $P(a_i)$, we will only keep the first n tuples and truncate the rest, and if there are less than n tuples in $P(a_i)$, we will randomly oversample tuples in $P(a_i)$ to the point where the length reaches n. The obtained fixed-length multivariate sequence is denoted as $S(a_i) = \langle x_1, ..., x_n \rangle$.

3.2.2 LSTM Encoder

To encode the propagation path that we have constructed, we utilize Long Short-Term Memory (LSTM) Hochreiter and Schmidhuber (1997), a variant of recurrent neural networks optimized for long sequences. The relevant formula is shown in (3), Where W_i, W_f, $W_{\tilde{c}}$, $W_o \in \mathbb{R}^{m \times d}$, U_i, U_f, $U_{\tilde{c}}$, $U_o \in \mathbb{R}^{m \times m}$ are weight matrices, m, and d are the hidden output and user vector dimensions. h_t is the hidden state and x_t is the input at time t. σ and tanh represent the sigmoid and the hyperbolic tangent functions. $\odot$ is the Hadamard product,

and i_t, f_t, $\widetilde{c}_t$, and o_t are the input, forget, cell, and output gates, respectively.

$$\begin{aligned} i_t &= \sigma(U_i x_t + W_i h_{t-1} + b_i) \\ f_t &= \sigma(U_f x_t + W_f h_{t-1} + b_f) \\ \widetilde{c}_t &= tanh(U_{\widetilde{c}} x_t + W_{\widetilde{c}} h_{t-1} + b_{\widetilde{c}}) \\ o_t &= \sigma(U_o x_t + W_o h_{t-1} + b_o) \\ c_t &= f_t \odot c_{t-1} + \ i_t \odot \widetilde{c}_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \tag{3}$$

After utilizing the Long Short-Term Memory (LSTM) network to process the given fixed-length multivariate sequence $S(a_i)$, the LSTM network generates a sequence of hidden states, where each hidden state h_t represents the internal state of the network at time step t. Considering that we are interested in obtaining a single representative vector for the entire sequence, we employ mean pooling as a pooling technique to reduce the number of hidden state vectors. The mean pooling operation is performed by calculating the average of all the hidden state vectors h_t from the LSTM network. This process can be represented as (4), in which n represents the number of hidden states generated by the LSTM network (equal to the length of $S(a_i)$). By summing up all the hidden state vectors and dividing the sum by n, we obtain the mean value capturing the overall information from the LSTM hidden states, which yields the final vector RE_i representing the constructed sequence of engaged users.

$$RE_i = \frac{1}{n} \sum_{t=1}^{n} h_t \tag{4}$$

3.3 GNN-Based Propagation Graph Representation

The GNN-Based Propagation Graph Representation component leverages a selected Graph Neural Network (GNN) to capture the hidden and valuable patterns within the news propagation graph. In this section, first, we provide a detailed description of how we estimate and construct the news propagation graph. Next, we describe the process of encoding the constructed graph to obtain the final representation of the news propagation graph, encapsulating the essential characteristics and hidden patterns within the propagation structure.

3.3.1 Propagation Graph Construction

Given a source news tweet on Twitter, we use available retweet information to estimate and construct a propagation graph. We do this in a manner similar to previous studies such as Dou et al (2021); Monti et al (2019). This section explains what happens in the graph construction module shown in Fig. 3. Given a source tweet r_1 and its retweets sorted by time $\{r_2, ..., r_n\}$ and their corresponding authors (users) $\{u_1, ..., u_n\}$ we will construct a graph with nodes

representing tweets/retweets and their authors and edges representing diffusion paths plus social relations with two rules:

1. If u_i follows any of the users in $\{u_1, ..., u_{(i-1)}\}$ we assume news has spread from a tweet/retweet in $\{r_1, ..., r_{(i-1)}\}$ whose author is followed by u_i and has the latest timestamp.
2. If u_i follows no one from $\{u_1, ..., u_{(i-1)}\}$ we assume the news has spread from a tweet/retweet in $\{r_1, ..., r_{(i-1)}\}$ that its author has the largest number of followers.

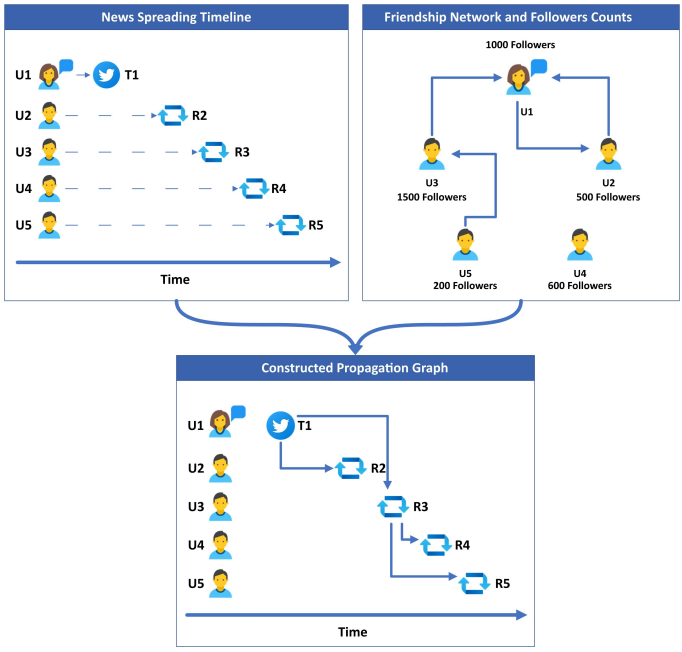


Fig. 4 An example of estimation and constructing propagation graph from the news spreading timeline, friendship network, and followers counts.

