



Article

A Survey on the Use of Graph Convolutional Networks for Combating Fake News

Iraklis Varlamis ^{1,*} , Dimitrios Michail ¹ , Foteini Glykou ² and Panagiotis Tsantilas ²

¹ Department of Informatics and Telematics, Harokopio University of Athens, 17671 Athens, Greece; michail@hua.gr

² Palo Services Ltd., 10562 Athens, Greece; fglykou@gmail.com (F.G.); pt@paloservices.com (P.T.)

* Correspondence: varlamis@hua.gr; Tel.: +30-210-954-9405

Abstract: The combat against fake news and disinformation is an ongoing, multi-faceted task for researchers in social media and social networks domains, which comprises not only the detection of false facts in published content but also the detection of accountability mechanisms that keep a record of the trustfulness of sources that generate news and, lately, of the networks that deliberately distribute fake information. In the direction of detecting and handling organized disinformation networks, major social media and social networking sites are currently developing strategies and mechanisms to block such attempts. The role of machine learning techniques, especially neural networks, is crucial in this task. The current work focuses on the popular and promising graph representation techniques and performs a survey of the works that employ Graph Convolutional Networks (GCNs) to the task of detecting fake news, fake accounts and rumors that spread in social networks. It also highlights the available benchmark datasets employed in current research for validating the performance of the proposed methods. This work is a comprehensive survey of the use of GCNs in the combat against fake news and aims to be an ideal starting point for future researchers in the field.



Citation: varlamis, I.; Michail, D.; Glykou, F.; Tsantilas, P. A Survey on the Use of Graph Convolutional Networks for Combating Fake News. *Future Internet* **2022**, *14*, 70. <https://doi.org/10.3390/fi14030070>

Academic Editor: Daniel Gutiérrez Reina

Received: 28 January 2022

Accepted: 22 February 2022

Published: 24 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: fake news; graph convolutional networks (GCNs); misinformation; fake accounts; bots; astroturfing

1. Introduction

The recent debate on COVID-19 vaccination, political debates that took place in national level the last two decades, other events and issues of global interest, such as world tragedies, war-related migration, global warming, etc., raise discussions in social media and online news. Social media also had their own special role in some of these events, starting from the role of Twitter in the US Presidential Elections of 2008 until the more recent Elections of 2016 [1] that popularized the use of the term “fake news” around the world and made it the Word of the Year in the Collins Dictionary in 2017 [2]. The COVID-19 pandemic and the lack of information regarding the reasons, prevention or cure, especially during the first months of the pandemic, fueled the spread of numerous rumors and hoaxes and led to several organized attempts to spread misinformation [3].

The combat against fake news and disinformation is an ongoing, multi-faceted task for researchers in the social media and social network domains, which comprises, as shown in Figure 1 not only the detection of false facts in published content but also the detection of accountability mechanisms that keep record of the trustfulness of sources that generate news and, lately, of the networks that deliberately distribute fake information. In the first direction, widely popular fact-checking sites, such as Snopes (<https://www.snopes.com/>, accessed on 17 February 2022), FactCheck.org (<https://www.factcheck.org/>, accessed on 17 February 2022) and Politifact (<https://www.politifact.com/>, accessed on 17 February 2022), have taken on the mission of checking the rumors, health claims and political claims

that flood up the news sites but primarily social media. These sites monitor a large amount of news sources, including popular social media accounts of politicians, artists, bloggers and other public persons, and have devised their truthfulness rating systems, such as the Truth-O-Meter of PolitiFact, which assigns a score to each public statement and are useful resources for checking the facts of individual posts or the overall credibility of public persons.

In the same discipline are the sites that detect and debunk hoaxes, such as the Hellenic Hoaxes (<https://www.ellinikahoaxes.gr/>, accessed on 17 February 2022) in Greece, Cek-Fakta (<https://cekfakta.com/>, accessed on 17 February 2022) in Indonesia and others, who mostly focus on rumors and stories that lurk on social media and intentionally attempt to misinform readers. Human editors are collecting facts that debunk the myths of social media and reveal the true story behind the fake facts. The International Fact-Checking Network (<https://www.poynter.org/ifcn/>, accessed on 17 February 2022) is a growing community of fact-checkers around the world that collects factual information and facilitates the networking, capacity building and collaboration of fact-checkers in order to combat against fake facts and news. Hoaxy (<https://hoaxy.osome.iu.edu/>, accessed on 17 February 2022) is another popular tool in this domain, which mostly focuses on the visualization of how news (including fake news and hoaxes) flows within social media. Hoaxy tracks the social sharing of links to stories published by two types of websites: (i) low-credibility sources that often publish inaccurate, unverified claims and (ii) independent fact-checking organizations, such as snopes, politifact and factcheck, that routinely fact check unverified claims.

In the second direction, tools such as NewsGuard (<https://www.newsguardtech.com/>, accessed on 17 February 2022) allow news consumers to navigate through reliable and unreliable news sources online, harnessing the power of a large crowd of journalists and reporters, who manually review news sources based on different criteria, including the false content rate, responsibility in correcting errors, use of deceptive headlines (furthermore, known as clickbait), ownership and financing disclosure, etc., which assess basic practices of credibility and transparency. The respective news sources are categorized as: (i) green when they adhere to the basic standards of credibility and transparency, (ii) red when they fail to meet these standards and severely violate journalistic standards, (iii) satire when they are not actual news sites but clearly publish false information for satire purposes, or (iv) platform when the content is user-generated and thus has to be cross-checked per case for its reliability. Another category of tools that capitalizes on the detection of fake sources includes Botometer (<https://botometer.osome.iu.edu/>, accessed on 17 February 2022), a site that checks the activity of a Twitter account and gives it a score that defines its bot-like activity.

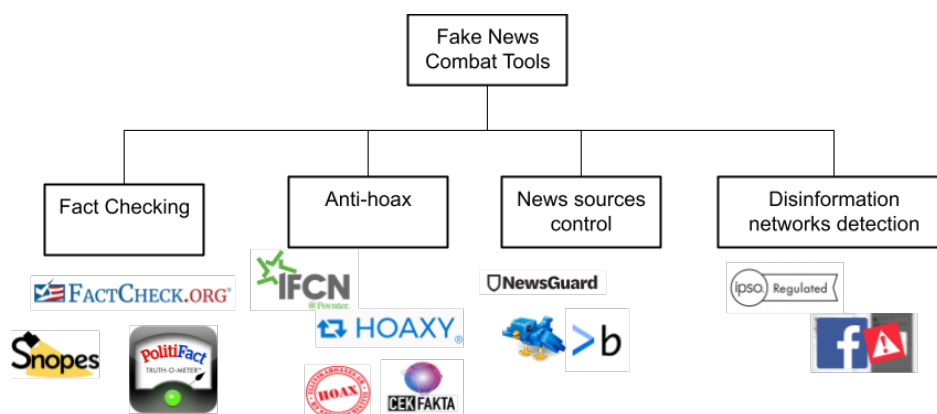


Figure 1. An overview of the various tools that combat fake news and misinformation.

In the direction of detecting and handling organized disinformation networks, also called astroturfing campaigns [4], major social media and social networking sites are

currently developing strategies and mechanisms to block such attempts. For example, Facebook recently announced that it took down disinformation networks tied to major political actors and events around the world (e.g., militant groups, groups that intensify the immigration crisis, anti-vaccine groups etc.) (<https://www.washingtonpost.com/technology/2021/12/01/facebook-disinformation-report/>, accessed on 17 February 2022). However, the task of uncovering organized fake news campaigns is still a hard one that requires the proper understanding of the social landscape [5] and the internal mechanisms of social media and networking platforms and the ways news spread on them [6].

In this paper, we focus on the above tasks, the detection and evaluation of fake news sources, fake accounts and organized disinformation attempts that generate and spread fake news in social media. We mainly examine the network structure and the network-related features of this spreading of news and emphasize the use of graph neural networks in the modeling of similar tasks. Although there already exist a few survey works on GCNs and their use in classification tasks [7,8] and several works that tackle the fake news problem using deep neural networks [9–12], this is the first work, according to our knowledge, that specifically surveys the use of Graph Convolutional Networks for the detection of fake news, fake users and rumors.

