Unsupervised Fake News Detection: A Graph-based Approach

Siva Charan Reddy Gangireddy
sivag@iiitd.ac.in
IIIT Delhi, India

Deepak P
deepaksp@acm.org
Queen's University Belfast, UK

Cheng Long
c.long@ntu.edu.sg
Nanyang Technological University, Singapore

Tanmoy Chakraborty
tanmoy@iiitd.ac.in
IIIT Delhi, India

ABSTRACT

Fake news has become more prevalent than ever, correlating with the rise of social media that allows every user to rapidly publish their views or hearsay. Today, fake news spans almost every realm of human activity, across diverse fields such as politics and healthcare. Most existing methods for fake news detection leverage supervised learning methods and expect a large labelled corpus of articles and social media user engagement information, which are often hard, time-consuming and costly to procure. In this paper, we consider the task of *unsupervised fake news detection*, which considers fake news detection in the absence of labelled historical data. We develop GTUT, *a graph-based approach* for the task which operates in three phases. Starting off with identifying a seed set of fake and legitimate articles exploiting high-level observations on inter-user behavior in fake news propagation, it progressively expands the labelling to all articles in the dataset. Our technique draws upon graph-based methods such as biclique identification, graph-based feature vector learning and label spreading. Through an extensive empirical evaluation over multiple real-world datasets, we establish the improved effectiveness of our method over state-of-the-art techniques for the task.

ACM Reference Format:
Siva Charan Reddy Gangireddy, Deepak P, Cheng Long, and Tanmoy Chakraborty. 2020. Unsupervised Fake News Detection: A Graph-based Approach. In *Proceedings of the 31st ACM Conference on Hypertext and Social Media (HT '20), July 13–15, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3372923.3404783

1 INTRODUCTION

Fake news, a term that was barely used until a few years ago, has penetrated into the public discourse rapidly. The choice of post-truth and fake news as the word of the year by Oxford[1] and Collins[2] dictionaries respectively in consecutive years underlines widespread acknowledgment of its growing prominence. Fake News is commonly used to refer to deliberate presentation of misleading claims [11]. Of late, the news ecosystem has evolved from a small set of regulated and trusted sources to numerous online news sources and social media. Such new media sources come with limited liability for misinformation, and thus have been a major vehicle for fake news. It is also notable that the spread and impact of fake news is facilitated by intrinsic human tendencies such as confirmation bias [7] and the demonstrated inability of people to shrug off influence of fake news even after it having been debunked [29]. These, in the light of studies that suggest that an average US individual read one or more political fake news stories during the presidential election period in 2016 [2], have sparked off serious concerns on the threats that fake news pose to democracy and free debate.

Existing work on fake news detection largely relies on the usage of extensive historical labelled datasets and supervised learning methods that work over them. This is also true with related tasks within the same umbrella, such as rumour identification [34, 46] and hate speech [31]. Given that all of these tasks have a similar structure when viewed from a data science perspective, we will use *fake news detection* broadly to refer to all of these. Techniques for fake news detection in social media may be broadly seen as exploiting three kinds of information: (i) the (textual or other) content of the news item, (ii) the social network around the user who shares the news item, and (iii) temporal propagation information as gauged through re-tweets, re-shares and mentions. Some techniques have considered usage of only the temporal propagation information [23]; some others make use of both temporal and network information [39] while still not making use of content. While it may be safe to say that sophisticated usage of content has been lacking given most research has been driven by datasets from Twitter which have sparse content information, usage of content has ranged from shallow usage (e.g., counts of mentions, pictures and smileys [43], aggregated tf-idf [22]) to somewhat intricate modelling (e.g., pos-tag patterns [28] and psychological categories [20]).

While supervised methods are the natural first step towards addressing any labelling task, they suffer from several drawbacks. First, supervised methods require significant amounts of labelled data to learn meaningful models for effective detection of fake news. Obtaining manual annotations on veracity is labour-intensive as

[1] https://languages.oup.com/word-of-the-year/2016/

[2] https://www.independent.co.uk/news/uk/home-news/fake-news-word-of-the-year-2017-collins-dictionary-donald-trump-kellyanne-conway-antifa-corbynmania-a8032751.html

HT '20, July 13–15, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7098-1/20/07…$15.00
https://doi.org/10.1145/3372923.3404783

well as time-consuming, and efforts at crowdsourcing the data collection effort may entail quality deterioration [18]. Second, the usage of network and temporal features often lead to learning specific network and temporal behavioral patterns of individual users, sometimes making the effectiveness of the technique dependent on the permanence of users' network positioning and their behavioral patterns. Third, techniques that make use of content information would be affected by topic drift besides being unable to generalize across natural languages. For example, unless making use of people names is explicitly avoided, an algorithm tuned well on 2016 presidential elections data would be ill-suited for the 2020 edition.

Within that backdrop, We consider the task of *unsupervised fake news detection*. The unsupervised task is significantly more challenging than the supervised version given that unsupervised techniques do not have the luxury of labelled data to guide the search process. Thus, such methods would need to be embedded with high-level assumptions on the user dynamics around fake news. As an example, a recent work on unsupervised fake news detection [41] uses the presumption that *verified users* offer credible opinions on the veracity of a news piece. In contrast to [41], we rely on observations and insights from the *dynamics of inter-user behavior* and use that to design our method for unsupervised fake news detection. Specifically, our technique is designed using the assumption that synchronized behavior in news propagation is intrinsically more prevalent among fake news (vis-a-vis legitimate news). This assumption is motivated by two main observations; (i) that fake news is often driven by monetary, electoral or other gains, making it more likely to be propagated through blackmarket services where collusive user behavior is common [9], (ii) fake news is often driven by the expression of one's socio-political identity (as identified in a recent BBC study [5]) which would lead to implicit orchestrated behavior across specific cross-sections of the user base.

Our main contributions are as follows:

- We investigate, for the first time to our best knowledge, the usage of inter-user behavior dynamics for the task of unsupervised fake news detection.
- We propose an unsupervised learning method that is driven by graph-based techniques to identify fake and legitimate news.
- Through an extensive set of experiments across real-world datasets, we establish the effectiveness of our method over the state-of-the-art for unsupervised fake news detection.