An example of how a propagation graph is estimated and constructed using the propagation graph construction module, having friendship networks, number of followers, and spreading timeline is depicted in Fig. 4. In the example, based on the news spreading timeline the list of retweets for news tweet $T1$ posted by User $U1$ sorted by time would be $\{R2, R3, R4, R5\}$ and the corresponding retweeters are $\{U2, U3, U4, U5\}$. First, we estimate news spreads from $T1$ to $R2$. Then for $R3$, we estimate that news propagates from $U1$ and not $U2$ because even though $R2$ has the latest timestamp, $U3$ does not follow $U2$. In the case of $R4$, $U4$ does not follow any of the users in $\{U1, U2, U3\}$ so we turn to the second rule and estimate that the news has propagated from the user with the most number of followers which would be $U3$. Finally, for $R5$, we estimate the source is $R3$ because $U5$ follows $U3$.

3.3.2 GNN Encoder

In the previous section, we constructed news propagation graph $G = (A, F)$ with n nodes representing tweets/retweets and their authors and m edges representing diffusion paths plus social relations, $A \in \{0,1\}^{n \times n}$ is the corresponding adjacency matrix and $F \in R^{n \times d}$ is the node feature matrix. With G as an input, the output of the graph neural network is hidden representations of the graph's nodes. After k steps the output matrix can be represented as $H^{(k)} = f(A, H^{(k-1)};\ \theta^{(k)}) \in R^{n \times s_k}$ in which f is the propagation function parametrized by θ, S_k is kth step hidden size, and $H^{(0)}$ is initialized by the feature matrix $H^{(0)} = F$. The propagation function determines how information is passed between nodes so that the final representations contain a good sense of the graph's topology and features. This task is usually achieved by aggregating information from the node's neighbors with convolution or recurrent operators. In some approaches, especially in large graphs, a sampling module is also combined with the propagation function.

As described, we obtain hidden representations for each node in the graph through the graph neural network. These embeddings are denoted as H_j^K, where j represents the index of the node, and K denotes the number of propagation steps or layers performed by the GNN. To obtain the final representation vector GE_i for the news propagation graph, we utilize mean pooling. The mean pooling operation calculates the average of all the node embeddings H_j^K in the graph. This is done by (5), where n_i represents the total number of nodes in the graph G_i. By summing up all the node embeddings H_j^K and dividing the sum by n_i, we obtain the mean value. This mean value represents the final representation vector GE_i for the graph G_i. The mean pooling operation enables us to aggregate information from all the nodes in the graph and derive a condensed representation that captures the collective knowledge or characteristics of the graph's structure and node features.

$$GE_i = \frac{1}{n_i} \sum_{j=1}^{n_i} H_j^K \tag{5}$$

3.4 Embeddings Concatenation and News Classification

After obtaining BE_i, RE_i, and GE_i embeddings from BERT-based news text representation, RNN-based sequence of engaged users representation, and GNN-based propagation graph representation components, we concatenate these embeddings with some potentially useful additional features EF_i and form the final news embedding which is a single representation vector. The constructed vector is then fed to multiple fully connected neural network layers and the probability of news belonging to fake or real class is determined

according to (6):

$$\begin{aligned} GRaBE_i &= Concatenate(GE_i, RE_i, BE_i, EF_i) \\ I_j &= ReLU(W_j I_{j-1} + b_j), \forall j \in [q] \\ z &= Softmax(I_q) \end{aligned} \tag{6}$$

Where q is the number of hidden layers, $I_j \in \mathbb{R}^{v_j}$ is the output of jth hidden layer and $I_0 = GRaBE$, v_j is jth hidden layer size. $W_j \in \mathbb{R}^{v_j \times v_{j-1}}$ and $b_j \in \mathbb{R}^{v_j}$ are hidden layers' weights and biases respectively. Finally, $z \in \mathbb{R}^r$ is the output showing the probability of the news being fake or real.

4 Experiments

In this section, we present the experimental setup and results of our study. We begin by describing the datasets used for evaluation, followed by the feature extraction details. Next, we discuss the competing methods that we compare our framework against. Then, we provide details about the experimental settings employed in our evaluation. Subsequently, we present the main results of our experiments, highlighting the performance of our framework. Additionally, we examine the impact of setting a minimum retweets threshold and conduct an ablation study on our framework components.

4.1 Dataset and Evaluation Setting

In this section, we provide an overview of the datasets and evaluation settings used in our experiments. We describe the datasets we employed for training and testing our framework, as well as the feature extraction details. Additionally, we discuss the competing methods we compare our framework against to assess its performance. Lastly, we outline the experimental setting employed to conduct our evaluations.