In the sections that follow, we perform a survey on the methods that use Graph Convolutional Networks to model the information related to fake news (i.e., content and social graph) and detect fake news items, rumors, bot and spammer accounts. Section 2 introduces the main concepts of Graph Convolutional Networks and briefly explains how graphs are constructed. Section 3 details the three different research directions that focus on detecting fake news items, sources and rumors or misinformation campaigns and discusses the main research works in each direction. It also formulates the problem in each case and provides an overview of the processing pipeline in each task. Section 4 lists the datasets used in these works and provides a summary of their main features and sizes. Section 5 provides more implementation details on some of the works; lists the tools that they employ for handling text, creating the graphs or training the GCNs; and provide links to some useful code repositories for starting with GCNs and fake news detection. Section 6 performs a discussion on the main features of GCNs methods that have been proposed in the literature, comparatively evaluates their performance on the same datasets when possible and explains the advantages of each method. It also briefly compares GCNs against simple Convolutional Neural Networks and Recurrent Neural Networks. Finally, Section 7 summarizes the problems encountered by the different techniques and highlights areas for further research on the field. As a whole, the article provides a comprehensive survey of the very interesting task of detecting fake news and using Graph Convolutional Networks as promising tools for this task.

2. From Graph Embeddings to Graph Convolution Networks

The social media platform that mostly attracts the interest of researchers that study the organized diffusion of fake news is Twitter. This is mainly because the platform provides information about the individual accounts (i.e., sources of news), the content they create (i.e., tweets) and the content they reproduce or share (i.e., by retweeting). The literature behind disinformation campaigns on Twitter is long [4,13–15] and keeps growing when new election campaigns occur around the globe or when crypto-currency and stock investors want to influence the market through social media [16–19].

A common feature in all these campaigns is the creation of fake accounts. These fake accounts, also called social bots or sybil accounts, are massively created by software programs and are used to artificially amplify fake news by reproducing them in social media. Social media and networking sites develop mechanisms to detect and block the bulk creation of fake accounts, while researchers develop machine learning techniques [20] and services to detect bots and check the veracity of news. However, there are still ways for bots to infiltrate social media [9,21].

Graph representations naturally fit numerous application domains [8], particularly social analysis, because of their unique capability to capture the structural relations of social networks and the dynamics of data diffusion in them. Although the underlying connectivity patterns are often complex and diverse, the great success of representation learning in many domains makes graph representation learning a very promising yet challenging solution to many social network problems, including the detection of fake news and their diffusion networks. Graph embedding methods [22,23] can simplify the underlying graph structure and support solving simpler tasks, such as edge or node classification. However, they still fail to capture more complex graph patterns, and this is where deep graph neural networks come in hand. Graph Convolutional Networks (GCNs) have a great expressive power to learn the stationarity and compositionality of graphs to extract hierarchical patterns and high-level features.

GCNs [24] are multilayer neural networks that operate on a graph and learn embeddings for each node based on the properties of the node and its neighbors. Using a single layer of convolution, the GCN can only capture information about the immediate neighbors of the node, but this can be extended by stacking multiple GCN layers. Given a graph $G = (V, E)$, with a set of n vertices V and a set of m edges E that connect them, the adjacency matrix A of G contains 1s in the diagonal, if we assume that each vertex is connected to itself, and 1 in any place i, j when there is an edge e_{ij} connecting node v_i to node v_j . The feature matrix $X \in \mathbb{R}^{n \times k}$ contains in each row x_i the feature vector for vertex v_i , where k is the number of features that describe each vertex of G . When a single layer GCN is employed, the q -dimensional node feature matrix $L^{(1)}$ is defined as $L^{(1)} = \sigma(\bar{A}XW_0)$, where $\bar{A} = D^{-1/2}AD^{-1/2}$ is the normalized symmetric adjacency matrix, D is the degree matrix of G (a diagonal matrix with $D_{ii} = \sum_j A_{ij}$), $W_0 \in \mathbb{R}^{k \times q}$ is a trainable weight matrix and σ is an activation function, such as ReLU. Consequently, X can be considered $L^{(0)}$, the original feature vector representation of graph nodes. Higher order neighbor information (e.g., at h hops) can be incorporated by stacking multiple GCN layers and computing the representation in an iterative way:

$$L^{h+1} = \sigma(\bar{A}L^{(h)}W_h), \quad (1)$$

where W_h is a trainable weight matrix.

Graph convolutions can be applied in any type of data that can be represented as a graph, including text corpora [25], video or images [26], in an attempt to capture the intrinsic semantic information hidden in the documents or the spatio-temporal information hidden in the consecutive images of a video.

In the former case, word and document nodes are connected with each other based on word occurrence in documents (document–word edges) and word co-occurrence in the whole corpus (word–word edges). Edge weights are based either on the term frequency-inverse document frequency (TF-IDF) of the word in the document or in the point-wise mutual information (PMI) of two words in the documents of the collection. The resulting text graph captures both document–word relations and global word–word relations, and the respective GCN model computes the new features of a node (either document or word) as the weighted average of itself and its k -order neighbors [25]. The new node representations can then be fed to a classification or clustering algorithm to solve a respective text task.

In the latter case, the nodes of the graph can be images or video scenes, which are originally represented using handcrafted feature vectors. In the case of activity recognition from videos, skeleton-based data are obtained from each frame in a sequence of frames. The resulting spatio-temporal graph comprises the joints as nodes and the natural connections between human body parts as edges. An additional set of edges comprises the connections between each particular joint in consecutive frames, which constitute the joint trajectory over time [26]. The respective node representations that result after applying multiple convolutions on the original node feature matrix capture the joint features across the frames, as well as the features of the neighboring joints, thus perfectly capturing the spatio-temporal variations that occur in neighboring joints during specific activities (e.g., the movement of

lower-body joints when walking or running and the movement of upper-body joints when we are performing a standing activity).

3. Using GCNs for the Detection of Fake News

As stated in the introduction, the detection of fake news in social media can be targeted into three different disciplines (see Figure 2): (i) the fake news content, (ii) the sources that generate fake news and (iii) the networks that amplify the fake news spreading. The survey of the application of GCN in the task of fake news detection will consequently be performed in these three directions in the subsections that follow.

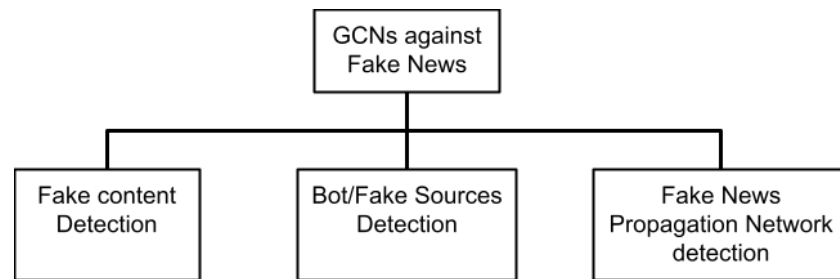


Figure 2. The three main directions in the use of GCNs in fake news detection tasks.

3.1. Detecting Fake Content Using GCNs

The approaches that detect fake news from content usually extract textual features from the text of each post or concatenate to them the visual features of any images that are included in the post, consequently training binary classifiers that decide whether the content is fake or not.

The authors in [27] proposed a Knowledge-driven Multimodal Graph Convolutional Network (KMGCN) to jointly model the textual and visual information of a post into a unified semantic representation. In a pre-processing step, the text of a post is analyzed using entity linking methods that detect the entities mentioned in the text. The entities are then conceptualized using external knowledge graphs. In the same step, they employ a trained YOLOv3 detector to recognize semantic objects in the associated post images. The labels of these objects are added to the post words, and the respective concepts from the knowledge graph constitute the nodes of the graph (represented using Word2Vec word embeddings) in the graph construction step. The edges of the graph are constructed based on the PMI of nodes in the documents of the collection. The network that learns the representation vector for each document comprises two GCN layers and a global mean pooling layer that aggregates the vertices of each graph. The representation vector of each post is then fed to a binary classifier.

In [28], the authors use word embeddings to fetch the low-dimension representation of a single word and sum-pooling to obtain the fixed-length representation vector for each news item (node features). They also use side information that corresponds to the profile of users that make the posts in order to update the adjacency matrix (i.e., to create edges between news items when they share authors with similar profile features). Instead of stacking multiple GCN layers to merge the long-distance information, they calculate the different distance proximity matrices to describe the correlation between nodes. Consequently, they feed different depth proximity matrices to the GCN and follow the update rule of Equation (1).

In a slightly different approach, the graph in [29] is created with words as nodes in an attempt to preserve non-consecutive and long-range dependencies among words and capture structural information at the level of entities. Each post is modeled as three separate graphs, which contain, as nodes, the post words and the respective concepts that result from knowledge conceptualization pre-processing. The graphs comprise a global graph H , a parameterized graph U and an individual graph Q . The edges in graph H are

weighted using the PMI score (word co-occurrence frequency). Graph U is parameterized and optimized during training, and the edges of graph Q are drawn using the normalized embedded Gaussian function to measure the similarity between two words in an embedding space. H, U and Q are added to the final adaptive adjacency matrix, and an adaptive Graph Convolutional Network is trained to exploit the rich semantic relations between words in a data-driven manner (mainly by adjusting the parameters of graph U). The visual features of each post are extracted by feeding images to a separate VGG-19 network and a fully connected layer. Then, a feature-level attention mechanism is employed to learn the correlations between visual and textual content. The concatenated text and visual features constitute the final representation of each post. A fully connected layer activated by a non-linear function is used to classify posts as fake news or not.