2 RELATED WORK

We cover related work in fake news detection under several different heads as follows.

Unimodal Approaches. Majority of the studies on fake news detection are supervised in nature and focus only on the textual part, though some of them additionally use user metadata features. There are also attempts to detect the creator (user accounts) of such fake news. Past research that have successfully detected unreliable accounts include methods using clustering [40] and steganalysis [1] techniques. Researchers have also attempted with methods that target early detection [14]. In addition to this, techniques using random walk and entropy minimization discretization [10] were also experimented with. The other important aspect seen while forging the content is changing its writing style, which has impacts on the reader [27, 33]. This further led to addition of studying another important attribute that studies user opinions in detecting fake news. Another modality that has been popularly used in manipulating information present in news articles are images. The image splicing technique used in [15] takes into account the EXIF meta featues to detect fake images. Recent research has also shown the use of GANs [24] forged images in the news articles.

Multimodal Approaches. As may be expected, using multiple information modalities for a task, when available, would be more beneficial than just one. Towards this direction, experiments were conducted by extracting features from two different modalities such as text and image. Works on that direction include EANN [37], MVAE [17], SpotFake [36] and SpotFake+ [35]. The EANN model, short for event adversarial neural networks, for multimodal fake news detection proposed by [37] consists of three sub-modules namely, textual feature extractor, visual feature extractor and an event discriminator module that when combined together is successful in detecting fake news. Inspired by [37], [17] built a similar architecture, titled as Multimodal Variational Autoencoder for Fake News Detection (MVAE). This architecture too consists of three sub modules. Later methods for the task include SpotFake [36] and SpotFake+ [35]. SpotFake extracted features from both the text and image modality. It differs from [17, 37] in not using a secondary task within the formulation. Other multimodal features that recently gained a lot of attention to solve the detection problem are those from social media. Various attributes in the form of user meta information, comments, likes *etc.* are being used to form a rich feature database that can better assist the detection task. Recently, Cui et al. [6] used opinions of the user gathered via Twitter to draw insights on information present in news that makes it unreliable. This method not only successfully detects a news article to be reliable or not but also adds an explainability quotient to it.

Knowledge Graph based Approaches. Analysing the facts in a news proves as an important factor in deciding an information to be true or not. Recent research (Pan et al. [26]) shows the use of knowledge graph in improving the fact checking quality in a news article. Such graphs are capable of modelling semantic information to assess the common sense reasoning present in a news article through statistical methods. For example, TransR model [21] can generate embeddings for entity relation triplets. To learn via KGE, the entities has to be projected from entity space to corresponding relation space. This helps in building translations between projected entities, which could yield insights that aid determining veracity.

Social Context Based Approaches. These methods use profile information, network structure and user's past history to determine news credibility. Castillo et al. [4] used four types of features, namely message-based features, user based features, topic-based features and propagation-based features to classify a tweet to be credible or not by a supervised classifier. Jin et al. [16] proposed a method to verify the news by analyzing the conflicting viewpoints of users about the news. They followed a three step process; first, a topic model is used to detect conflicting views about a news; second, a network is created whose vertices are tweets and edges indicate if tweets support or oppose each other; finally, an iterative method is used to deduce the credibility of a news and a process continues

until it converges. Other notable studies following this direction include [38], [30] and [20].

Unsupervised Approaches. There have been very limited attempts to detect fake news in an unsupervised manner. One of the prime reasons behind designing an unsupervised method is that fake news varies across the domains – political fake news may be different than the health-related fake news. Therefore, a model that is solely dependent on the content of the news (analyzes only the text) may not generalize well across all the domains. In this direction, Yin et al. [42] made an attempt to discover the true story from multiple sources of the web. They proposed TruthFinder, which considers the relationship between websites and the content present on these website to determine the truthfulness of a story. An iterative method uses the truthfulness of a link and the trustworthiness of a website to detect the conflicting information present in multiple websites about a story. Recently, Yang et al. [41] proposed UFD, a generative framework based on Bayesian principles by leveraging the opinion of users on the news and their engagements with the social network. Their method relies on the hypothesized increased credibility of the views of verified users. They establish that the extent of engagement (liking, sharing, commenting) of a user varies over time as the news propagates through the network. Apart from the above that address the general fake news problem, there has been work addressing related problems such as identifying political propagandists - specifically, those who seek to align public opinion on contemporary events with the official 'vision' - within an unsupervised setting [25].

Differences from Existing Unsupervised Approaches. Our proposed method, GTUT, differs from the two unsupervised fake news detection approaches [41, 42] discussed above in both the kind of information used as well as the approach taken to address the task. We do not limit our approach to those websites where multiple opinions exist on the web (unlike [42]) nor do we limit the ambit to those that are shared by a significant number of verified users [41]. We use a more general high-level heuristic that relies on assumptions about synchronous user behavior, which we believe leads to a more generalizable method. However, we consider both the methods as our baselines for the comparison.

3 PROBLEM DEFINITION

Consider a dataset of news articles, $\mathcal{A} = \{\ldots, A, \ldots\}$, whose veracity needs to be ascertained by labelling them as either *fake* or *legitimate*. Let $\mathcal{U} = \{\ldots, U, \ldots\}$ be a set of users in a social network, the user dynamics of which we will use to perform unsupervised fake news detection. The footprint of the news articles within the social media network manifests as a set of posts, $\mathcal{P} = \{\ldots, P, \ldots\}$. Each post $P_i = [U_i, A_i, t_i, c_i]$ is created by a user U_i comprising textual content denoted by c_i mentioning news article A_i at time t_i; we are only interested in posts that mention news articles, motivated by the nature of our task. Given that not all social networks expose the underlying social network (directly or indirectly), we make no assumptions about the knowledge of connections between users. The target of the unsupervised fake news detection method is to make use of $\{\mathcal{A}, \mathcal{U}, \mathcal{P}\}$ in order to estimate a label from $\{F, T\}$ to each article in $\mathcal{A}$, denoted as $\mathcal{L}(A)$ (for $A \in \mathcal{A}$), where F denotes *fake* and T denotes *legitimate/truthful*. It may be noted that $\mathcal{U}$ and $\mathcal{P}$ are usually part of the public digital footprint that microblogging services expose, variously referred to as activity log or traces. Following the links within the posts in $\mathcal{P}$ would lead us to $\mathcal{A}$. Thus, all of $\{\mathcal{A}, \mathcal{U}, \mathcal{P}\}$ are easily available in typical scenarios.