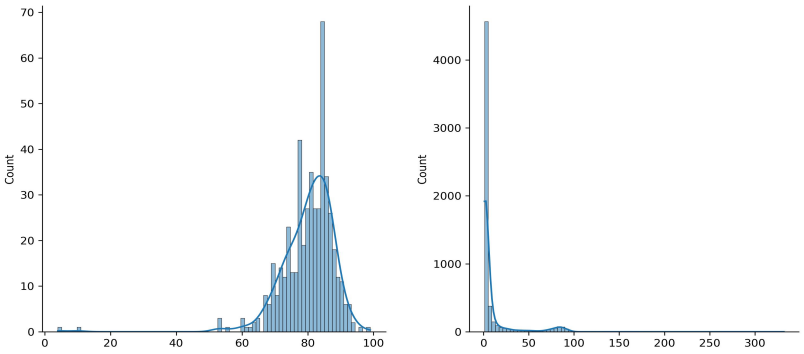
4.1.1 Dataset

We conduct our experiments on two Twitter datasets: FANG Nguyen et al (2020) and TWITTER15 Ma et al (2017). FANG dataset was collected from several related works in rumor and fake news detection. For the TWITTER15 dataset, we used only tweets with true and false labels. We utilized the tweet IDs from the mentioned public datasets to extract all the necessary information, such as tweets and user profiles, through the Twitter API. We acknowledge and respect all privacy rules and ethical considerations when working with Twitter data. Since the public datasets were collected some time ago, it was normal to find that a portion of the tweets and their related information were no longer available. This was expected since some users delete their tweets or accounts over time, or Twitter removes them for violating its rules. We still collect, balance (using under-sampling), and utilize the available data.

Table 2 The statistics of the crawled datasets. The terms "Min 1 Retweet" and "Min 5 Retweets" are used to indicate tweets that have at least 1 or 5 retweets, respectively.

Statistic	**FANG** Nguyen et al (2020)		**TWITTER15** Ma et al (2017)	
	Min 1 Retweet	Min 5 Retweets	Min 1 Retweet	Min 5 Retweets
#SourceTweets (#Fake)	5862 (2931)	1340 (670)	490 (245)	488 (244)
#Retweets	54514	47540	39229	39225
Avg. #Retweets per Tweet	9.29	31.83	80.03	80.24
#Unique Engaged Users	26403	24600	25137	25135

The statistics of our collected datasets can be seen in Table 2. "Min 1 Retweet" and "Min 5 Retweets" indicate tweets with at least 1 or 5 retweets, respectively. It can be observed that the tweets from the TWITTER15 dataset have a higher number of retweets compared to those from FANG.



(a) Distribution of the number of retweets in TWITTER15

(b) Distribution of the number of retweets in FANG

Fig. 5 Comparison of the number of retweets per tweet distribution in our chosen datasets

The frequency of the number of retweets for each tweet in the extracted datasets TWITTER15 and FANG is depicted in Fig. 5. The graph indicates that in the FANG dataset, which contains more data, many news tweets have only a few or zero available retweets. In comparison, news from the TWITTER15 dataset generally receives more retweets. Due to this observation, we will employ a filtering mechanism to ensure a fair evaluation of the framework. Specifically, we will exclude tweets that have less than five available retweets. Further details about the selection of this threshold are provided in section 4.2.2.

4.1.2 Feature Extraction

For each user in the sequence of engaged users, we have a feature vector that includes 100d GloVe Pennington et al (2014) embeddings of the user's self-description text, as well as several selected user-based features: number of user

name words (emojis are counted as words), number of user self-description words (emojis are counted as words), average user favorites per day, number of user friends and followers, user listed counts, user geo-enabled, whether a user is authorized, user status count and relative time of user's engagement with respect to the source tweet (refer to Table 3).

Table 3 List of user features that are added to 100d GloVe embeddings of the user's self-description to construct feature vectors of the users.

No.	Feature	Type
1	username word count	Integer
2	user description word	Integer
3	user favourites per day	Float
4	user followers count	Integer
5	user friends count	Integer
6	user geo enabled	Binary
7	user listed count	Integer
8	user verified	Binary
9	user statuses count	Integer
10	relative engagement time (mins)	Integer

We use BERT embeddings of user self-descriptions as node features for the constructed propagation graph. Finally, to supplement the output vectors of GNN, RNN, and BERT, we select the following five explicit features from the tweet: number of emojis in the tweet, number of hashtags in the tweet, number of user mentions in the tweet, number of URLs in the tweet, and tweet favorite retweet ratio.