Finally, the authors in [30] construct a heterogeneous graph with different types of nodes and edges to integrate information about news' relevance in time, content, topic and source and propose a GCN that utilizes a wider receptive field, a neighbor sampling strategy and a hierarchical attention mechanism. The heterogeneous graph contains four types of nodes (i.e., news, domains, reviews and sources) and its edges link the news nodes with other news nodes with similar content or with their domain, source or review nodes. The neighbor sampling strategy allows controlling the number of nodes that are considered (indirectly, through a common domain or a common source, for example) relevant to a news' node and the respective explosion in the number of neighbors that may occur at each graph convolution layer. This approach attempts to include side information about the news, such as the reviews, which in a way goes beyond the news content itself and examines the actual response of the social network. However, it does not examine the diffusion of a fake news item in the network.

The problem formulation begins with a set of news posts $\mathcal{P} = \{p_1, \dots, p_n\}$, which contain textual or visual content and side information that may correspond to the profile of the user that made the post. The content of each $p \in \mathcal{P}$ is pre-processed to extract key concepts, terms or features, which are mapped to the respective textual R_t or visual R_v embeddings either using pre-trained word or image embedding models, such as GloVe or VGG-19. The next step, as shown in Figure 3, is the creation of an undirected graph $\mathcal{G} = V, E$, with concepts/features as nodes $u \in V$ and edges $e \in E$ that denote relations between concepts (e.g., based on co-occurrence in posts). Using the embeddings R (concatenations of R_t and R_v) of all nodes, the input matrix $X \in \mathbb{R}^{n \times d}$ is created, where n is the number of nodes and d is the dimensionality of the node embeddings. The adjacency matrix $A \in \mathbb{R}^{n \times n}$ is used to represent the set of edges E .

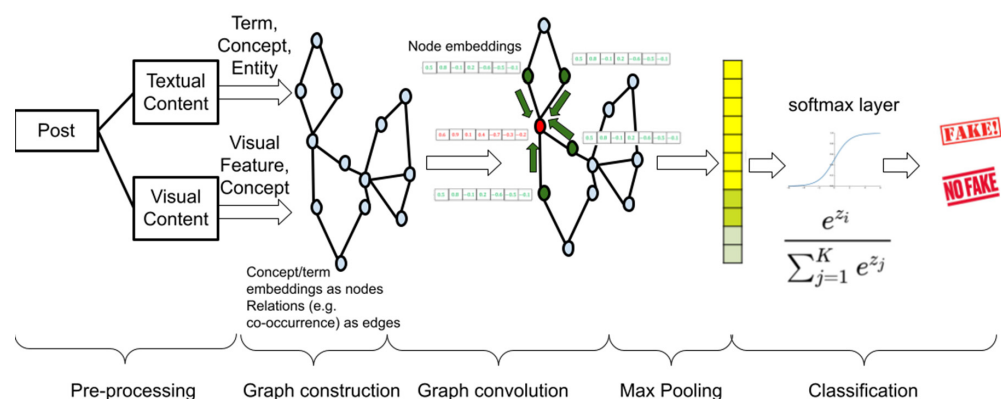


Figure 3. An overview of the fake news detection process using GCN.

The Graph Convolution Network (GCN) in the next step aims to capture the effect of neighboring nodes (i.e., concepts) to the embedding of each node. Each GC layer is formally represented as a function that takes as input the feature matrix from the previous layer $Z^{(j)}$ and outputs a higher level matrix $Z^{(j+1)}$, as shown in Equation (2):

$$Z^{(j+1)} = \sigma(\hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2} Z^{(j)} W) \quad (2)$$

where $\hat{A} = A + I$, I is the identity matrix, $\hat{D}_{ii} = \sum_j \hat{A}_{ij}$, W is the transformation matrix for the j -th layer, and $Z^{(0)} = X$. σ is a non-linear activation function, e.g., a ReLU $\sigma(x) = \max(0, x)$.

The next step is to apply a global mean pooling to aggregate the vertices of each graph and obtain the representation vector of posts. The vector of each post is fed to a binary classifier that employs a cross-entropy classification loss to distinguish between fake and non-fake news.

3.2. Detecting Fake Sources Using GCNs

The detection of bot or fake accounts is an important yet challenging task that can significantly help in combating the propagation of fake news in social networks. The main characteristic of such accounts is that they disguise and pretend to be legitimate user accounts and that they usually operate collectively and in bursts. Graph representations and GCNs have proven helpful in the task of correctly detecting such accounts, taking into account the features of the accounts primarily but also information from their neighborhood.

Feng et al. [31] combine the user account description, the content that it posts, other numerical and categorical features that they can assign to an account and the information of neighboring accounts in order to distinguish bots among users. They use RoBERTa [32] to encode user description and user posts (tweets). They also attach numerical (e.g., number of followers, followees, likes, etc.) and categorical (e.g., geolocation enabled, contributors enabled, etc.) features available from the Twitter API. Consequently, the users are treated as nodes in a graph, which has two types of edges (i.e., following and follower). By applying Relational Graph Convolutional Networks to the graph, they learn user representations, which are fed to a softmax layer for the binary classification task (i.e., bot or not).

Similarly, the authors in [33] propose a bot detection technique that combines social network altmetrics and GCN to distinguish between bots and humans. The unweighted and undirected graph that is the basis of this technique contains Twitter users as nodes and edges that correspond to retweets or mentions. The properties of a user refer to several altmetrics, including degree, triangle count, closeness centrality, etc. A four-layer GCN model and a softmax dense layer are used for classifying the nodes as tweet posted by humans or bots.

In a very recent work [34], the authors combine BERT and GCN and propose a transductive learning method for detecting social bots. The method is based on a heterogeneous graph that comprises word and document nodes, which represent the unique words (vocabulary) and documents in the collection. Edges either denote the word occurrence in a document (word–document edges) or word co-occurrence (word–word edges) and are weighted using TF-IDF and PMI, respectively. The resulting graph is fed to a two-tier GCN, and its output is sent to the softmax classifier. An auxiliary classifier is mainly built by embedding the document using BERT and feeding it to a dense layer with softmax activation. The joint optimization of BERT and GCN parameters is carried out by using the cross-entropy loss at the nodes of the markup document.

Social spammers are the targets in [35], where the authors model the social network as a graph with both uni- and bi-directional edges and users as nodes. They also replace the forward propagation rule of GCN on a single directed graph (as shown in Equation (1)) with a layer-wise propagation rule that aggregates the propagation over the incoming, outgoing and bi-directional edges of a node. They also assume that spammers have more outgoing neighbors than incoming ones and use a pairwise Markov Random Field (MRF) to model the joint probability distribution of all users' identities. The MRF layer is stacked on top of the modified GCN and the whole model can be trained in an end-to-end manner to correctly classify a node as a spammer user or not.

Recently, the authors in [36] proposed a weak supervised learning approach to transform the PHEME dataset into a rumor spreader dataset. They exploited the user-user reply

graph that is inherent in this dataset and use sentiment analysis of the tweets to identify rumors (i.e., tweets with negative sentiment in the replies are identified as rumors). The text and user profile features along with some ego-network features of each user constitute the node (user) information of the social graph, which is fed to a GCN in order to classify users as rumor spreaders or not.

Dong et al. [37] focused on detecting rumors and proposed a GCN-based method for the identification of multiple rumor sources. They use knowledge from infection and influence models in order to model the problem of rumors that spread in the social network. The input to their mode is a set of different infection states of a given network and the output is the actual rumor sources. Their method is based on the fact that rumor sources are surrounded by more infected nodes than normal sources, and non-infected nodes are usually far from the rumor sources. In order to capture the features of multi-order neighbor nodes, they stack multiple GCN layers that use RELU and a single dense layer that uses the sigmoid function. They consequently modify GCN to be directly applied to the supervised learning task of detecting rumor sources. They evaluate their method on popular benchmark social networks in which they define infection sources that the method has to detect and apply different propagation models.