3.1 Evaluation

The effectiveness of the unsupervised learning method would be determined on the basis of the correctness of the assigned labels gauged against a set of manually labelled articles. As may be obvious, the manual labellings would be used only for the purposes of evaluation and would be unavailable to the learning technique itself. We will use classical metrics such as precision, recall and F-measure, as will be detailed in the empirical evaluation section.

4 GTUT: OUR METHOD

We now describe our method for unsupervised fake news detection, codenamed **GTUT**, to indicate the usage of **G**raph mining methods over **T**extual, **U**ser and **T**emporal data from social network traces across $\{\mathcal{A}, \mathcal{U}, \mathcal{P}\}$. GTUT is a three phase method that starts by identifying seed fake and legitimate articles in the first phase, and progressively expands the labelling in the second and third phases to cover all articles in $\mathcal{A}$.

4.1 Phase 1: Label Seeding using Bi-cliques

This phase is primarily designed to exploit the high-level observation that *fake news is often spread within a social network in a synchronized manner* (relative to legitimate news) to ensure wide attention and reach in order to further the monetary, electoral or market incentives that motivate their creation. While this observation is particularly motivated by observing political perception campaigns on Twitter[3] which often use slightly different wordings, this is also facilitated by observations that users' expression of sociopolitical leanings yield to prompt propagation of fake news (BBC report [5]). However, given that fake news could propagate through other mechanisms and need not be fully identifiable through synchronized sharing, we only use it as a seeding heuristic in order to identify a seed set of fake and legitimate articles. Our seed set identification relies on identifying synchronous posting behavior involving **same articles** at **similar times** through **textually similar posts**.

We start by modelling a bi-partite graph $\mathcal{G}$ with nodes as articles (i.e., $\mathcal{A}$) and users ($\mathcal{U}$), with an edge induced between a particular user and article *iff* the user has authored a social media post mentioning the article. The edge, E_{AU}, is labelled with the set of posts involving the user-article pair, given that a user could have authored multiple posts involving the same article.

$$Label(E_{AU}) = \{P_i | P_i \in \mathcal{P} \wedge U_i = U \wedge A_i = A\} \quad (1)$$

where $P_i = [U_i, A_i, t_i, c_i]$ as outlined in Section 3. In the interest of identifying similar posting behavior across users involving the **same articles**, we first identify all maximal bi-cliques [44], denoted as $\mathcal{B}$, in the user-article graph. A bi-clique $B \in \mathcal{B}$ is formed by a set of users $B_U \subseteq \mathcal{U}$ *and* set of articles $B_A \subseteq \mathcal{A}$ such that there

[3] https://www.indiatoday.in/india/story/bjp-social-media-cell-trending-document-hack-inner-workings-1454919-2019-02-13

exists an edge between *each* user in B_U and *each* article in B_A. A maximal bi-clique is one which cannot be extended by adding users or articles (to the respective sets that form the bi-clique) while still retaining the property of being a bi-clique; maximal bi-clique finding disallows both bi-cliques and bi-cliques formed by their proper subsets being present in the result together, reducing redundancy and enhancing meaningfulness of result sets. Given that a bi-clique is expected to be *fully connected*, each bi-clique identifies a set of users B_U who have all shared the same set of articles B_A. The definition of bi-cliques becomes vague when it approaches low values of $|B_U|$ and $|B_A|$; for example, each edge in the bi-partite graph is a bi-clique with $|B_U| = |B_A| = 1$. Thus, we would like to impose a minimum threshold on $|B_U|$ as well as on $|B_A|$ so we consider only bi-cliques with good connectivity of users and articles. As a thresholding strategy, we choose a threshold of 5 for the number of users, and choose a threshold on the number of articles such that the bi-cliques collectively cover around 40% of articles in the dataset. More details are in the experimental evaluation section.

Having identified users who tweet the same articles by way of bi-cliques, we would like to score the bi-cliques based on whether the component users tweet the articles at **similar times** using **textually similar posts**. First, we consider each bi-clique separately, and score each article within the bi-clique on the *temporal coherence* of social media footprint within the bi-clique. The temporal coherence of an article $A \in B_A$ is estimated as:

$$Temporal(A \in B_A, B) = max\left\{1 - \frac{BAS(A,B)}{T_{max}}, 0\right\} \tag{2}$$

where $BAS(A,B)$ denotes what we call as the *bursty attention span* of article A among users within bi-clique B. The *span of attention* that an article A had gathered among users in B may simply be estimated as the time difference between the latest post involving A and the earliest post involving A (both obviously computed among posts within the bi-clique). However, such an attention span is sensitive to outlier behavior when a single user in U_B posts about A many days after the burst of attention has died down; we have observed cases of stray posts after a year, making this attention span definition very fragile. Thus, we define the *bursty attention span* as the smallest timespan that encompasses at least 80% of the posts posted about A among users in B. The bursty attention span may be greedily computed by simply excluding posts from either end of the temporal window based on what shortens the remaining span maximally, until only 80% of posts remain. The article-specific temporal coherence, $Temporal(A \in B_A, B)$ of article A within bi-clique B is then computed as being inversely related to the bursty attention span, flattened off beyond a threshold of T_{max}. The temporal coherence of the bi-clique is then estimated as the average of the temporal coherence of articles within it:

$$Temporal(B) = \frac{\sum_{A \in B_A} Temporal(A,B)}{|B_A|} \tag{3}$$

We now turn our attention to similarly estimating the *textual similarity* of posts involving each article $A \in B_A$ within the context of bi-clique B. The article level textual similarity is estimated as:

$$Textual(A \in B_A, B) = \frac{\sum_{(P_x,P_y) \in Posts(B,A)} sim(rep(c_x), rep(c_y))}{(|Posts(B,A)|) \times (|Posts(B,A)| - 1)} \tag{4}$$