4.1.3 Competing Methods

We conduct a thorough evaluation of our proposed model against several baselines and state-of-the-art methods. We also evaluate the individual models that comprise our proposed framework as competing methods. We leverage three popular graph representation learning methods, namely, GCN Kipf and Welling (2016), GAT Veličković et al (2018), and GraphSAGE Hamilton et al (2017), to encode news propagation graphs. GCN (Graph Convolutional Network) performs message passing and feature aggregation over graph neighborhoods. GAT (Graph Attention Network) uses an attention mechanism to weight each node's representation based on its importance. GraphSAGE (GraphSAmple and aggreGatE) is an inductive representation learning method that leverages node feature information and local graph structure to generate embeddings. BERT (Bidirectional Encoder Representations from Transformers) Devlin et al (2018) is a pre-trained language model that has achieved remarkable performance on various natural language processing (NLP) tasks like fake news detection. BERT is based on the Transformer architecture and

is pre-trained using a large corpus of text data to learn general language representations. Long Short-Term Memory (LSTM) Hochreiter and Schmidhuber (1997) is a type of recurrent neural network (RNN) that has been widely used for sequence modeling tasks. We leverage LSTM to classify news based on the sequence of users involved in its dissemination. The LSTM model takes in this sequence of user embeddings and learns to capture the temporal dependencies and patterns in the news dissemination process. Hybrid CNN-RNN Nasir et al (2021) is a hybrid deep learning model that uses a combination of convolutional and recurrent neural networks to extract features from GloVe embeddings of the news text and classify the news. GDP Monti et al (2019) is an automatic fake news detection model based on geometric deep learning that leverages graph convolutional networks for the first time in a fake news detection task. GCN with an attention layer is used to encode news propagation and for fair evaluation, we use BERT embeddings as the model's text encoder. BERT and Hybrid CNN-RNN only rely on news content to classify news. On the other hand, GCN, GAT, GraphSAGE, and LSTM are context-based methods that take into account only news propagation and engaged users. Similar to our approach, GDP uses both the content and context of the news to detect fake news.

4.1.4 Experimental Setting

Our models are implemented using Pytorch[7] and Pytorch Geometric[8] libraries. PyTorch is an open-source machine-learning library. It is widely used in research and industry for developing deep learning models for tasks such as computer vision, natural language processing, and reinforcement learning. PyTorch Geometric is a PyTorch library for deep learning on graphs and other irregular data structures. It provides several graph neural network layers, data preprocessing utilities, and evaluation metrics to enable efficient and scalable learning on graph-structured data. We utilize the computing resources of Google Colaboratory[9], a free Jupyter notebook environment that runs on Google's cloud servers. This enables us to write and execute Python code efficiently and effectively, without the need for local hardware resources.

We randomly divided our dataset into the training, validation, and test sets with a split of 70%, 15%, and 15%, respectively. Each experiment was executed ten times, and the results reported are the average values. After pre-processing the text, considering the statistics of the number of words in pre-processed user self-description texts with their emoji meanings and with trial and error, we concatenate the GloVe embeddings of the first 10 words of each text. Finally, a vector with a length of 1000 is created for each self-description text in the user's profile. For the sequence of involved users' length, we choose 20 taking into account the statistics of the number of users involved in each news and trial and error. For BERT input text we choose the first 20 words

[7] https://pytorch.org/
[8] https://pytorch-geometric.readthedocs.io/
[9] https://colab.research.google.com/

Table 4 BRaG framework chosen hyperparameters

Hyperparameter	**Our Choice**
Epochs	100
Batch Size	64
Learning Rate	0.007
LSTM Hidden Dim	128
GNN Hidden Dim	128
Fully Connected Layers Hidden Dim	256 - 128
Dropout Rate	0.5

of the tweets considering the statistics of the number of words in. The length of each BERT output embedding vector is equal to 768. Cascades containing less than 5 retweets were discarded. We chose this threshold because smaller cascades may not provide enough information, more details are provided in section 4.2.2. The chosen hyperparameters of our framework can be found in Table 4.

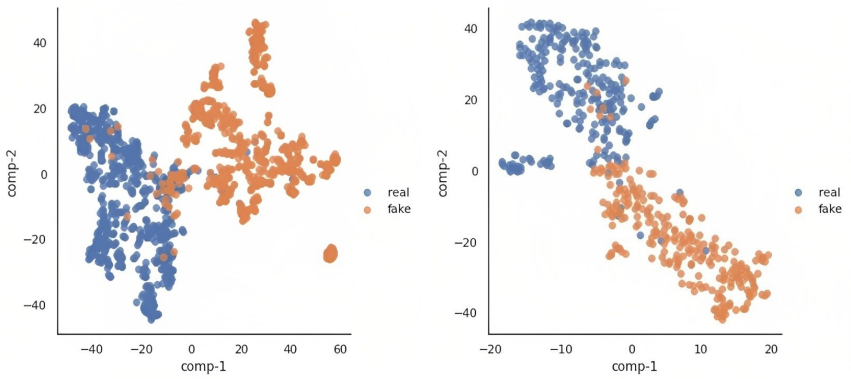
4.2 Results

In this section, we present the results of our framework for detecting fake news. We evaluate its performance based on key metrics such as accuracy (Acc), recall (Rec), precision (Prec), F1 score (F1), and area under the curve (AUC). Additionally, we compare the results obtained from our framework with those of competing methods to assess its effectiveness and superiority. Furthermore, we investigate specific aspects of the results by examining the impact of the minimum retweets threshold and conducting an ablation study on our framework components.