The problem formulation in this case begins with a set of social media users (accounts) $U = \{u_1, \dots, u_n\}$, who connect to each other with social links and have social and textual metadata associated with them based on the posts they make, their profile information and the social features (e.g., connectivity metrics) they have. As shown in Figure 4, in the pre-processing step, the context of user profile and user posts is processed with language models (e.g., transformers) to produce the textual information embeddings, which are concatenated with the numerical and categorical social features. The vectors are attached to the nodes $v \in V$ of the directed graph $\mathcal{G} = V, E$, which has users as nodes $u \in V$ and their social links as edges $e \in E$.

Once again, the concatenated textual embeddings and social features constitute the feature vector R of each node, and the vectors of all nodes form the input matrix $X \in \mathbb{R}^{n \times d}$, where n is the number of nodes and d is the dimensionality of the node vectors. The adjacency matrix $A \in \mathbb{R}^{n \times n}$ is used to represent the set of edges E and the same graph convolution and max pooling steps are applied, as in the case of news classification, to classify

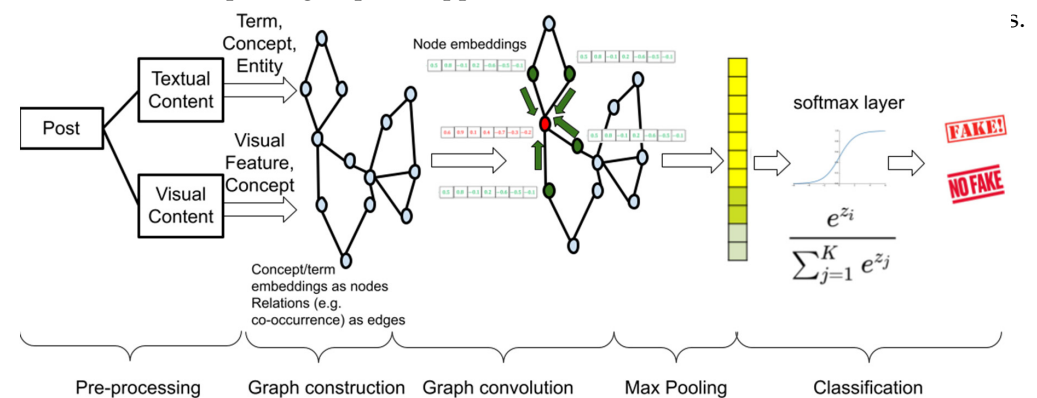


Figure 4. An overview of the spammer detection process using GCN.

3.3. Combining Fake Characteristics with Dispersion Information Using GCN

In an attempt to combine the power of GCNs to represent and capture the structural dimension of fake news and the ability of Recursive Neural Networks (RNNs) to model the way news is spread on social media, Bian et al. [38] proposed a Bi-Directional GCN to explore the propagation and dispersion characteristics of rumors in tandem. Consequently, they model the propagation and dispersion chains of posts using a graph, with the original post and all the subsequent posts related to it as vertices, and directed edges between posts that represents the response relation between a post and any other post that refers to it (or it refers to). The original post and consequently the resulting graphs are labeled as False or True rumors. The authors also applied the DropEdge technique to reduce the

over-fitting of their GCNs. The high-level node representations from the two GCNs are concatenated along with the hidden feature vector of the root node using mean-pooling operators. Several full connection layers and a softmax layer are then employed to predict the label of the event.

In a similar approach, Li and Goldwasser [39] construct the social information graph, which consists of vertices that correspond: (i) to political users of Twitter, (ii) other Twitter users that spread content and (iii) news items that are shared by them. The edges of the graph correspond to follower relations between political users and simple users that follow them and post relations between Twitter users and the news items they share. They evaluate both first-order graph embeddings that capture the direct relationships between the graph nodes of different types but also employ a two-layer GCN for node classification. This allows them to classify news items as fake or not using the social representation and the resulting embedding from the GCN.

The authors in [40] attempt to represent rumor propagation using a dynamic graph and consequently propose a Dynamic GCN approach for detecting rumors in social media. In their approach, they take graph snapshots and perform representation learning using an attention mechanism that captures both structural and temporal information of rumor diffusion. The proposed method captures the evolving pattern of the rumor diffusion using a series of snapshots of the same propagation graph (actually a tree) that has posts (and post feature vectors) as nodes and edges that denote the response of a post from another post. Sequential snapshots comprising an increasing number of nodes and edges in each step or temporal snapshots that contain a varying number of additional vertices and edges between snapshots are both examined. A two-layer bi-directional GCN with ReLU as an activation function and a global graph pooling (mean pooling) layer is used to convert node representation to graph the representation, and an additive (or dot-product) attention mechanism is employed to retrieve a global graph embedding that re-weights the graph snapshot embeddings. The output graph embedding is fed to a multi-layer perceptron for classifying the propagation graph as rumor-related or not.

In the same direction, the authors in [41] create a reply tree and a user graph based on the originally posted item (tweet) and the items that respond to it (reply tweets) and employ GCNs to detect rumor conversations. Both the reply tree and the user graph have directed edges based on the timestamp of tweets and their nodes are tweets and Twitter users, respectively. The user nodes are characterized by features such as the number of followers and friends, the number of user tweets liked by others, the user's verified identity on Twitter, the user's profile description, etc. The high-frequency words of the source tweets and the propagation time interval between the reply tweets and the source tweet are the two features in the nodes of the reply tree. Two graph convolution layers with an intermediate dropout layer to avoid overfitting, a global max pooling layer and a fully connected layer on top are employed in their model to process each graph separately and generate the respective vector representations. These representations are concatenated and are given to a fully connected layer to calculate the label of the conversation.

In a different approach that does not employ GCNs but focuses on the news' propagation path, the work in [42] emphasizes the detection of fake news in the early stages of their propagation in social networks. Each propagation path is modeled as a multivariate time series, where a numerical vector at each time-step represents the characteristics of a user who engaged in spreading the news. A time-series classifier that combines Recurrent and Convolutional Neural Networks is employed for classifying the propagation path as fake or not. More specifically, Gated Recurrent Units (GRUs) process the information of the source tweet and its retweets and are maxed pooled to create a vector representation of the news' spreading path. A similar neural network structure, with Convolutional Neural Network units (CNNs) instead of GRUs, generates a second representation of the news' spreading path. The two vectors are concatenated and fed into a multi-layer feedforward neural network that finally predicts the class label for the corresponding propagation path.

The works that focus on the detection of rumors in social media combine information from the social graph formed between the users and the conceptual graph that is formed by the content they share and the concepts it conveys. The set of source news items $S = \{s_1, s_2, \dots, s_n\}$ that start the rumors and all the subsequent posts P from the users of the social network form the rumors dataset. As depicted in Figure 5, the input in this case comprises: i) a social graph $\mathcal{G}_S = V, E$, with users as nodes $u \in V$ and their social links as (usually undirected) edges $e \in E$ and ii) a set of rumor propagation graphs in the form of $\mathcal{G}_i = V, E_i$, each one corresponding to the propagation of a news item s_i in the social network. A directed edge $e = (u \rightarrow v) \in E_i$, denotes that user $u \in V$ shared/re-posted/commented/responded s_i , influenced by user $v \in V$. The adjacency matrix of this graph is denoted with A_i . The text embeddings extracted from s_i and the posts associated with it constitute the feature matrix X_i of the rumor propagation graph.

Once the graph is constructed, the convolution and classification steps are the same as before. The classification task can be binary (e.g., rumor or not) or multi-class by distinguishing between true or false rumors, non-rumors and unverified ones.

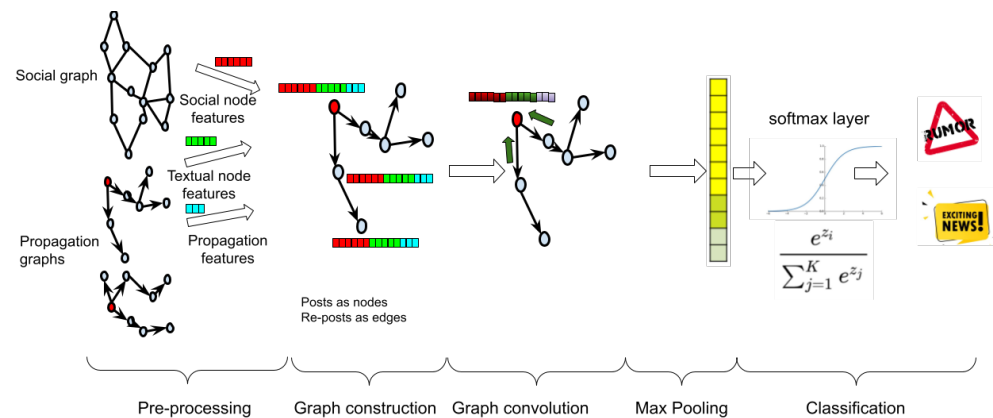


Figure 5. An overview of the rumor detection process using GCN.

