Epidemiology Inspired Framework for False Information Mitigation in Social Networks

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Bhavtosh Rath

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Jaideep Srivastava

December, 2020

Acknowledgements

First and foremost, I am grateful to my advisor Jaideep Srivastava for his guidance over the past five years. I could not have asked for a better guide who gave me freedom to pursue new ideas, and at the same time was very accessible when I required direction and counselling. Without his persistent help this dissertation would not have been possible. I would also like to thank Atanu Roy (whose preliminary work on computational trust was the stepping stone of my research journey), Wei Gao (my first collaborator who taught me the basics of presenting research) and Jisu Huh (with whom I actively collaborated on interdisciplinary projects at the university). Last but not the least, I am grateful to all members of our research lab who created a conducive environment to do research.

I submit my heartiest gratitude to professors Jaideep Srivastava, Abhishek Chandra, Jisu Huh and Joseph Konstan for serving on my thesis and proposal examination committee. They are research giants of their respective fields, and their feedback during the proposal examination immensely helped shape my final dissertation.

My PhD journey would have been incomplete without the love and support of my parents (Anjali Rath, Jibitesh Rath), sister (Manaswenee Rath Sharma), my brother-in-law (Deepak Sharma), niece (Ariyana) and my New York family (Daadi, Parbodh Sharma, Savita Sharma, Aakash Aggarwal, Jyoti Sharma Aggarwal and nephew Armaan). I am thankful to them from the bottom of my heart for their continuous and unparalleled love. Also a special thank you to my friends - Vaibhav Sharma, Sampreeti Jena, Biswaranjan Mohanty, Adway De, Vy Le, Kerry Wang, Maria Mendez Enriquez and her family, members of the Odia Minnesota community, and all extended family members part of the watsapp group - *family fun:)* for keeping my sanity levels in check, especially over the past two years. There are other friends and family members who I do not mention here, but my journey would have been incomplete without them.

Lastly, I would like to thank the administrative and tech support staff of the department of College of Science & Engineering, and also the staff of the International Student and Scholar Services at the University for promptly answering my queries over the years.

Dedication

To my mother *Anjali Rath* and my maternal grandmother *Belin Mishra* . . .
For their selfless love and affection over the years.

Abstract

Social networking platforms like Facebook and Twitter are used by millions of people around the world to not only share information but also personal opinions about it. Often these information and opinions are unverified, which has caused the problem of spreading of false information, popularly termed *Fake News*. As social media platforms generate huge volumes of data, computational models for the detection and prevention of false information spreading has gained a lot of attention over the last decade, with most proposed models trying to identify the veracity of the information. Techniques involve extracting features from the information's propagation path in social networks or from the information content itself. In this thesis we propose a complementary approach to false information mitigation inspired from the domain of Epidemiology.

Epidemiology is the field of medicine which deals with the incidence, distribution and control of disease among populations. This dissertation proposes an epidemiology inspired framework where false information is analogous to disease, social network is analogous to population and how likely are people to believe an information endorser is analogous to their vulnerability to disease. In this context we propose four phases that fall in the domain of social network analysis. The first phase is called *Vulnerability assessment*, where we estimate how likely are nodes and communities to believing false information before an information starts spreading. This is equivalent to assessing the vulnerability (i.e. immunity) of people before infection spreading begins. The second phase is called *Identification of infected population*, where given the complete spreading paths of information, we identify the false information spreaders from true information spreaders. This is equivalent to identifying infected population after the infection spreading is complete. The third phase is called *Risk assessment of population*, where given the partial spreading paths of false information, we predict nodes that are most likely to be infected in future. This is equivalent to contact tracing, where we want to identify the exposed population that needs to be quarantined to prevent spreading of the infection. The final phase is called *Infection control and prevention* where we identify people as false information spreaders, refutation information spreaders or non-spreaders in co-existing false and refutation information networks. This can aid in strategies to target people with refutation information to a) change the role of a false information spreader into a true information spreader (i.e. using refutation information as an antidote) and b) prevent a person from becoming a false information spreader (i.e. using refutation information as a vaccine).