4.2.1 Main Results

This section focuses on the main results obtained from our proposed framework and provides a comparative analysis with the previously introduced baselines. T-distributed stochastic neighbor embedding (t-SNE) is a statistical method that helps visualize high-dimensional data in a lower-dimensional space. The t-SNE embeddings of the features extracted from the second-to-last layer of the proposed framework for the two datasets used are depicted in Fig. 6. In this figure, each point represents a news instance in the dataset, with blue representing real news and orange representing fake news. The clear distinction between the two categories of news is evidence of the effectiveness of the embeddings generated by our framework. It shows that our framework has learned meaningful features for classification, thus validating its performance.

The results in Table 5 suggest that our proposed framework outperforms other approaches in all evaluation criteria and both datasets. However, it should be acknowledged that the proposed framework has a higher complexity compared to other approaches, which leads to slower training speeds and



(a) FANG (b) TWITTER15

Fig. 6 T-SNE embeddings of features extracted from the second-to-last layer of our proposed framework for the two datasets.

Table 5 Fake news detection performance of baselines and our proposed framework.

Feature Type	Model	FANG					TWITTER15				
		Acc.	Rec.	Prec.	F1	AUC	Acc.	Rec.	Prec.	F1	AUC
Content-based	BERT	0.865	0.875	0.865	0.870	0.929	0.828	0.808	0.860	0.833	0.914
	Hybrid CNN-RNN	0.720	0.743	0.728	0.734	0.788	0.661	0.662	0.685	0.671	0.727
Context-based	GAT	0.710	0.759	0.718	0.735	0.775	0.657	0.599	0.725	0.654	0.698
	GCN	0.680	0.710	0.690	0.670	0.742	0.620	0.573	0.669	0.616	0.701
	GraphSAGE	0.727	0.743	0.745	0.742	0.788	0.646	0.585	0.713	0.640	0.723
	LSTM	0.677	0.626	0.727	0.6711	0.718	0.570	0.448	0.629	0.519	0.598
Content + Context based	GDP	0.814	0.814	0.832	0.822	0.890	0.768	0.751	0.802	0.773	0.836
	BRaG	0.886	0.881	0.895	0.888	0.952	0.860	0.870	0.872	0.869	0.921

higher resource requirements. Despite this, given the continuous advancements in hardware resources and the increasing power of graphic processing units, the overhead associated with our approach may become negligible over time.

4.2.2 Min Retweets Threshold

As seen in Fig. 5, in one of our selected datasets, FANG, a large proportion of the source tweets have very few users retweeting them. This poses a challenge for our proposed framework, as one of its components relies on a graph neural network (GNN) generating embeddings for news propagation graphs, which capture useful propagation patterns. In the absence of clear propagation patterns, the performance of the GNN may be compromised. To demonstrate the power of our chosen GNN variation, GraphSAGE, we investigated its performance under different cascade size thresholds, as also done in Monti et al (2019). Specifically, we filtered out samples with less than a certain number of retweets and trained the GNN on the resulting dataset. Fig. 7

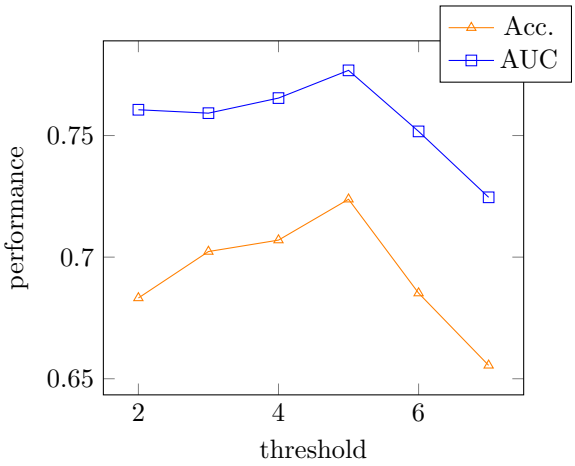


Fig. 7 GraphSAGE fake news detection performance with different minimum number of retweets thresholds, measured by accuracy and area under the receiver operating characteristic curve (ROC-AUC).

displays the obtained results which demonstrate the impact of varying the minimum number of retweets threshold on the model's ability to accurately distinguish between real and fake news. As expected, the performance of the GNN improves as we increase the threshold, as this results in clearer distribution patterns that can aid in classification. However, we observed a decrease in performance after the threshold of five, which could be attributed to the fact that as the threshold increases, more data with retweets less than the threshold are filtered out, resulting in a loss of samples from the dataset. This gradual decrease in the number of samples may ultimately impact the performance of the GNN. Based on these findings, we decided to set the threshold for the minimum number of retweets to five and use the refined dataset for training purposes going forward.

4.2.3 Ablation Study

In this section, we conduct an ablation study to analyze the impact of different components within our fake news detection framework. We explore various graph neural network (GNN) models as propagation graph encoders to determine the most effective variant. Additionally, we evaluate each framework component individually and assess the consequences of removing them.