4. Datasets

The works presented in this survey either employ some standalone (i.e., without link or any other dispersion information) text corpora and classify them in a binary classification task or employ Twitter for creating custom data collections. In the latter case, they take advantage of the fact that Twitter provides both the textual content of tweets, as well as the profile information of Twitter users and information about the retweets, which can be used to model the news dispersion.

In the case of news classification tasks, public real-world social media datasets have been employed in the literature such as:

- PHEME [43]: A Twitter dataset containing 5802 annotated tweets, of which 34% are rumors. A collection script is also made available by the authors (<https://github.com/azubiaga/pHEME-twitterconversation-collection>, accessed on 17 February 2022).
- Weibo [44]: 3.8 million weibo.com posts from 2.7 million users of Weibo discussing 4664 events, of which 50% are rumors.
- LIAR [45]: 12,836 labeled short statements with six fine-grained truthfulness labels, from totally fake to completely true.
- MediaEval Twitter [46]: A dataset with around 17,000 unique tweets spanning over different events, comprising 9000 fake news tweets, 6000 real news tweets in the development set and 2000 tweets in the test set. The tweets comprise text, images and video.
- Twitter15 and Twitter16 [47]: Comprising 331,612 posts from 276,663 users on 1490 events and 204,820 posts from 173,487 users on 818 events, respectively, classified in four rumor classes.

- Fakeddit [48]: This multimodal (text and images) dataset comprises over 1 million samples from multiple subreddits from Reddit, split into 2, 3 or 6 classes, along with comment data and metadata.

For the detection of bot or spam accounts that create rumors the list of datasets employed comprises:

- TwiBot-20 [49]: A publicly available Twitter bot detection dataset that provides the follow relationship between users for 230,000 users.
- RTbust [50]: A Twitter dataset comprising 9,989,819 retweets related to 1,691,865 distinct original Italian tweets shared by 1,446,250 distinct users. The dataset contains social bots and user-operated accounts and covers the need of OSN administrators that look for automated behaviors in retweeting in order to decide about banning accounts from social platforms.
- Botometer-feedback [51]: A mix of training data from eight previously published works containing 57,155 bot and 30,853 accounts.
- Gilani (<https://goo.gl/SigsQB>, accessed on 17 February 2022) [52]: Approximately 65 million tweets (2–2.5 million per day) from 2.9 million unique accounts.
- \$FAKE [53]: A dataset comprising stock microblogs collected from Twitter. A total of 6689 stocks were used to fetch the content from Twitter and resulted in 9 M tweets (22% are retweets) posted by 2.5 M distinct users. The tweets mention 30,032 companies for which financial information was collected from Google Finance.
- Midterm [54]: An amalgam of 14 other datasets that consists of 94,124 bot accounts (lured by honeypot accounts) and 43,396 verified human accounts.
- Twitter Social [55]: A dataset, which has been collected using 60 social honeypot accounts on Twitter that tweet normal posts, links or popular n-grams in order to tease their followers to retweet. The social honeypots tempted 36,043 Twitter users, 5773 (24%) of which followed more than one honeypot and 23,869 followed only one honeypot.
- Twitter 1KS-10KN [56]: A dataset comprising 485,721 Twitter accounts with 14,401,157 tweets and 5,805,351 URLs. From these accounts, 10,004 are considered malicious affected accounts that post malicious URLs.
- Altmetrics (https://github.com/slab-itu/altmetrics_bot, accessed on 17 February 2022) [33]: A dataset with 457,714 tweets posted by a total of 16,264 unique users, of which 64 are bots. The 31,380 graph edges correspond to retweets or mentions between users. The dataset is a merging of the datasets used in [57,58].

In some cases, original news articles are employed as the primary source of a fake news disinformation campaign. For example, the authors in [39] collected 10,385 news articles from two news aggregation websites on different events in 2020 discussing 94 event types, such as elections, terrorism, etc. They also collected information from 1604 highly active Twitter users, who follow known political Twitter accounts (135 accounts were used as a seed) and frequently share political news.

5. Code Repositories

When performing research in a new field, it is important for researchers to have access to useful and reusable resources. This includes datasets, as well as code, model and algorithm implementations, that can be employed. Luckily, in the case of GCNs, there are many resources, and in the following, we try to compile a list of the most useful code repositories and their content.

Authors in [36] provide a link to their code repository (<https://github.com/shakshi12/Rumor-Spreaders-using-GNN-approach-PHEME-dataset->, accessed on 17 February 2022), which employs NLTK (<https://www.nltk.org/>, accessed on 17 February 2022) for pre-processing the English tweets and extracting their sentiment and Gensim (<https://radimrehurek.com/gensim/>, accessed on 17 February 2022) to obtain word embeddings. They also employ the NetworkX (<https://networkx.org/>, accessed on 17 February 2022) library to construct

the graph and the Spektral (<https://graphneural.network/>, accessed on 17 February 2022) python library to perform the Graph Convolution.

Authors in [29] provide a link to a useful code repository (https://github.com/nikhilmaram/Show_and_Tell, accessed on 17 February 2022) for extracting visual features from images with the image2sentence model, whereas in [33], the authors share the code and data for Twitter bot detection using GCNs on github (https://github.com/slab-itu/altmetrics_bot, accessed on 17 February 2022). Their python code employs NetworkX for graph creation and a TensorFlow implementation of the Graph Convolution Layer. The work performed in [34] can be replicated using the shared data and code (<https://github.com/shanmon110/BGSRD>, accessed on 17 February 2022), which employs the BERT model to convert documents into a heterogeneous graph that contains both documents and words as nodes and edges that correspond to the global word co-occurrence (i.e., PMI) in the corpus.

In the context of rumor detection, code (<https://github.com/jihochoi/dynamic-gcn>, accessed on 17 February 2022) and data (<https://figshare.com/s/d8984fd39557a3d295e8>, accessed on 17 February 2022) are provided by [40]. The code for the Dynamic GCNs is developed on PyTorch Geometric, and several execution parameters allow trying different snapshot numbers, attention modules and datasets. When it comes down to evolving graphs and handling graph snapshots [59], the EvolveGCN (<https://github.com/IBM/EvolveGCN>, accessed on 17 February 2022) code can be modified to capture the evolution of the rumor spreading graph.

Other useful resources include the original TensorFlow implementation of GCNs (<https://github.com/tkipf/gcn>, accessed on 17 February 2022) by Thomas Kipf, the Deep Graph Library (<https://www.dgl.ai/>, accessed on 17 February 2022) built on PyTorch, TensorFlow and MXNet and the PyTorch Geometric (<https://pytorch-geometric.readthedocs.io/>, accessed on 17 February 2022) that is built on PyTorch.

6. Discussion

Table 1 summarizes the methods surveyed in this work, which capitalize on the use of Graph Convolutional Networks, which are broadly divided into three main groups:

- Text classification approaches that map post words or whole posts as nodes in the graph and use edges to denote their semantic or positional relations (i.e., co-occurrence, posted by similar users.). They consequently classify graphs using the representations learned using GCNs.
- User/source classification approaches that map social network accounts to nodes in the graph and use edges to denote their social relations. The GCNs are used to classify sources as fake/bot/spammers or not.
- Propagation graph classification approaches that classify the propagation graphs (either the final graph or several snapshots of it) as fake or not using GCNs.

The majority of the works are based on assumptions about the underlying publish and propagation models, for example:

- A fake post employs entities that do not usually appear together in other posts;
- The productivity or re-production rate of a bot or spam account is much higher than that of a normal account;
- The nodes that are near (within the graph) to a spam or rumor producing account, are more affected than the nodes that are far from the node;
- A fake news propagation graph has different characteristics (fanout, density, etc.) from that of a legitimate news one.

Table 1. The summary of methods that use GCN for detecting fake news, fake accounts, fake news sources or fake news propagation networks.