Through experiments on real world information spreading networks on Twitter, we showed the effectiveness of our proposed models and confirm our hypothesis that spreading of false information is more sensitive to behavioral properties like trust and credibility than spreading of true information.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 False Information Mitigation in Social Media

Social media is an integral part of our lives. As of 2019, the average time spent on social media platforms was projected to be 144 minutes per day[1]. A major portion of this time is spent to interact with others and in the process consume and share information and opinions. With major social media platforms like Facebook and Twitter becoming the 'go-to place' for news consumption, where practically all the information is user-generated and whose veracity is not guaranteed, society is facing the serious and well-nigh existential threat of spreading of false information, popularly called *Fake News*. The spreading of fake news can happen both intentionally (i.e. disinformation) and unintentionally (i.e. misinformation). While the former is more dangerous, often being the result of a well-coordinated campaign, both have negative consequences for the society.

The wide adoption of social media platforms has resulted in the creation of social big data, and its accessibility has motivating researchers to propose various computational models for combating false information. So far the focus of most research has been on determining whether an information is false or not using features extracted manually (through feature engineering) or automatically (through machine learning/ deep learning) from data. In this dissertation we explore a complementary approach to false

[1] https://www.broadbandsearch.net/blog/average-daily-time-on-social-media

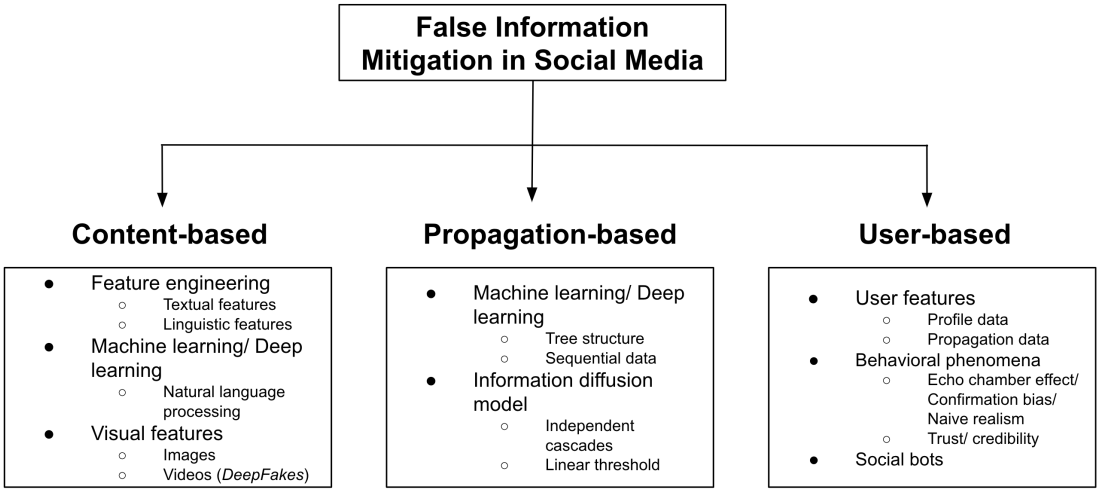


Figure 1.1: Summary of false information mitigation research in social media.

information mitigation. We propose models as part of a novel false information prevention and control framework inspired from the domain of epidemiology to determine whether a person will endorse a false information or not. Our framework incorporates people's behavioral data along with their underlying network structure. Like in epidemiology, models proposed within the framework cover the entire life cycle of infection spreading: i.e. before the false information originates, after the false information starts spreading and containment of its further spreading.

1.2 Related Work

Literature of research in false information detection and prevention strategies in social media is vast, and can be divided broadly into three categories as shown in Figure 1.1. They are *Content-based*, *Propagation-based* and *User-based* models.

1.2.1 Content-based models

Majority of research in false information mitigation can be categorized as content-based models. Proposed approaches mostly rely on textual or visual based data. Earlier work relied mostly on hand engineering relevant data that exploited linguistic features.

Some relevant features used were grammatical structure of a sentence [1] and parts-of-speech [2]. Textual features that other models used included topic distribution and sentiment information [3], language complexity and stylistic features [4], pronoun and exclamation marks [5], content embedding [6], Language complexity, readability, moral foundations and psycholinguistic cues [7], tweet and user topics [8] and topic distribution [9] to mention a few. A major limitation of such hand engineered features is that they do not fully exploit the rich semantic and syntactic information in the content.