GNN Encoder Variants

In our proposed framework for fake news detection, we explored different graph neural network (GNN) models as propagation graph encoders to identify the one that yields the best results. Specifically, we experimented with three popular GNN variants: GCN Kipf and Welling (2016), GAT Veličković et al (2018), and GraphSAGE Hamilton et al (2017). The results of our experiments, depicted in Fig. 8, indicate that our proposed framework achieves the

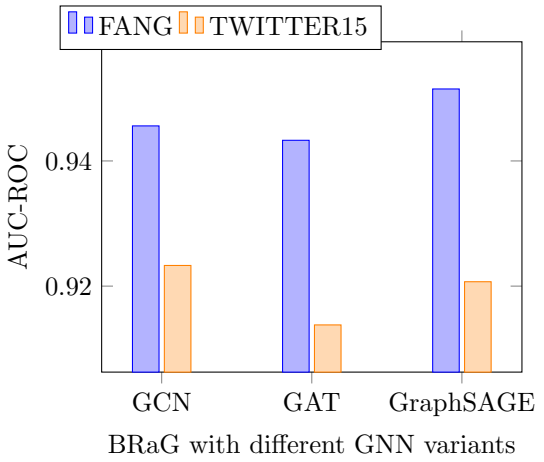


Fig. 8 The impact of various graph neural network (GNN) variations on the performance of the proposed framework, measured by the area under the receiver operating characteristic curve (AUC-ROC).

best performance when using GraphSAGE as the graph encoder. Therefore, we select GraphSAGE as our chosen GNN model for subsequent experiments and evaluations.

Framework Components

To assess the effectiveness and contributions of the components in our proposed framework for fake news detection, we conducted experiments in which we used each component individually with its corresponding features, and also removed each component separately from the framework. The results of these experiments are presented in Table 6. Specifically, "BERT", "RNN", and

Table 6 Proposed framework performance with each component individually or with different components removed.

Model	**FANG**					**TWITTER15**				
	Acc.	Rec.	Prec.	F1	AUC	Acc.	Rec.	Prec.	F1	AUC
BERT	0.865	0.875	0.865	0.870	0.929	0.828	0.808	0.860	0.833	0.914
RNN	0.677	0.626	0.727	0.671	0.718	0.570	0.448	0.629	0.519	0.598
GNN	0.710	0.759	0.718	0.735	0.775	0.657	0.599	0.725	0.654	0.698
BRaG -G	0.878	0.869	0.893	0.880	0.916	0.847	0.861	0.857	0.857	0.912
BRaG -R	0.837	0.832	0.863	0.846	0.915	0.877	0.864	0.895	0.879	0.945
BRaG -B	0.723	0.756	0.736	0.743	0.785	0.641	0.533	0.732	0.613	0.720
BRaG -Emojis	0.870	0.878	0.872	0.875	0.946	0.858	0.876	0.864	0.860	0.918
BRaG	0.886	0.881	0.895	0.888	0.952	0.860	0.870	0.872	0.869	0.921

"GNN" refer to when we used our framework with only one of its components,

while "BRaG -G", "BRaG -R", and "BRaG -B" refer to when we removed the GNN, RNN, and BERT components, respectively, from our framework. Additionally, we investigated the impact of removing emojis from the features, which can be seen in the "BRaG -Emojis" row of the table.

Our study reveals that news text features fed into BERT have the highest performance when used individually, emphasizing the importance of both textual features and advanced language models, such as BERT, in fake news detection. By exploiting BERT's ability to capture complex semantic and contextual information in text, we extract meaningful features that significantly enhance the performance of our proposed framework. Text embeddings were found to have the most substantial contribution to the performance of the framework in both datasets, as their removal resulted in the most significant decrease in performance. Furthermore, while the sequence of users involved in news dissemination fed into the LSTM had the lowest individual performance, these features still provided useful embeddings that contributed to the overall performance of the framework, as their removal reduced the performance metrics. Our analysis also reveals the impact of emojis on the performance of our framework. We observed that removing emojis led to a slight reduction in the performance metrics, indicating that emojis carry useful information in the context of fake news detection. Although the contribution of emojis may seem small, it is promising to consider their meanings in the text to improve the overall performance of the framework.

In conclusion, our experiments demonstrated that each component of our framework, including BERT-based news text representation, RNN-based sequence of engaged users representation, and GNN-based propagation graph representation, contributed to improving the overall performance of the framework. By combining their embeddings, we achieved the best performance in all evaluation criteria. Our findings highlight the importance of incorporating diverse features, including textual, user engagement, and graph structure information, to enhance the detection power of the framework. Moreover, we found that even small details such as emojis can contribute to the overall performance of the framework. Therefore, our work highlights the potential of leveraging various features and advanced models to enhance fake news detection.