In other words, the above computes the average of pairwise similarities between the textual contents of posts mentioning A posted by users in B (denoted as $Posts(B,A)$). We use average of pre-trained word2vec representations for $rep(.)$ and use cosine as the similarity function $sim(.,.)$. Analogous to the temporal case, the textual similarities are aggregated to the bi-clique level:

$$Textual(B) = \frac{\sum_{A \in B_A} Textual(A,B)}{|B_A|} \tag{5}$$

We now score bi-cliques based on how well they fare across the textual and temporal coherence measures using an aggregate scoring modelled as a weighted sum.

$$TTScore(B) = \lambda \times Temporal(B) + (1 - \lambda) \times Textual(B) \tag{6}$$

We consistently set $\lambda = 0.5$ to give equal weighting for both the scorings. The above offers a bi-clique level scoring; however, given that our task is to label articles, we would like to transfer these scores to the articles. Observe that an article could be part of more than one bi-clique. We estimate the score of an article as the average of scores of the bi-cliques it is part of:

$$TTScore(A \in \mathcal{A}) = \frac{\sum_{B \in BiCliques(A,\mathcal{B})} TTScore(B)}{|BiCliques(A,\mathcal{B})|} \tag{7}$$

where $BiCliques(A,\mathcal{B})$ is the set of all bi-cliques in $\mathcal{B}$ that contain article A. Based on our starting assumptions, articles scoring high on $TTScore(.)$ would likely be fake and vice versa. We estimate our seed set of fake articles F_{seed} as the τ articles that scored highest on $TTScore$ and the seed set of legitimate articles T_{seed} as the τ articles that scored lowest on $TTScore$. While τ can be treated as a tunable hyper-parameter, we consistently set τ to be 5% of articles in $\mathcal{A}$.

4.2 Phase 2: Label Spreading within Bi-cliques

Having identified seed fake and legitimate articles using synchronous behavior identification which is expected to be more prevalent among fake, we now consider spreading the labels more broadly within $\mathcal{A}$ using heuristics that are broadly applicable across fake and legitimate classes. In this phase, our target is to label every article across bi-cliques in $\mathcal{B}$. Our heuristic is that articles are likely to be of the same veracity label (i.e., fake or legitimate) if they are (i) *part of the same bi-cliques*, (ii) *have a lot of common users among those who shared them*, and (iii) *are textually similar*.

Towards this, we model a graph with articles across bi-cliques in $\mathcal{B}$ as vertices and edges between pairs of articles being weighted by a score that is computed as follows:

$$Weight(E(A,A')) = \alpha \times \frac{|Bicliques(A) \cap Bicliques(A')|}{|Bicliques(A) \cup Bicliques(A')|} + \beta \times \frac{|Users(A) \cap Users(A')|}{|Users(A) \cup Users(A')|} + (1 - \alpha - \beta) \times Sim(A,A') \tag{8}$$

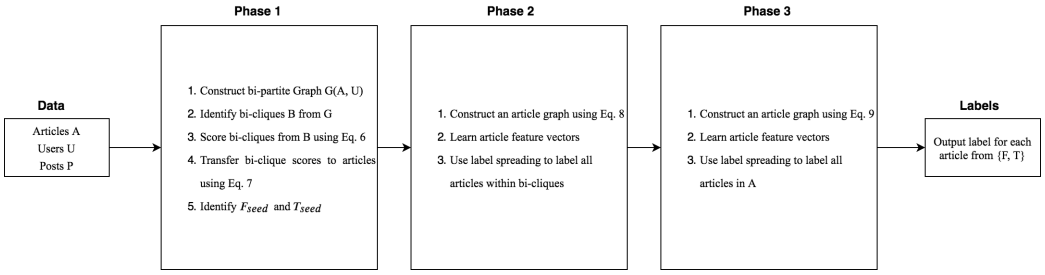

Figure 1: Block diagram of GTUT.

Thus, the edge weight between articles A and A' is modelled as a weighted sum of the number of bi-cliques in which A and A' co-occur ($Bicliques(A)$ denoting the set of bi-cliques containing A), the Jaccard similarity between the sets of users who have shared the articles ($Users(A)$ denoting the users who have shared article A on social media posts), and the textual similarity between the contents of the respective articles themselves. Each of the weights are in $[0, 1]$ with all of them adding up to 1.0. The weights α and β are hyper-parameters, and could be set based on the weighting required for each of these three factors. We use cosine similarity over distributed representations such as *doc2vec* as an estimate of textual similarity. It may be noted that some of the articles in this graph would already have labels due to being part of F_{seed} and T_{seed} from the previous phase.

Having modelled this graph of articles with edge weights set appropriately, we use node2vec [13] to learn feature vectors for each of the articles. Given the existence of labels for a subset of the nodes in the graph, we make use of the label spreading method from [45] over the node2vec feature vectors in order to get all vertices (i.e., all articles across bi-cliques in $\mathcal{B}$) labelled as either *fake* or *legitimate*.

4.3 Phase 3: Full Dataset Labelling

Having labelled all articles from across bi-cliques in $\mathcal{B}$, we now turn our attention to labelling articles outside bi-cliques. Much like in Phase 2, we use a graph-modelling approach, followed by feature vector learning and label spreading.

The weights between nodes in the graph follows the construction of Eq. 8, but omits the first term since the articles yet to be labelled are not part of any bi-cliques. Thus, the weights are set as follows:

$$Weight(E(A, A')) = \gamma \times \frac{|Users(A) \cap Users(A')|}{|Users(A) \cup Users(A')|} + (1-\gamma) \times Sim(A, A') \tag{9}$$

Here, the hyper-parameter γ controls the relative weighting between the user and textual similarities. As in Phase 2, we model the feature vectors using node2vec, and spread the labels to articles outside the bi-cliques using label spreading. This completes the labelling of all articles in $\mathcal{A}$.