Detection Task	Input	Graph(s)	Nodes	Edges	GCN type	Citation	Dataset
Fake content	post text and images	undirected, weighted	post words, concepts, image labels	PMI based node similarity	Knowledge-driven Multimodal GCN	[27]	PHEME, Weibo
	post, user profile	undirected, unweighted	posted documents	based on user profile similarity	Multi-Depth GCN	[28]	LIAR
	post text and images	(3x) undirected, weighted	words, concepts	PMI based node similarity	GCN + VGG-19 for images	[29]	Weibo, MediaEval, PHEME
	posts, users, reviews, domains	undirected, unweighted	news, domains, reviews, sources	between news based on content similarity, between news and other node types	GCN	[30]	Weibo, Fakeddit
Bot detection	user profile, posts, neighbor profiles	directed, unweighted	users	follow	GCN	[31]	Twibot-20
	posts, users	undirected, unweighted	users	retweets, mentions	GCN and altmetrics	[33]	Altmetrics
	posts	undirected, weighted	words, posts	word-post (TF-IDF) word-word (PMI)	GCN + BERT	[34]	RTbust, Gilani, Botometer-feedback, \$FAKE, Midterm
	social graph	directed, unweighted	users	bidirectional, unidirectional incoming & outgoing	GCN + MRF	[35]	Twitter Social, Twitter 1KS-10KN
Rumour source detection	social graph infection snapshots	undirected, unweighted	users	social relations	modified GCN	[37]	General purpose social graphs
	social graph	undirected, unweighted	users	social relations	GCN	[36]	PHEME
Fake news propagation	propagation & dispersion chains of posts	directed, unweighted	original post, reposts, responses	reference between posts	Bi-Directional GCN	[38]	Weibo, Twitter15, Twitter16
	news items, propagation graphs	undirected, unweighted	political users, twitter users, posts	users follow political accounts, users post news	GCN	[39]	Custom dataset
	Graph snapshots	directed, unweighted	posts, reposts	reference between posts	Dynamic bi-directional GCN	[40]	Twitter15, Twitter16, Weibo
	reply tree, user graph	directed, unweighted	posts, replies users	post references, follow	GCN	[41]	PHEME

The methods proposed so far take advantage of the core social network information, employ language models, such as BERT or word embeddings, in order to produce comprehensive text representations, and, in some cases, employ multimodal information to support the classification task. In the case of content classification, images are either processed using parallel deep neural network architectures, and the resulting representations are concatenated with the text representations before being fed to the GCN. In the case of source or propagation graph classification, the emphasis is on features that can enhance the user profile information, such as features from the content of the post, the graph structure or the evolution of the propagation graph over consecutive snapshots.

Most of the methods generate homogeneous uni-partite graphs, which either comprises word or user nodes interconnected with edges based on similarity or on social relations, respectively. Some works introduce the use of heterogeneous graphs that comprise both textual nodes (words, concepts or posts) and user nodes who are connected with directed or undirected edges depending on the case. Different edge weighting schemes are employed in order to enhance the graph edges with additional information concerning the strength of node relations, whereas a multitude of features are attached to the graph nodes in an attempt to enrich their representation.

The surveyed methods usually stack from two up to four GCN layers in order to capture higher-order neighborhood information in the graph and an additional dense layer to solve the respective classification task. GCNs allow combining the embeddings of each node with those of its first, second or higher-order neighbors. Thus, in the case of heterogeneous networks, such as those that comprise posts and users, it allows learning how the posts of a user's neighbors affect the credibility of the user or how the neighborhood of a user affects the credibility of the content she/he posts. In the case of homogeneous networks with directed edges, GCNs learn how a user is affected by her/his followers or the people that follow them or how the terms or concepts in fake news items affect the veracity of other news that contains the same terms or terms related to them. The learned parameters are the node embeddings and the weights of all neighbors within each layer.

As shown in Figure 6, the overall architecture of the GCN comprise one or more graph convolution layers, which are responsible for aggregating information from its neighboring nodes in each node. The GC layers are followed by pooling layers that coarse the graph into sub-graphs, with nodes that represent higher graph-level representations. A readout layer with a mean, max, sum or any other aggregation function allows summarizing the final node representations in a single representation for the whole graph. This allows classifying the whole graph (which may correspond to a text and its context, a user and their neighbors and profiles or a rumor propagation graph) to one of many class labels using multiple dense layer and a softmax layer at the end.

It is possible to compare approaches that employ different techniques when they are evaluated on the same dataset and the same task. For example, the KMGCN proposed in [27] achieved an accuracy of 0.886 on the Weibo dataset, whereas the KMAGCN presented in [29] using a pre-trained XLnet improved the accuracy to 0.944. This improvement is mainly due to the use of visual features, which are captured using VGG-19. On the other side, the heterogeneous graph in [30] allows the HDGCN method to achieve an accuracy as high as 0.961 on the same dataset.

Experiments on the Weibo dataset for the detection of rumors also demonstrated the ability of GCNs to predict such cases early. More specifically, the combined use of two GCNs (i.e., one for the propagation and one for the dispersion graph) in the Bi-GCN early stopping method achieves a prediction accuracy of 0.961, as reported in [38]. The same model has been evaluated in [40] on the same dataset using a different train-validation-test split and 10 runs, and the reported average accuracy for Bi-GCN was 0.928. In addition, the proposed early stopping method of this work, DYNGCN, improved the accuracy to 0.936 in the same experimental setup. The pattern of results is the same in both papers for the Twitter 15 and Twitter 16 datasets. The main advantage of DYNGCN over the Bi-GCN is that it trains a different GCN for each snapshot of the rumor diffusion graph and adds

an attention layer that weights the importance of the graph snapshot embeddings. Both additive and dot-product attention mechanisms have been tested with the additive model to provide slightly better results in all datasets.

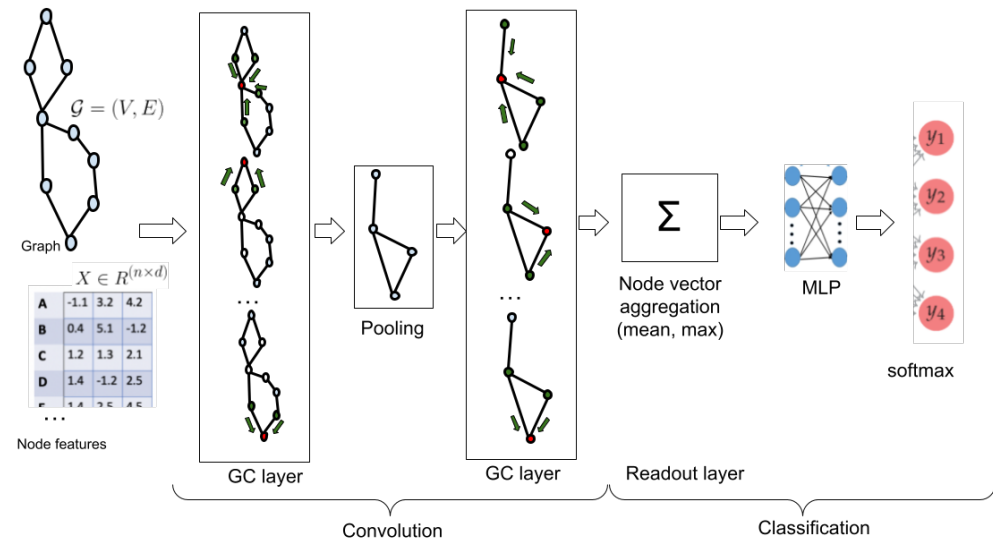


Figure 6. The architecture of GCNs in a generic graph classification task.

Another straight comparison of the various methods can be performed on the PHEME dataset. In [27], the authors report an accuracy of 0.8756 with their best variant of KMGCN, whereas the authors in [29] report a slightly worse accuracy of 0.867 for the KMGCN method using BERT. The same dataset has been used in [36,41] but for a completely different task, which is the identification of rumor spreaders. Both works have employed specific subsets of the dataset that refer to breaking news and, more specifically, the Charlie Hebdo shooting, Ferguson unrest, Ottawa shooting, Sydney hostage crisis and Germanwings crash. The GCN method used in [36] achieved an accuracy of 0.790, 0.705, 0.675, 0.655 and 0.715, respectively, in each subset. The User-Reply-GCN of [41] was tested on the same dataset but with a random oversampling technique that changed the percentage ratio between rumors and non-rumors. In this case, the authors reported an overall F1-score of 0.80 for the rumor class in all subsets and 0.79 for the non-rumor class.

Compared to the approaches that use simple Convolutional Neural Networks, GCNs share the same intuition since they try to learn the relationships between neighboring nodes similarly to learning the relationships between neighboring pixels. However, CNNs rely on two-dimensional data (pixels) with a specific structure and positioning, whereas GCNs can work with any kind of arbitrary structured data (nodes). Recurrent Neural Networks can ideally be combined with GCNs in order to capture the evolution of dynamic graphs [59] and have several applications ranging from scene perception and action-recognition [60] to traffic prediction [61]. This combination is very promising in the case of fake news that propagate through social networks and have very recently been applied [40] in their detection task. From a design point of view, they seem to be more appropriate than earlier RNN-CNN combinations that converted the propagation paths to multivariate time-series [42].