5 Discussion

In this paper, we introduced a novel framework for detecting fake news on social media called BRaG. The framework comprises three main components that capture different aspects of news generated and propagated on social media: BERT-based news text representation, RNN-based sequence of engaged users representation, and GNN-based propagation graph representation. The hidden representations obtained from the three components are then concatenated with a few additional potentially useful explicit tweet features to construct the final news embedding vector. This final embedding vector is fed to multiple fully connected layers, and prediction is made. Our framework

takes a distinctive approach by taking into account multiple aspects of news dissemination and recognizing the importance of considering a range of criteria in detecting fake news, enabling us to offer a comprehensive and robust solution for detecting fake news. Unlike many of the previous works that primarily focused on limited features, particularly the news text alone, our framework acknowledges the increasing sophistication of fake news and the necessity of leveraging multiple aspects and features to accurately identify these types of deceptive content. This approach ensures that no single feature set is overly relied upon, enhancing the overall effectiveness of our framework in combating the evolving landscape of fake news.

In the evaluation of our proposed framework on real-life datasets, TWITTER15 and FANG, our work outperforms other baselines and state-of-the-art methods. With an accuracy of approximately 89% for the FANG dataset and 86% for the TWITTER15 dataset, along with corresponding F1 measures of 0.89 and 0.87 respectively, our results provide compelling evidence for the effectiveness of our approach in accurately detecting fake news. Furthermore, we analyzed the individual components as well as the complete framework without certain components. This investigation revealed that all components positively contribute to the overall performance. Notably, the text embeddings played a significant role in improving the framework's effectiveness as their removal led to a notable decline in performance. Specifically, the absence of BERT text embeddings resulted in a 16% decrease in accuracy for the FANG dataset and a 22% decrease for the TWITTER15 dataset. However, there remains untapped potential in exploiting contextual features, suggesting further opportunities for enhancement and exploration in future research.

Moreover, our approach goes beyond conventional pre-processing steps by incorporating the use of text emoji meanings. Rather than discarding emojis, we acknowledge their role in conveying emotional context. By considering the emotions expressed through emojis, we aim to capture subtle nuances and better understand the underlying sentiment and intent behind the news, leading to better performance of our framework. Our experiments revealed a slight performance improvement when considering emoji meanings, with approximately a 2% increase in precision for the FANG dataset and a 1% increase for the TWITTER15 dataset. These results highlight the potential for further enhancements and underscore the valuable insights conveyed through emojis.

However, our study is not without limitations. One major limitation is the reliance on high-quality datasets to effectively train the models. Obtaining such datasets for fake news detection is a challenging task as it requires extensive time and effort in annotating large volumes of data. Additionally, while contextual features have shown promising results, accessing such data from social media platforms can be challenging due to privacy policies and restrictions. Another limitation is the increased resource requirements of more complex models. As models become more sophisticated, they often require substantial computational resources to process large amounts of data. Nevertheless, with

advancements in technology and the availability of more powerful computing resources, this limitation is expected to be mitigated in the future.

6 Conclusion and Future Work

6.1 Conclusion

This paper introduces a novel hybrid and multi-feature framework designed to detect fake news on social media by leveraging both content and context information. The framework combines BERT-based news text representation, RNN-based sequence of engaged users representation, and GNN-based propagation graph representation. Through evaluation on real-life datasets TWITTER15 and FANG, our framework outperforms other baselines and state-of-the-art approaches. With accuracy rates of approximately 89% and 86% respectively, accompanied by corresponding F1 measures of 0.89 and 0.87, our framework demonstrates its effectiveness in accurately detecting and classifying fake news. Furthermore, through individual component analysis and evaluating the framework without each component, we observe positive contributions from all components toward the overall performance, validating the effectiveness of our approach. Additionally, the incorporation of text emoji meanings shows potential, resulting in a slight improvement in the framework's performance. Overall, our framework offers a comprehensive and reliable solution for detecting fake news on social media, considering various aspects of news dissemination and incorporating diverse criteria.

6.2 Future Work

In our future work, we plan to enhance the capabilities of our framework by investigating alternative types of graph neural networks and exploring different pre-trained language models. We also aim to experiment with various normalization techniques and incorporate stance detection and diverse user scoring methods to further improve the performance of our system. Additionally, we see potential in integrating topic and event detection Comito et al (2017) into fake news detection. Furthermore, we are interested in incorporating explainability into our framework to gain insights into the decision-making process of the model. These future endeavors will help us to create a more comprehensive and accurate fake news detection system, enabling us to combat the growing threat of misinformation in today's society.

References

Ajao O, Bhowmik D, Zargari S (2018) Fake news identification on twitter with hybrid cnn and rnn models. In: Proceedings of the 9th international conference on social media and society, pp 226–230

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

Allcott H, Gentzkow M (2017) Social media and fake news in the 2016 election. Journal of Economic Perspectives 31(2):211–236. https://doi.org/10.1257/jep.31.2.211

Almeida F, Xexéo G (2019) Word embeddings: A survey. arXiv preprint arXiv:190109069

Bian T, Xiao X, Xu T, et al (2020) Rumor detection on social media with bi-directional graph convolutional networks. In: Proceedings of the AAAI conference on artificial intelligence, pp 549–556

Bondielli A, Marcelloni F (2019) A survey on fake news and rumour detection techniques. Information Sciences 497:38–55