More recently deep learning based models have gained popularity as they can automatically extract both simple features and more complex features for text classification. Models implementing recurrent neural networks [10, 11] convolutional neural network [12], convolutional and recurrent neural networks combined [13] have been proposed. More recently other sophisticated deep learning based models have gained popularity. Zhang et al. [14] proposed a graph neural network based multimodal model that aggregates textual information news articles, creators and subjects to identify news veracity. Ma et al. [15] applied generative adversarial networks to counter rumor dissemination by generating confusing training examples to challenge the discriminator of its detection capacity. Shu et al. [16] applied attention mechanism that captures both news contents and user comments to propose an explainable fake news detection system. Khattar et al. [17] used textual and visual information in a variational autoencoder model coupled with a binary classifier for the task of fake news detection.

Apart from content-based models using features extracted from text, models have also been proposed that extract features from visual content such as images and videos. Jin et al. [18] proposed hand-crafted features from news images to study fake news detection. More recently the problem of *Deepfakes* has been addressed using generative adversarial based models [19].

1.2.2 Propagation-based models

Information propagates in social networks through the action of sharing/retweeting that results in a diffusion cascade or tree structure with the source poster at the root. The propagation dynamics of information contents have also been utilized to propose false information detection models. Ma et al. [20] proposed a propagation tree kernel methods to compare the similarity between information propagation trees. Wu et al. [21] similarly defined a random walk graph kernel to compute similarity between different propagation trees. Ma et al. [22] also proposed an improved model using recursive neural networks that used syntactic and semantic parsing to extract features from information cascades. Some models inspired from information diffusion concepts such as Linear Threshold and Independent Cascades have also been proposed [23].

Ruths [24] proposed a context–based detection methods mainly leverages features extracted from the process of information propagation. Recurrent neural networks [25, 12] are used to model temporal data, where propagation data is modeled as sequential data and both temporal and content features are integrated. Lu and Li [26] integrated attention mechanism with graph neural networks using text information and propagation structure to identify whether the source information is fake or not.

Many other propagation based models also incorporate textual features and thus can be categorized as content-based models.

1.2.3 User-based models

User-based models are based on features extracted from two types of data. First kind of features are extracted from content of user profiles. Castillo et al. [27] analyzed hand-crafted user features to study user credibility. Wu and Liu [28] constructed user representations using network embedding approaches on the social network using profile data. The second kind of features is obtained from behavioral data of sharing and responding. Tacchini et al. [29] build a classification model based on who liked the news. Qazvinian et al. [30] collected user engagements to model user behavior patterns that could help identify false news. The role of *bots* (nefarious actors that aim to artificially engineer the virality of information) in false information spreading has also been studied. Their use in political campaigns has been studied in detail [31, 32] which has led to building of bot detection tools [33]. The role of *echo-chamber effect* (polarization of a person's viewpoints leading to less diverse exposure and discussion between unaligned users) in spreading of false has been studied [34, 35]. Two other major behavioral characteristics namely *confirmation bias* (the tendency of people to interpret evidence that confirms their pre-existing notions) and *naive realism* (people's belief that their perception of reality is true) [36] have been found to make people vulnerable to believing false information.

1.2.4 Empirical research of false and true news

Apart from computational research for false information mitigation, researchers have also empirically studied the characteristics that differentiate fake news from true news. Horne et al. [4] identified characteristics that differentiate fake news contents from true news contents by studying the textual characteristics of articles. They showed that titles of fake news articles are longer, had more capitalized words, used fewer stop words, were repetitive and had fewer nouns. Perez et al. [37] found that fake news articles contained more social words, verbs and temporal words, suggesting that false information focused more on on the present and future. Newman et al. [38] showed that deceptive stories had lower cognitive complexity, fewer exclusive words, more negative emotion words, and more action words. Qian et al. [39] analyzed sentiments of the responses to false and true news and found that false news received more negative and questioning responses than true news. Vosoughi et al. [40] showed that false news reached more people than the truth and that falsehood diffused faster and deeper than the truth.

Summary

An exhaustive list of work on false information mitigation research in social media can be found in the following survey papers: [41, 42, 36, 43, 44]. A major limitation with most existing computational models is that they rely on the presence of false information to generate meaningful features, thus making it difficult to model false information mitigation strategies. Also, not many computational models have been proposed exploring psychological concepts from historical behavioral data that make people vulnerable to spreading false information. Our framework proposes content-agnostic models using two important components that do not rely on the presence of false information: people's historical behavioral data and underlying network structure. Contributions made in this thesis towards false information mitigation research can be visually summarized in Figure 1.2.