7. Conclusions

The different GCN techniques presented in this article demonstrate the advantages of graph-based representations and GCNs over simple CNNs or RNNs in graph and node classification tasks. The graphs better capture the latent relations between concepts in the news text, between the users that spread the news or between users and the texts they share. The graph convolution allows communicating this information between the graph nodes and capturing the effect of social relations (neighborhood) in the dispersion of fake news.

Finally, temporal graph-convolution techniques manage to better capture the evolution of the news spreading graphs and detect rumors early.

On the opposite side of their good performance, GCN-based methods suffer from high complexity, e.g., for the computation of the decomposition of the Laplacian matrix. Even when the eigenvectors are pre-computed or rank approximations of the eigenvalue decomposition are employed, the complexity is high, especially for large-scale graphs. Another major issue that is evident in all works is the large amount of hyper-parameters that must be learned in each case. The methods usually update this parameters in an end-to-end learning manner, which limits their modularity and the ability to reuse certain components of each architecture in a different task. The work in [8] performs an in-depth discussion of GCNs and their scalability and complexity issues.

The importance for component reusability is evident in many of the works that employ pre-trained language models (e.g., BERT, XLNet) for text pre-processing or pre-trained CNNs (e.g., VGG-19) to extract visual features. With this in mind, pre-trained graph embeddings, such as the knowledge graph embedding of DGL-KE (<https://github.com/awslabs/dgl-ke>, accessed on 17 February 2022), could be an interesting addition and a useful component for developing graph-based solutions for fake news, spammers or rumor detection.

An issue reported in the works surveyed in this study is the difficulty in extracting and handling the visual content of fake news and in associating the visual features with the respective concepts that appear in the textual content. The use of pre-trained models definitely assists feature and concept extraction, but the association of features with concepts still requires training for fine-tuning the different hyper-parameters. Another possible pitfall for the GCN model that employs the early stopping criterion is the need for an additional validation set for model selection. In the absence of a validation dataset, a GCN optimized on the training data will probably only have a sharp drop in performance, especially when the training size is small [62].

Future research in the field has to experiment with more variants of the GCN that incorporate multimodal content information [63] (using word, document and image embeddings [64]), graph structure information (in the form of graph embeddings [65]) and side features for nodes and edges. Global graph pooling techniques [66] or any other structured [67] or hierarchical [68] graph pooling method that can capture local properties of the graph formed around a fake news source have to be examined in combination with GCN in order to improve the current state of the art. Finally, all the possible GCN issues [69] must be considered under the prism of the specific fake news and rumor detection tasks.

Author Contributions: Conceptualization, I.V. and D.M.; methodology, F.G.; investigation, F.G.; writing—original draft preparation, I.V. and D.M.; writing—review and editing, I.V.; supervision, I.V.; project administration, P.T.; funding acquisition, P.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was co-funded by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH - CREATE - INNOVATE. Grant number: T2EDK-5055943. This research has been co-funded by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH—CREATE—INNOVATE. Grant number: T1EDK-03470.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GCN	Graph Convolutional Network
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
BERT	Bidirectional Encoder Representations from Transformers
PMI	Pointwise Mutual Information
MRF	Markov Random Field

References

1. Allcott, H.; Gentzkow, M. Social media and fake news in the 2016 election. *J. Econ. Perspect.* **2017**, *31*, 211–236. [\[CrossRef\]](#)
2. Ha, L.; Andreu Perez, L.; Ray, R. Mapping recent development in scholarship on fake news and misinformation, 2008 to 2017: Disciplinary contribution, topics, and impact. *Am. Behav. Sci.* **2021**, *65*, 290–315. [\[CrossRef\]](#)
3. Tasnim, S.; Hossain, M.M.; Mazumder, H. Impact of rumors and misinformation on COVID-19 in social media. *J. Prev. Med. Public Health* **2020**, *53*, 171–174. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Keller, F.B.; Schoch, D.; Stier, S.; Yang, J. Political astroturfing on twitter: How to coordinate a disinformation campaign. *Political Commun.* **2020**, *37*, 256–280. [\[CrossRef\]](#)
5. Benkler, Y.; Faris, R.; Roberts, H. *Network Propaganda: Manipulation, Disinformation, and Radicalization in American Politics*; Oxford University Press: Oxford, UK, 2018.
6. Vosoughi, S.; Roy, D.; Aral, S. The spread of true and false news online. *Science* **2018**, *359*, 1146–1151. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Zhang, S.; Tong, H.; Xu, J.; Maciejewski, R. Graph convolutional networks: Algorithms, applications and open challenges. In Proceedings of the International Conference on Computational Social Networks, Shanghai, China, 18–20 December 2018; pp. 79–91.
8. Zhang, S.; Tong, H.; Xu, J.; Maciejewski, R. Graph convolutional networks: A comprehensive review. *Comput. Soc. Netw.* **2019**, *6*, 1–23. [\[CrossRef\]](#)
9. Shu, K.; Sliva, A.; Wang, S.; Tang, J.; Liu, H. Fake News Detection on Social Media: A Data Mining Perspective. *ACM SIGKDD Explor. Newsl.* **2017**, *19*, 22–36. [\[CrossRef\]](#)
10. Cardoso Durier da Silva, F.; Vieira, R.; Garcia, A.C. Can machines learn to detect fake news? A survey focused on social media. In Proceedings of the 52nd Hawaii International Conference on System Sciences, Maui, HI, USA, 8–11 January 2019.
11. Zhou, X.; Zafarani, R. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–40. [\[CrossRef\]](#)
12. Varma, R.; Verma, Y.; Vijayvargiya, P.; Churi, P.P. A systematic survey on deep learning and machine learning approaches of fake news detection in the pre-and post-COVID-19 pandemic. *Int. J. Intell. Comput. Cybern.* **2021**, *14*, 617–646. [\[CrossRef\]](#)
13. Hindman, M.; Barash, V. *Disinformation, and Influence Campaigns on Twitter*; Knight Foundation: George Washington University: Washington, DC, USA, 2018.
14. Freelon, D.; Lokot, T. Russian Twitter Disinformation Campaigns Reach across the American Political Spectrum. *Misinformation Review*, 14 January 2020.
15. Recuero, R.; Soares, F.B.; Gruz, A. Hyperpartisanship, disinformation and political conversations on Twitter: The Brazilian presidential election of 2018. In Proceedings of the International AAAI Conference on Web and Social Media, Atlanta, GA, USA, 8–11 June 2020; Volume 14; pp. 569–578.
16. Tardelli, S.; Avvenuti, M.; Tesconi, M.; Cresci, S. Detecting inorganic financial campaigns on Twitter. *Inform. Syst.* **2021**, *103*, 101769. [\[CrossRef\]](#)
17. Tardelli, S.; Avvenuti, M.; Tesconi, M.; Cresci, S. Characterizing social bots spreading financial disinformation. In Proceedings of the International Conference on Human-Computer Interaction, Copenhagen, Denmark, 19–24 July 2020; pp. 376–392.
18. Nizzoli, L.; Tardelli, S.; Avvenuti, M.; Cresci, S.; Tesconi, M.; Ferrara, E. Charting the landscape of online cryptocurrency manipulation. *IEEE Access* **2020**, *8*, 113230–113245. [\[CrossRef\]](#)
19. Cresci, S.; Lillo, F.; Regoli, D.; Tardelli, S.; Tesconi, M. Cashtag piggybacking: Uncovering spam and bot activity in stock microblogs on Twitter. *ACM Trans. Web (TWEB)* **2019**, *13*, 1–27. [\[CrossRef\]](#)
20. Orabi, M.; Mouheb, D.; Al Aghbari, Z.; Kamel, I. Detection of bots in social media: A systematic review. *Inf. Process. Manag.* **2020**, *57*, 102250. [\[CrossRef\]](#)
21. Freitas, C.; Benevenuto, F.; Ghosh, S.; Veloso, A. Reverse engineering socialbot infiltration strategies in twitter. In Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Paris, France, 25–28 August 2015; pp. 25–32.
22. Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 13–17 August 2016; pp. 855–864.
23. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.
24. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.