4.4 Summary

Across the three phases, GTUT uses various criteria and methods to progressively label all articles in $\mathcal{A}$ as shown in Figure 1. A tabular overview of the labelling progression and methods used appear in Table 1, and a procedural overview appears in Algorithm 1. As an unsupervised method that does not have the luxury of labelled information to characterize, we employ the heuristic of *higher prevalence of synchronous sharing among fake news* to identify a seed set of fake and legitimate articles in Phase 1. Phase 2 and Phase 3 employ heuristics on bi-clique, user and text similarity that are broadly applicable across fake and legitimate classes of articles to spread the labels across all articles.

Note on Scalability: While the bi-clique enumeration step, as may be obvious, would be the most computationally expensive step in GTUT, recent research in enumerating cliques have yielded massive scalability improvements. In particular, a recent method [8] reports clique enumeration methods that can complete in a matter of seconds over million-sized graphs. Our implementation, based on [44], yielded response times in the order of minutes over benchmark datasets we use in our empirical evaluation. Further, our method is not designed for usage in a real-time environment, and is thus unlike real-time user-query based retrieval systems for which response time is critical.

Algorithm 1: GTUT Overview

Data: Articles $\mathcal{A}$, Users $\mathcal{U}$ and Posts $\mathcal{P}$
Result: Label for each article from $\{F, T\}$
Phase 1;
Construct bi-partite graph $\mathcal{G}$ comprising articles and users;
Identify bi-cliques $\mathcal{B}$ from $\mathcal{G}$;
Score bi-cliques within $\mathcal{B}$ using Eq. 6;
Transfer bi-clique scores to articles using Eq. 7;
Identify seed set of fake articles, F_{seed}, and legitimate articles, T_{seed}, as τ% of articles at the high and low end of the score spectrum respectively;
Phase 2;
Construct an article graph using edges weighted as in Eq. 8;
Learn article feature vectors using node2vec;
Use label spreading to label all articles within bi-cliques;
Phase 3;
Construct an article graph using edges weighted as in Eq. 9;
Learn article feature vectors using node2vec;
Use label spreading to label all articles in $\mathcal{A}$;
Output the article labels;

	Phase 1	Phase 2	Phase 3
Articles Labelled	Seed set of articles within bi-cliques	All remaining articles within bi-cliques	All articles outside bi-cliques
Method Used	Bi-clique Mining, Synchronous Sharing	Bi-clique, User and Text Similarity	User and Text Similarity

Table 1: GTUT phases summary.

5 EXPERIMENTAL EVALUATION

We now describe the experimental evaluation of our method vis-a-vis baseline methods. We first start with detailing the datasets, experimental setup and the baselines used in the comparative analysis. This is followed by our empirical results over the datasets and a deeper analysis of our method over various parameter settings.

5.1 Datasets and Experimental Setup

5.1.1 Datasets: We use the code from the *FakeNewsNet* repository [32] in order to collect the two benchmark datasets, *PolitiFact* and *GossipCop*. In *PolitiFact*, a dataset collected from the web portal with the same name[4], journalists and domain experts review political news and provide fact-checking evaluation results to identify news articles as fake or real. *GossipCop*[5], on the other hand, is a website for factchecking entertainment stories collected from various media outlets. They provide a rating score, ranging from 0 to 10, to classify an article from fake to real. These scores have been discretized into binary fake/real labels, as outlined in [32]. While these datasets are datasets of news stories, the social context behind these stories are captured using the Twitter Advanced API, as outlined in the same paper. Specifically, tweets that share the news story are collected, along with the userids of users who authored such tweet and the time of posting the tweet. The statistics for the datasets are shown in Tables 2 and 3 respectively. Articles are labelled as either fake or legitimate/real; each post is labelled with the label of the article which it posts about. Given the length limitation of tweets, we did not come across posts that mention multiple articles in the same tweet. For purposes of collecting statistics that are shown in the tables, we consider each user who has tweeted at least one fake article as being a *fake (tweeting) user* and each user who has tweeted at least one legitimate/real article as a *real (tweeting) user*. Evidently, there could be users who have tweeted both real and fake articles, making these user sets overlapping. The extent of the overlap is also recorded in the tables. As may be observed from the description and the statistics of the datasets, the datasets are from disparate domains and also have very different distributions across the real and fake classes. While non-fake tweets and users dominate the PolitiFact dataset, the vice versa is true of the GossipCop dataset. The empirical evaluation over such widely varying benchmark datasets, we believe, will inspire confidence in the generalizability of our empirical study.

5.1.2 Baselines: As outlined in Section 2, there has been very limited work in the area of unsupervised fake news detection. The recent unsupervised fake news detection technique from [41], called *UFD*, forms our primary baseline in the empirical evaluation. We also use an early and very popular (cited 600+ times) fake news detection method, *TruthFinder* [42], as another baseline method for our empirical study. Our third baseline is the *Majority Voting* (MV) approach outlined in [41].

Type	Truthful/Real	Fake
Articles	369	367
Tweets	498005	355290
Users	283400	85208
Users posting both truthful and fake articles: 16060		

Table 2: Statistics of the PolitiFact dataset.

Type	Truthful/Real	Fake
Articles	600	450
Tweets	41580	123875
Users	6013	53288
Users posting both truthful and fake articles: 2520		

Table 3: Statistics of the GossipCop dataset.

5.1.3 Evaluation Measures: Our unsupervised fake news detection task involves assigning a T/F label to each of the articles in the dataset, by making use of user and temporal patterns of sharing them, as available from Twitter. As a natural way of measuring the accuracy of the labelling output from the techniques, we use *Precision*, *Recall* and *F-Score* for each of the two classes (i.e., $\{T, F\}$) as evaluation measures [12]. Consider a single class, say F; *precision* and *recall* measure the number of fake articles as a fraction of those labelled fake by the method being evaluated and the gold-standard respectively. *F-Score* is the harmonic mean of precision and recall, which achieves high scores when both are high. Given that our methods do not yield scores but crisp labellings, conventional curves such as precision-recall or ROC curves are not applicable for this scenario. Further to these, we also examine the various methods on the overall accuracy, which is the fraction of correctly labelled instances.