1.3 Computational Trust

Trust is an important part of any social interaction, and in the context of social media, researchers have been using social networks widely to understand how trust manifests

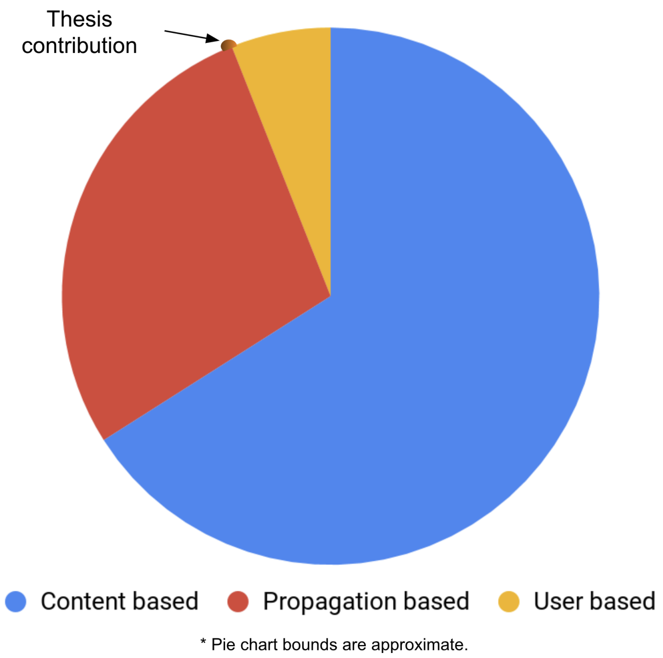


Figure 1.2: Thesis contribution in false information mitigation research.

among users. However, such an abstract concept of trust is generally very hard to compute. In general, trust in a social network is defined as a set of scores assigned to each actor in the network, representing his/her level of trust. Specifically, the level of trust can be manifested by assigning a pair of trust scores to each actor which are termed as trustingness and trustworthiness scores [45]. The former is defined as the propensity of an actor to trust his neighbors in the network, while the latter is defined as the willingness of the network as a whole to trust an individual actor.

For quantifying trust, we require proxies of trust that can map the social interactions to the original concepts of trust. In the context of network on Twitter, there can be various levels of user interactions acting as the proxies, such as following, retweeting, liking, replying, etc. For example, a user whose tweets are more likely to be retweeted by others is expected to have a high trustworthiness score, while a user who is more likely to retweet others' tweets is expected to have a high trustingness score. Without the loss of generality, in our work, we adopt following and retweeting as the proxy of trust, and our proposed model is generic and can be straightforwardly extended to accommodate any other kind of proxies. Figure 1.3 illustrates the trust relationships among the users in a retweet network, where the direction of edges indicates that trust is given to tweeters from retweeters (i.e. retweeter follows tweeter) and the number of times of retweeting between two users can be used as edge weights.

1.3.1 Background

Various researchers have tried to assign trust scores [46, 47, 48] to nodes in a network to accomplish various tasks. Trust scores can be defined as scores that an algorithm puts on a node in a trust network based on various structural aspects of the node. Another algorithm Eigentrust [47] proposes to rate trust scores of peers in a P2P network. These scores help an ordinary user in the network to identify the trustworthy peers and initiate content download from them. Eigentrust, like Pagerank [49] calculates a single score for each node in the network. However, in this algorithm, one's reputation does not play a part in the weight of the node's trust vote. Other researchers have proposed measures to rank bias and deserve of a node in a network [48]. They used an iterative matrix algorithm to calculate bias and deserve of nodes which reinforce each other.