25. Yao, L.; Mao, C.; Luo, Y. Graph convolutional networks for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February, 2019; Volume 33, pp. 7370–7377.
26. Yan, S.; Xiong, Y.; Lin, D. Spatial temporal graph convolutional networks for skeleton-based action recognition. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
27. Wang, Y.; Qian, S.; Hu, J.; Fang, Q.; Xu, C. Fake news detection via knowledge-driven multimodal graph convolutional networks. In Proceedings of the 2020 International Conference on Multimedia Retrieval, Dublin, Ireland, 8–11 June 2020; pp. 540–547.
28. Hu, G.; Ding, Y.; Qi, S.; Wang, X.; Liao, Q. Multi-depth graph convolutional networks for fake news detection. In Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing, Dunhuang, China, 9–14 October 2019; pp. 698–710.
29. Qian, S.; Hu, J.; Fang, Q.; Xu, C. Knowledge-aware multi-modal adaptive graph convolutional networks for fake news detection. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2021**, *17*, 1–23. [\[CrossRef\]](#)
30. Kang, Z.; Cao, Y.; Shang, Y.; Liang, T.; Tang, H.; Tong, L. *Fake News Detection with Heterogenous Deep Graph Convolutional Network*; PAKDD (1); Springer: Berlin, Germany 2021; pp. 408–420.
31. Feng, S.; Wan, H.; Wang, N.; Luo, M. BotRGCN: Twitter Bot Detection with Relational Graph Convolutional Networks. *arXiv* **2021**, arXiv:2106.13092.
32. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
33. Aljohani, N.R.; Fayoumi, A.; Hassan, S.U. Bot prediction on social networks of Twitter in altmetrics using deep graph convolutional networks. *Soft Comput.* **2020**, *24*, 11109–11120. [\[CrossRef\]](#)
34. Guo, Q.; Xie, H.; Li, Y.; Ma, W.; Zhang, C. Social Bots Detection via Fusing BERT and Graph Convolutional Networks. *Symmetry* **2022**, *14*, 30. [\[CrossRef\]](#)
35. Wu, Y.; Lian, D.; Xu, Y.; Wu, L.; Chen, E. Graph convolutional networks with Markov random field reasoning for social spammer detection. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 1054–1061. [\[CrossRef\]](#)
36. Sharma, S.; Sharma, R. Identifying possible rumor spreaders on twitter: A weak supervised learning approach. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–8.
37. Dong, M.; Zheng, B.; Quoc Viet Hung, N.; Su, H.; Li, G. Multiple rumor source detection with graph convolutional networks. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, New York, NY, USA, 3–7 November 2019; pp. 569–578.
38. Bian, T.; Xiao, X.; Xu, T.; Zhao, P.; Huang, W.; Rong, Y.; Huang, J. Rumor detection on social media with bi-directional graph convolutional networks. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 549–556. [\[CrossRef\]](#)
39. Li, C.; Goldwasser, D. Encoding social information with graph convolutional networks for political perspective detection in news media. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 2594–2604.
40. Choi, J.; Ko, T.; Choi, Y.; Byun, H.; Kim, C.k. Dynamic graph convolutional networks with attention mechanism for rumor detection on social media. *PLoS ONE* **2021**, *16*, e0256039.
41. Lotfi, S.; Mirzarezaee, M.; Hosseinzadeh, M.; Seydi, V. Detection of rumor conversations in Twitter using graph convolutional networks. *Appl. Intell.* **2021**, *51*, 4774–4787. [\[CrossRef\]](#)
42. Liu, Y.; Wu, Y.F.B. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
43. Zubiaga, A.; Liakata, M.; Procter, R. Exploiting context for rumour detection in social media. In Proceedings of the International Conference on Social Informatics, Oxford, UK, 13–15 September 2017; pp. 109–123.
44. Ma, J.; Gao, W.; Mitra, P.; Kwon, S.; Jansen, B.J.; Wong, K.F.; Cha, M. Detecting rumors from microblogs with recurrent neural networks. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016.
45. Wang, W.Y. “liar, liar pants on fire”: A new benchmark dataset for fake news detection. *arXiv* **2017**, arXiv:1705.00648.
46. Khattar, D.; Goud, J.S.; Gupta, M.; Varma, V. Mvae: Multimodal variational autoencoder for fake news detection. In Proceedings of the World Wide Web Conference, New York, NY, USA, 13–17 May 2019; pp. 2915–2921.
47. Ma, J.; Gao, W.; Wong, K.F. *Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017.
48. Nakamura, K.; Levy, S.; Wang, W.Y. r/fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection. *arXiv* **2019**, arXiv:1911.03854.
49. Feng, S.; Wan, H.; Wang, N.; Li, J.; Luo, M. TwiBot-20: A Comprehensive Twitter Bot Detection Benchmark. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Gold Coast, QLD, Australia, 1–5 November 2021; pp. 4485–4494.
50. Mazza, M.; Cresci, S.; Avvenuti, M.; Quattrociocchi, W.; Tesconi, M. Rtbust: Exploiting temporal patterns for botnet detection on twitter. In Proceedings of the 10th ACM Conference on Web Science, New York, NY, USA, 27–30 May 2019; pp. 183–192.

51. Yang, K.C.; Varol, O.; Davis, C.A.; Ferrara, E.; Flammini, A.; Menczer, F. Arming the public with artificial intelligence to counter social bots. *Hum. Behav. Emerg. Technol.* **2019**, *1*, 48–61. [\[CrossRef\]](#)
52. Gilani, Z.; Farahbakhsh, R.; Tyson, G.; Wang, L.; Crowcroft, J. Of bots and humans (on twitter). In Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, New York, NY, USA, 31 July–3 August 2017; pp. 349–354.
53. Cresci, S.; Lillo, F.; Regoli, D.; Tardelli, S.; Tesconi, M. \$FAKE: Evidence of spam and bot activity in stock microblogs on Twitter. In Proceedings of the Twelfth International AAAI Conference on Web and Social Media, Stanford, CA, USA, 25–28 June 2018.
54. Yang, K.C.; Varol, O.; Hui, P.M.; Menczer, F. Scalable and generalizable social bot detection through data selection. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 1096–1103. [\[CrossRef\]](#)
55. Lee, K.; Eoff, B.D.; Caverlee, J. Seven months with the devils: A long-term study of content polluters on twitter. In Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, Barcelona, Spain, 17–21 July 2011.
56. Yang, C.; Harkreader, R.; Zhang, J.; Shin, S.; Gu, G. Analyzing spammers' social networks for fun and profit: A case study of cyber criminal ecosystem on twitter. In Proceedings of the 21st International Conference on World Wide Web, New York, NY, USA, 16–20 April 2012; pp. 71–80.
57. Didegah, F.; Mejlgaard, N.; Sørensen, M.P. Investigating the quality of interactions and public engagement around scientific papers on Twitter. *J. Informetr.* **2018**, *12*, 960–971. [\[CrossRef\]](#)
58. Hassan, S.U.; Imran, M.; Gillani, U.; Aljohani, N.R.; Bowman, T.D.; Didegah, F. Measuring social media activity of scientific literature: An exhaustive comparison of scopus and novel altmetrics big data. *Scientometrics* **2017**, *113*, 1037–1057. [\[CrossRef\]](#)
59. Pareja, A.; Domeniconi, G.; Chen, J.; Ma, T.; Suzumura, T.; Kanezashi, H.; Kaler, T.; Schardl, T.; Leiserson, C. Evolvegen: Evolving graph convolutional networks for dynamic graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 5363–5370. [\[CrossRef\]](#)
60. Si, C.; Chen, W.; Wang, W.; Wang, L.; Tan, T. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 1227–1236.
61. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3848–3858. [\[CrossRef\]](#)
62. Li, Q.; Han, Z.; Wu, X.M. Deeper insights into graph convolutional networks for semi-supervised learning. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA 2–7 February 2018.
63. Giachanou, A.; Zhang, G.; Rosso, P. Multimodal fake news detection with textual, visual and semantic information. In Proceedings of the International Conference on Text, Speech, and Dialogue, Olomouc, Czech Republic, 8–11 September 2020; pp. 30–38.
64. Khrulkov, V.; Mirvakhabova, L.; Ustinova, E.; Oseledets, I.; Lempitsky, V. Hyperbolic image embeddings. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6418–6428.
65. Hamdi, T.; Slimi, H.; Bounhas, I.; Slimani, Y. A hybrid approach for fake news detection in twitter based on user features and graph embedding. In Proceedings of the International Conference on Distributed Computing and Internet Technology, Bhubaneswar, India, 9–12 January, 2020; pp. 266–280.
66. Wang, Z.; Ji, S. Second-order pooling for graph neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**. [\[CrossRef\]](#)
67. Yuan, H.; Ji, S. Structpool: Structured graph pooling via conditional random fields. In Proceedings of the 8th International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
68. Zhang, Z.; Bu, J.; Ester, M.; Zhang, J.; Yao, C.; Yu, Z.; Wang, C. Hierarchical graph pooling with structure learning. *arXiv* **2019**, arXiv:1911.05954.
69. Shchur, O.; Mumme, M.; Bojchevski, A.; Günnemann, S. Pitfalls of graph neural network evaluation. *arXiv* **2018**, arXiv:1811.05868.