5.1.4 Experimental Setup: We use python implementations of GTUT and the baseline methods in our empirical study. We use the recommended hyper-parameter settings for the baseline methods from their respective papers. GTUT has three hyper-parameters, α and β from Phase 2, and γ from Phase 3. We consistently use $\alpha = 0.4$, $\beta = 0.4$ and $\gamma = 0.75$ across our experiments unless otherwise mentioned.

5.2 Comparative Analysis

The results of the comparative analysis pitching GTUT against the three baseline methods is illustrated across Table 4 (PolitiFact Dataset Results) and Table 5 (GossipCop Dataset Results). As may

[4]http://www.politifact.com
[5]http://www.gossipcop.com

Detection Method	True			Fake		
	Precision	*Recall*	*F-Score*	*Precision*	*Recall*	*F-Score*
UFD	0.72	**0.83**	0.77	0.62	0.46	0.53
TruthFinder	0.69	0.62	0.65	0.45	0.53	0.49
MV	0.74	0.81	0.77	0.62	0.52	0.57
GTUT	**0.77**	**0.83**	**0.80**	**0.83**	**0.76**	**0.79**

Table 4: PolitiFact dataset results. The best result for each evaluation measure (i.e., each column) is shown in bold.

Detection Method	True			Fake		
	Precision	*Recall*	*F-Score*	*Precision*	*Recall*	*F-Score*
UFD	0.58	0.85	0.69	0.81	0.51	0.62
TruthFinder	**0.75**	0.65	0.69	0.60	**0.71**	0.65
MV	0.58	0.75	0.65	0.45	0.27	0.34
GTUT	0.72	**0.93**	**0.81**	**0.87**	0.56	**0.68**

Table 5: GossipCop dataset results. The best result for each evaluation measure (i.e., each column) is shown in bold.

Method	PolitiFact	GossipCop
UDF	0.70	0.66
TruthFinder	0.59	0.67
MV	0.65	0.55
GTUT	**0.80**	**0.77**

Table 6: Accuracy analysis.

be seen therein, GTUT is seen to outperform the baseline methods comfortably across both the datasets, except in a couple of cases in GossipCop where it trails behind TruthFinder. While these establish the effectiveness of the fake news modelling within GTUT vis-a-vis the baseline methods, some trends are noteworthy. First, it maybe recollected that the GTUT Phase 1 chooses the same number of true and fake articles to seed the respective classes; this was set to 5% of the dataset size. This reflects an expectation that there is a reasonable parity in the number of true and fake articles in the dataset. While this is true for the PolitiFact dataset, the GossipCop dataset contains a higher number of true articles than fake ones. This imbalance makes the seeding inaccurate reflecting in a larger deviation between the true and fake F-scores for GTUT over the GossipCop dataset. This could be corrected for by changing the seeding parity across true and fake classes, if high-level statistics of true and fake class cardinalities are available. Second, the fact that GTUT outperforms all methods on the F-Score in each of the four combinations indicates that GTUT is able to choose a reasonable trade-off between precision and recall across widely varying datasets, underlining the efficacy of the GTUT technique.

Table 6 plots the accuracy values for the various methods. Accuracy computes the number of correctly labelled instances as a fraction of the dataset, and thus provides a convenient single metric that captures the correctness of the labellings across the T/F classes together. The accuracy analysis indicates that GTUT convincingly surpasses the baseline methods, achieving a gain of 10 percentage points in accuracy on both the datasets. It is also notable that the second method is different across the datasets, indicating that the margin of improvement achieved by GTUT against any single method is much higher. The accuracy analysis thus further underlines the superiority of the GTUT formulation.

5.3 GTUT Analysis

Having analyzed the comparative performance of GTUT against baseline methods, our focus is now on analyzing the GTUT framework at some depth.

PolitiFact Dataset	
#minart	Dataset Covered
3	48%
4	40%
5	31%
6	23%
GossipCop Dataset	
#minart	Dataset Covered
7	45%
8	43%
9	41%
10	39%
11	37%

Table 7: GTUT Phase 1 Bi-clique selection.

5.3.1 Article Cardinality Thresholds in Phase 1. As mentioned in Section 4.1, we use a minimum cardinality threshold on both the number of users and the number of articles within the bi-clique identification step, to avoid identifying meaningless small patterns as bi-cliques. Our strategy has been to impose a threshold of a minimum of 5 users, and a threshold on the number of articles in a way that the bi-cliques together cover around 40% of the dataset. The threshold as a function of the covered dataset proportion offers applicability across datasets of widely varying sizes. This strategy yielded a minimum article threshold of 4 and 9 for the PolitiFact and GossipCop datasets respectively. We analyze the covered dataset

proportion around these chosen settings of minimum number of articles, or *#minart* for short, in Table 7. As may be seen therein, the dataset coverage could widely vary, making exact threshold attainment not always feasible. However, we observed that GTUT is not affected much by choosing dataset coverage thresholds anywhere in the 30% − 50% range.

PolitiFact Dataset			
α	β	$1-\alpha-\beta$	Accuracy
0.45	0.35	0.20	0.82
0.35	0.45	0.20	0.82
0.30	0.55	0.15	0.83
0.25	0.65	0.10	0.81
GossipCop Dataset			
α	β	$1-\alpha-\beta$	Accuracy
0.45	0.35	0.20	0.79
0.35	0.45	0.20	0.81
0.30	0.55	0.15	0.80
0.25	0.65	0.10	0.79

Table 8: GTUT Study over phase 2 parameters.

5.3.2 Hyper-parameters from Phase 2. The Phase 2 hyper-parameters are α and β which determine the importance of bi-clique and user similarity terms, with them also implicitly defining the importance of the textual similarity term whose weight is modelled as $(1-\alpha-\beta)$. For convenience of analysis and to reduce clutter, we analyze the varying performance against the accuracy measure which captures the labelling performance on both the true and fake categories. We analyze the performance of a few (α, β) parameter settings around which high overall accuracy was obtained. Table 8 plots the trends, which illustrates a high stability of accuracy around the region studied. Such smooth movements were observed all over the space of values of α and β indicating that GTUT is not highly sensitive to small variations in the values of the parameters.