Chang C, Zhang Y, Szabo C, et al (2016) Extreme user and political rumor detection on twitter. In: International conference on advanced data mining and applications, Springer, pp 751–763

Choudhary M, Chouhan SS, Pilli ES, et al (2021) Berconvonet: A deep learning framework for fake news classification. Applied Soft Computing 110:107,614

Comito C, Falcone D, Talia D (2017) A peak detection method to uncover events from social media. In: 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), IEEE, pp 459–467

Devlin J, Chang MW, Lee K, et al (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805

Dou Y, Shu K, Xia C, et al (2021) User preference-aware fake news detection. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 2051–2055

Granik M, Mesyura V (2017) Fake news detection using naive bayes classifier. In: 2017 IEEE first Ukraine conference on electrical and computer engineering (UKRCON), IEEE, pp 900–903

Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. Advances in neural information processing systems 30

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural computation 9(8):1735–1780

Islam Shovon I, Shin S (2023) The performance of graph neural network in detecting fake news from social media feeds. In: 2023 International Conference on Information Networking (ICOIN), pp 560–564, https://doi.org/10.1109/ICOIN56518.2023.10048961

Kaliyar RK, Goswami A, Narang P, et al (2020) Fndnet–a deep convolutional neural network for fake news detection. Cognitive Systems Research 61:32–44

Kaliyar RK, Goswami A, Narang P (2021) Fakebert: Fake news detection in social media with a bert-based deep learning approach. Multimedia tools and applications 80(8):11,765–11,788

Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:160902907

Liu Y, Wu YF (2018) Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In: Proceedings of the AAAI conference on artificial intelligence

Lu YJ, Li CT (2020) Gcan: Graph-aware co-attention networks for explainable fake news detection on social media. arXiv preprint arXiv:200411648

Ma J, Gao W, Mitra P, et al (2016) Detecting rumors from microblogs with recurrent neural networks. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. AAAI Press, IJCAI'16, p 3818–3824

Ma J, Gao W, Wong KF (2017) Detect rumors in microblog posts using propagation structure via kernel learning. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Vancouver, Canada, pp 708–717, https://doi.org/10.18653/v1/P17-1066, URL https://aclanthology.org/P17-1066

Monti F, Frasca F, Eynard D, et al (2019) Fake news detection on social media using geometric deep learning. arXiv preprint arXiv:190206673

Borges do Nascimento IJ, Pizarro AB, Almeida JM, et al (2022) Infodemics and health misinformation: a systematic review of reviews. Bulletin of the World Health Organization 100(9):544–561

Nasir JA, Khan OS, Varlamis I (2021) Fake news detection: A hybrid cnn-rnn based deep learning approach. International Journal of Information Management Data Insights 1(1):100,007

Nguyen VH, Sugiyama K, Nakov P, et al (2020) Fang: Leveraging social context for fake news detection using graph representation. In: Proceedings of the 29th ACM international conference on information & knowledge management, pp 1165–1174

Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543

Prakash O, Kumar R (2023) Fake news detection in social networks using attention mechanism. In: Proceedings of the International Conference on Cognitive and Intelligent Computing: ICCIC 2021, Volume 2, Springer, pp 453–462

Rezaei S, Kahani M, Behkamal B, et al (2022) Early multi-class ensemble-based fake news detection using content features. Social Network Analysis and Mining 13(1):16

Ruchansky N, Seo S, Liu Y (2017) Csi: A hybrid deep model for fake news detection. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp 797–806

Shu K, Sliva A, Wang S, et al (2017) Fake news detection on social media: A data mining perspective. ACM SIGKDD explorations newsletter 19(1):22–36

Veličković P, Cucurull G, Casanova A, et al (2018) Graph attention networks. In: International Conference on Learning Representations, URL https://openreview.net/forum?id=rJXMpikCZ

Vishwakarma D, Meel P, Yadav A, et al (2023) A framework of fake news detection on web platform using convnet. Social Network Analysis and Mining 13. https://doi.org/10.1007/s13278-023-01026-7

Vosoughi S, Roy D, Aral S (2018) The spread of true and false news online. science 359(6380):1146–1151

Wu K, Yang S, Zhu KQ (2015) False rumors detection on sina weibo by propagation structures. In: 2015 IEEE 31st international conference on data engineering, IEEE, pp 651–662

Wu Y, Schuster M, Chen Z, et al (2016) Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:160908144

Yang Y, Zheng L, Zhang J, et al (2018) Ti-cnn: Convolutional neural networks for fake news detection. arXiv preprint arXiv:180600749

Yu F, Liu Q, Wu S, et al (2017) A convolutional approach for misinformation identification. In: IJCAI, pp 3901–3907

Zhou X, Mulay A, Ferrara E, et al (2020a) Recovery: A multimodal repository for covid-19 news credibility research. In: Proceedings of the 29th ACM international conference on information & knowledge management, pp 3205–3212

Zhou X, Wu J, Zafarani R (2020b) SAFE: Similarity-Aware Multi-modal Fake News Detection. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, pp 354–367

Zhu Y, Kiros R, Zemel R, et al (2015) Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In: Proceedings of the IEEE international conference on computer vision, pp 19–27