1.3.2 Trustingness and Trustworthiness

To calculate trustingness and trustworthiness scores, the TSM algorithm [45] takes a directed graph as input together with a specified convergence criteria or a maximum permitted number of iterations. In each iteration for every node in the network, trustingness and trustworthiness are computed using the equations below:

$$ti(v) = \sum_{\forall x \in out(v)} \left(\frac{w(v,x)}{1 + (tw(x))^s} \right) \tag{1.1}$$

$$tw(u) = \sum_{\forall x \in in(u)} \left(\frac{w(x,u)}{1 + (ti(x))^s} \right) \tag{1.2}$$

where u and v are user nodes, $ti(v)$ and $tw(u)$ are trustingness and trustworthiness scores of v and u, respectively, $w(v,x)$ is the weight of edge from v to x, $out(v)$ is the set of outgoing edges of v, $in(u)$ is the set of incoming edges of u, and s is the involvement score of the network. Involvement is basically the potential risk an actor takes when creating a link in the network, which is set to a constant empirically in [45].

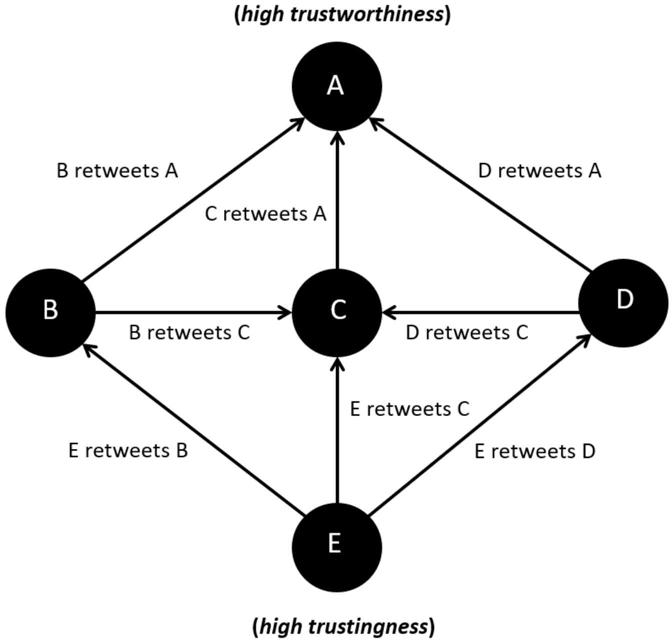
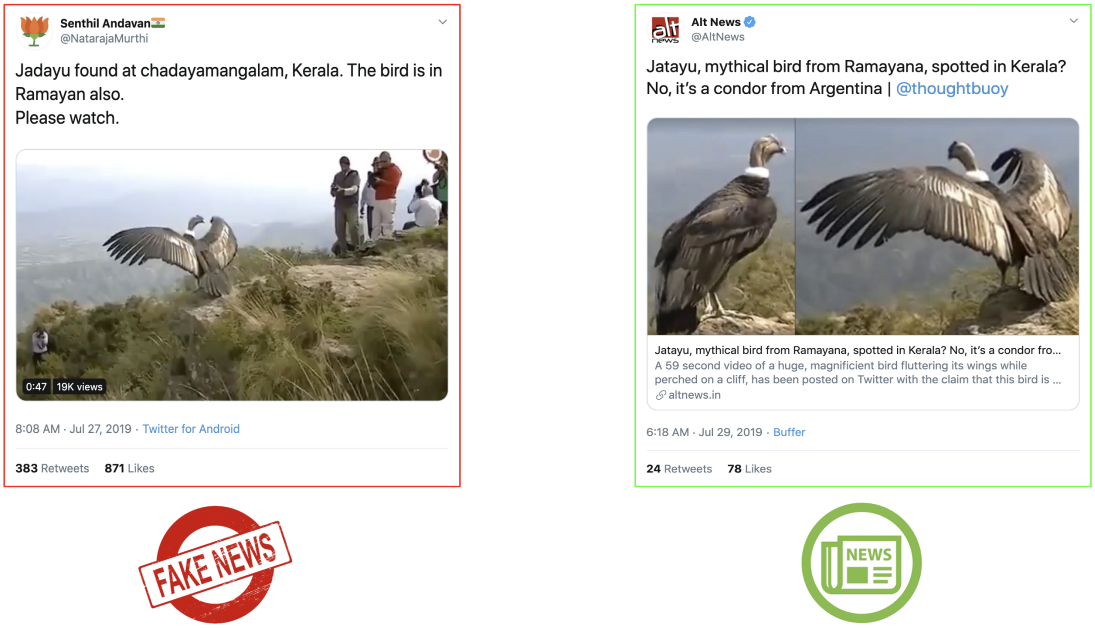


Figure 1.3: An illustration of trust relationship in a retweet network.