PolitiFact Dataset		
γ	$1-\gamma$	Accuracy
0.80	0.20	0.77
0.75	0.25	0.80
0.70	0.30	0.78
GossipCop Dataset		
γ	$1-\gamma$	Accuracy
0.80	0.20	0.76
0.75	0.25	0.77
0.70	0.30	0.74

Table 9: GTUT Study over phase 3 parameters.

5.3.3 Hyper-parameters from Phase 3. Phase 3, while being similar in construction to Phase 2, does not have the bi-clique similarity term, and thus, has one fewer hyper-parameter. γ determines the weight associated with the user similarity term, whereas $1-\gamma$ is the weight for the textual similarity term. As seen in Section 5.3.2, the trends on γ also shows high stability over small variations.

6 CONCLUSIONS AND FUTURE WORK

We considered the task of unsupervised detection of fake news articles by making use of social media traces. Unsupervised fake news identification is significantly more challenging than its supervised counterpart due to the absence of labelled data to aid modelling the distinction between fake and legitimate news. We develop a three-phase graph-based approach, code-named GTUT, that uses a graph-based approach for the task. GTUT uses a three phase approach, wherein the first phase uses high-level assumptions of inter-user behavioral dynamics to identify a seed set of fake and real news articles. This is enabled through identification of bi-cliques and scoring them on textual and temporal coherence. The second phase expands the labelling from the seed set to all articles involved in bi-cliques making use of three kinds of similarity information, viz., bi-clique similarity, user similarity and textual similarity. This phase makes use of graph modelling followed by graph embeddings and label spreading. The third phase targets labelling the articles not involved in bi-cliques through graph modelling and label spreading to ensure that all articles in the dataset are assigned as either fake or real. Through an extensive set of experiments over popular real-world datasets used in the fake news detection task, we establish that GTUT significantly outperforms state-of-the-art and popular baselines for the task. In particular, GTUT is seen to achieve gains of over 10 percentage points on accuracy, achieving accuracies close to 80% for unsupervised fake news detection. We also performed an extensive study on the sensitivity of GTUT to its hyperparameters, and illustrated that GTUT is quite stable and insensitive to minor variations in the hyperparameters. Our extensive empirical analysis establishes the effectiveness of GTUT for unsupervised fake news detection.

Future Work. Our task in GTUT was to make use of social media traces, to ensure its applicability for cases where social media user connectivity information is not necessarily available. That said, social media connectivity would likely provide some useful information for the task. We are exploring the possibilities of characterizing network connectivity patterns in fake news propagation through high-level heuristics to inform the fake/real seeding phase in GTUT. Secondly, social media sharing of a news article often offers very vivid digital footprints. For example, a tweet may be re-tweeted, liked or shared, whereas a facebook post can be *'reacted to'* using emotionally rich responses. Given a large number of studies that relate fake news to emotions, we are considering the usage of emotion information within such social media traces in order to improve unsupervised fake news detection. Thirdly, the real-fake spectrum in news is often very broad and binary labellings, as used in fake news detection techniques including GTUT, may be criticized as an oversimplification. We are considering ways in which this binary labelling can be relaxed in order to offer more fine-grained labellings, staying within the graph-based framework. Other directions include generating interpretable results [3] and enriching semantic query expansion (e.g., [19]) with veracity orientation.

Acknowledgements

This work was supported by the MHRD (India) under the SPARC programme project #P620. The authors would like to thank Kaiqiang Yu (PhD candidate at NTU) for enriching discussions.

REFERENCES

[1] Ehsan Ahmadizadeh, Erfan Aghasian, Hossein Pour Taheri, and Roohollah Fallah. 2015. An Automated Model to Detect Fake Profiles and botnets in Online Social Networks Using Steganography Technique. *iosrjournals* 17 (02 2015), 2278–661. https://doi.org/10.9790/0661-17146571

[2] Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of economic perspectives* 31, 2 (2017), 211–36.

[3] Vipin Balachandran, P Deepak, and Deepak Khemani. 2012. Interpretable and reconfigurable clustering of document datasets by deriving word-based rules. *Knowledge and information systems* 32, 3 (2012), 475–503.

[4] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*. 675–684.

[5] Santanu Chakrabarti, Lucile Stengel, and Sapna Solanki. 2018. Duty, Identity, Credibility:'Fake News' and the Ordinary Citizen in India. *BBC World Service Audiences Research* (2018).

[6] Limeng Cui, Kai Shu, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. dEFEND: A System for Explainable Fake News Detection. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (Eds.). ACM, 2961–2964. https://doi.org/10.1145/3357384.3357862

[7] John M Darley and Paget H Gross. 1983. A hypothesis-confirming bias in labeling effects. *Journal of Personality and Social Psychology* 44, 1 (1983), 20.

[8] Apurba Das, Seyed-Vahid Sanei-Mehri, and Srikanta Tirthapura. 2020. Shared-memory Parallel Maximal Clique Enumeration from Static and Dynamic Graphs. *ACM Transactions on Parallel Computing (TOPC)* 7, 1 (2020), 1–28.

[9] Hridoy Sankar Dutta, Aditya Chetan, Brihi Joshi, and Tanmoy Chakraborty. 2018. Retweet us, we will retweet you: Spotting collusive retweeters involved in blackmarket services. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 242–249.

[10] B. Erşahin, Ö. Aktaş, D. Kılınç, and C. Akyol. 2017. Twitter fake account detection. In *2017 International Conference on Computer Science and Engineering (UBMK)*. 388–392. https://doi.org/10.1109/UBMK.2017.8093420

[11] Axel Gelfert. 2018. Fake news: A definition. *Informal Logic* 38, 1 (2018), 84–117.

[12] Cyril Goutte and Eric Gaussier. 2005. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In *European Conference on Information Retrieval*. Springer, 345–359.

[13] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[14] Hassan Halawa, Matei Ripeanu, Konstantin Beznosov, Baris Coskun, and Meizhu Liu. 2017. An Early Warning System for Suspicious Accounts. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security* (Dallas, Texas, USA) *(AISec '17)*. ACM, New York, NY, USA, 51–52. https://doi.org/10.1145/3128572.3140455

[15] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A. Efros. 2018. Fighting Fake News: Image Splice Detection via Learned Self-Consistency. *CoRR* abs/1805.04096 (2018).

[16] Zhiwei Jin, Juan Cao, Yongdong Zhang, and Jiebo Luo. 2016. News verification by exploiting conflicting social viewpoints in microblogs. In *Thirtieth AAAI conference on artificial intelligence*.

[17] Dhruv Khattar, Jaipal Singh Goud, Manish Gupta, and Vasudeva Varma. 2019. MVAE: Multimodal Variational Autoencoder for Fake News Detection. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. ACM, New York, NY, USA, 2915–2921. https://doi.org/10.1145/3308558.3313552

[18] Jooyeon Kim, Behzad Tabibian, Alice Oh, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. 2018. Leveraging the crowd to detect and reduce the spread of fake news and misinformation. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 324–332.

[19] Adit Krishnan, P Deepak, Sayan Ranu, and Sameep Mehta. 2018. Leveraging semantic resources in diversified query expansion. *World Wide Web* 21, 4 (2018), 1041–1067.

[20] Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. 2013. Prominent features of rumor propagation in online social media. In *2013 IEEE 13th International Conference on Data Mining*. IEEE, 1103–1108.

[21] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (Austin, Texas) *(AAAI'15)*. AAAI Press, 2181–2187. http://dl.acm.org/citation.cfm?id=2886521.2886624

[22] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks. (2016).

[23] Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. Association for Computational Linguistics.

[24] F. Marra, D. Gragnaniello, D. Cozzolino, and L. Verdoliva. 2018. Detection of GAN-Generated Fake Images over Social Networks. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. 384–389. https://doi.org/10.1109/MIPR.2018.00084

[25] Michael Orlov and Marina Litvak. 2018. Using behavior and text analysis to detect propagandists and misinformers on twitter. In *Annual International Symposium on Information Management and Big Data*. Springer, 67–74.

[26] Jeff Z. Pan, Siyana Pavlova, Chenxi Li, Ningxi Li, Yangmei Li, and Jinshuo Liu. 2018. Content Based Fake News Detection Using Knowledge Graphs. In *International Semantic Web Conference (1) (Lecture Notes in Computer Science)*, Vol. 11136. Springer, 669–683.

[27] Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2017. A Stylometric Inquiry into Hyperpartisan and Fake News. *CoRR* abs/1702.05638 (2017).

[28] Vahed Qazvinian, Emily Rosengren, Dragomir R Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 1589–1599.

[29] Arne Roets et al. 2017. 'Fake news': Incorrect, but hard to correct. The role of cognitive ability on the impact of false information on social impressions. *Intelligence* 65 (2017), 107–110.

[30] Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 797–806.

[31] Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*. 1–10.

[32] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2018. FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media. *CoRR* abs/1809.01286 (2018). arXiv:1809.01286 http://arxiv.org/abs/1809.01286

[33] Kai Shu, Suhang Wang, and Huan Liu. 2017. Exploiting Tri-Relationship for Fake News Detection. *CoRR* abs/1712.07709 (2017).

[34] Rosa Sicilia, Stella Lo Giudice, Yulong Pei, Mykola Pechenizkiy, and Paolo Soda. 2018. Twitter rumour detection in the health domain. *Expert Systems with Applications* 110 (2018), 33–40.

[35] Shivangi Singhal, Anubha Kabra, Mohit Sharma, Rajiv Shah, Tanmoy Chakraborty, and Ponnurangam Kumaraguru. 2020. SpotFake+: A Multimodal Framework for Fake News Detection via Transfer Learning. In *AAAI* (NewYork).

[36] Shivangi Singhal, Rajiv Shah, Tanmoy Chakraborty, Ponnurangam Kumaraguru, and Shin'ichi Satoh. 2019. SpotFake: A Multimodal Framework for Fake News Detection. In *IEEE BigMM* (Singapore). http://precog.iiitd.edu.in/pubs/SpotFake-IEEE_BigMM.pdf

[37] Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection. In *SIGKDD*. ACM, New York, NY, USA, 849–857. https://doi.org/10.1145/3219819.3219903

[38] Ke Wu, Song Yang, and Kenny Q Zhu. 2015. False rumors detection on sina weibo by propagation structures. In *2015 IEEE 31st international conference on data engineering*. IEEE, 651–662.

[39] Liang Wu and Huan Liu. 2018. Tracing fake-news footprints: Characterizing social media messages by how they propagate. In *Proceedings of the eleventh ACM international conference on Web Search and Data Mining*. 637–645.

[40] Cao Xiao, David Mandell Freeman, and Theodore Hwa. 2015. Detecting Clusters of Fake Accounts in Online Social Networks. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security* (Denver, Colorado, USA) *(AISec '15)*. ACM, New York, NY, USA, 91–101. https://doi.org/10.1145/2808769.2808779

[41] Shuo Yang, Kai Shu, Suhang Wang, Renjie Gu, Fan Wu, and Huan Liu. 2019. Unsupervised fake news detection on social media: A generative approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5644–5651.

[42] Xiaoxin Yin, Jiawei Han, and S Yu Philip. 2008. Truth discovery with multiple conflicting information providers on the web. *IEEE Transactions on Knowledge and Data Engineering* 20, 6 (2008), 796–808.

[43] Yan Zhang, Weiling Chen, Chai Kiat Yeo, Chiew Tong Lau, and Bu Sung Lee. 2017. Detecting rumors on online social networks using multi-layer autoencoder. In *2017 IEEE Technology & Engineering Management Conference (TEMSCON)*. IEEE, 437–441.

[44] Yun Zhang, Charles A Phillips, Gary L Rogers, Erich J Baker, Elissa J Chesler, and Michael A Langston. 2014. On finding bicliques in bipartite graphs: a novel algorithm and its application to the integration of diverse biological data types. *BMC bioinformatics* 15, 1 (2014), 110.

[45] Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. 2004. Learning with local and global consistency. In *Advances in neural information processing systems*. 321–328.

[46] Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)* 51, 2 (2018), 1–36.