Once the trust scores are calculated for each node in the network, TSM normalizes the scores [45] by adhering to the normalization constraint so that both the sum of trustworthiness and the sum of trustingness of all nodes in the network equal to 1. However, a salient problem of such normalization method lies in that the scale of the scores is dependent of the size of the network. When the network is very large, the resulting scores will become extraordinarily small. To deal with the issue, a min-max normalization is performed based on the logarithm of the scores output by TSM to normalize the trustingness and trustworthiness scores into the range of (0,1].

1.4 Epidemiology inspired False Information Mitigation

Epidemiology is the field of medicine which deals with the incidence, distribution and control of infection among populations. In the proposed framework false information is analogous to infection, social network is analogous to population and the likelihood of people believing a news endorser in the immediate neighborhood is analogous to their vulnerability to getting infected when exposed. We consider false information as a



Senthil Andavan🇮🇳
@NatarajaMurthi

Jadayu found at chadayamangalam, Kerala. The bird is in Ramayan also.
Please watch.

0:47 19K views

8:08 AM · Jul 27, 2019 · Twitter for Android

383 Retweets 871 Likes

Alt News
@AltNews

Jatayu, mythical bird from Ramayana, spotted in Kerala? No, it's a condor from Argentina | @thoughtbuoy

Jatayu, mythical bird from Ramayana, spotted in Kerala? No, it's a condor fro...
A 59 second video of a huge, magnificient bird fluttering its wings while perched on a cliff, has been posted on Twitter with the claim that this bird is ...
altnews.in

6:18 AM · Jul 29, 2019 · Buffer

24 Retweets 78 Likes

Figure 1.4: Example of false and refutation information.

pathogen that intends to infect as many people as possible. An important assumption we make is that false information of all kinds is generalized as a single infection, unlike in epidemiology where people have different levels of immunity against different kinds of infections (i.e. the framework is information agnostic). Also we do not distinguish bots in the network population. The likelihood of a person getting infected (i.e. believing and spreading the false information) is dependent on two important factors: a) the likelihood of trusting a news endorser (a person is more likely to spread a news without verifying its claim if it is endorsed by a neighbor they trust); and b) the density of its neighborhood, similar to how high population density increases the likelihood of infection spreading, a modular network structure is more prone to false information spreading. After the infection spreading is identified there is a need to de-contaminate the population. A medicinal cure is used to treat the infected population and thus prevent further spreading of infection. In the context of false information, a refutation information can serve this purpose. Refutation information can be defined as true news that

Table 1.1: Mapping epidemiological concepts to false information spreading.

	Epidemiology context	False information spreading context
Infection	Infection	False information
Population	People and communities	Nodes and modular sub-graphs
Vulnerable	Likely to become infection carriers	Likely to become false information spreaders
Exposed	Neighbors are infected	Neighbor nodes are false information spreaders
Spreaders	Infected people	False information spreaders
Prevention	Medication	Refutation news
Control	Immunization	Refutation news
Recovered	Infection cured	Retract false information and/or spread refutation news

fact-checks a false information. Contents from popular fact-checking websites[2] are examples of refutation information. In epidemiology the medicine can have two purposes: As control mechanism (i.e. medication), with the intention to cure infected people (i.e. explicitly inform the false information spreaders about the refutation information) and as prevention mechanism (i.e. immunization), with the intention to prevent uninfected population from becoming infection carriers in future (i.e. prevent unexposed population from becoming false information spreaders). An infected person is said to have recovered if he either decides to retract from sharing the false information or decides to share the refutation information, or both. Mapping of epidemiological concepts to the context of false information spreading is summarized in Table 1.1. Example of false and its refutation information from Twitter is shown in Figure 1.4. The framework is not to be confused with popular information diffusion based models [23] because they a) usually categorize certain nodes and cannot be generalized to all nodes, b) consider only the propagation paths but not the underlying graph structure and c) can be generalized to information diffusion and need not be particular to false information spreading.

1.5 Thesis Overview

To model a person's likelihood to endorse a false information based on their belief in the endorser, we use the Trust in Social Media (TSM) algorithm. TSM [50] is a HITS-styled iterative matrix algorithm which assigns a pair of complementary trust scores, called *Trustingness* and *Trustworthiness* to every node in a social network. The former is defined as the propensity of a node to trust its neighbors in the network, while the

[2] https://www.snopes.com/, https://www.politifact.com/, https://www.altnews.in/

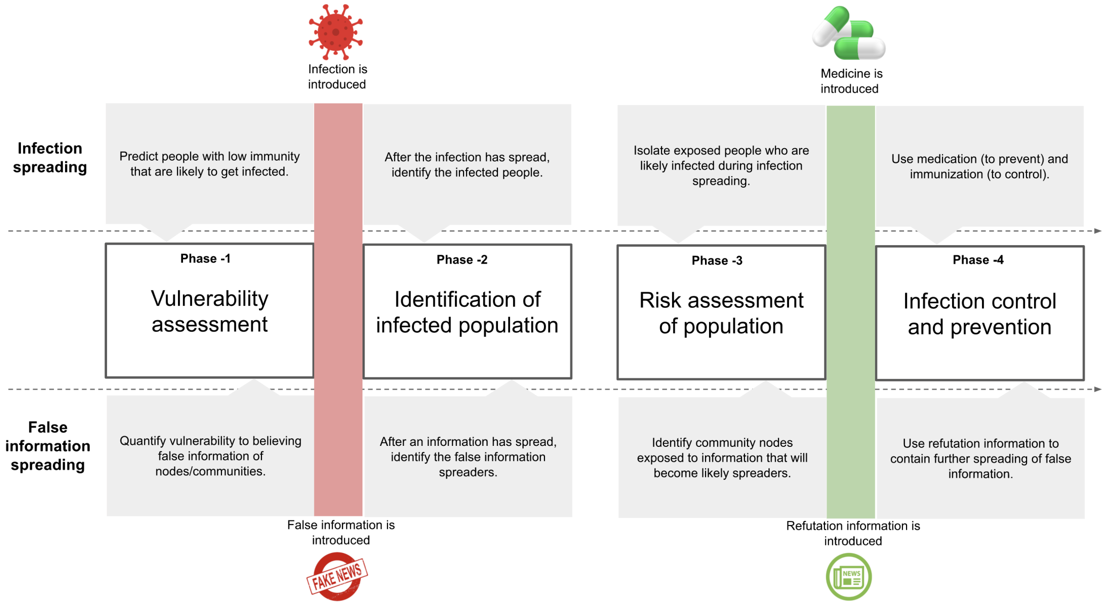

Infection is introduced
Medicine is introduced
Infection spreading
Predict people with low immunity that are likely to get infected.
After the infection has spread, identify the infected people.
Isolate exposed people who are likely infected during infection spreading.
Use medication (to prevent) and immunization (to control).
Phase -1
Vulnerability assessment
Phase -2
Identification of infected population
Phase -3
Risk assessment of population
Phase -4
Infection control and prevention
False information spreading
Quantify vulnerability to believing false information of nodes/communities.
After an information has spread, identify the false information spreaders.
Identify community nodes exposed to information that will become likely spreaders.
Use refutation information to contain further spreading of false information.
False information is introduced
Refutation information is introduced
FAKE NEWS
NEWS

Figure 1.5: Dissertation overview.

latter is defined as the willingness of the network as a whole to trust an individual node.

In the context of epidemiology we propose four sequential phases that fall into the domain of social network analysis. The first phase is called *Vulnerability Assessment*, where we estimate how likely are nodes and communities to believing false information before any information has begun spreading. The second phase is called *Identification of infected population*, where given the complete spread paths of an information, we identify the false information spreaders from true information spreaders. The third phase is called *Risk assessment of population*, where given the partial spread paths of false information, we predict nodes that are most likely to be infected next. The final phase is *Infection control and prevention* where we analyze the impact of refutation information on changing the role of a false information spreader into a true information spreader (i.e. an antidote) and also preventing further spread of false information (i.e. a vaccine). The overview of the epidemiological framework is shown in Figure 1.5.

Explaining each phase more comprehensively, *Vulnerability Assessment* phase [51], proposes novel metrics to quantify the vulnerability of nodes and communities to false information spreading. The role that community structures in determining how people get exposed to false information is explored using the *Community Health Assessment* model. In this model the concepts of neighbor, boundary and core nodes of a community are defined and the vulnerability of a node (individual-level) and a community (group-level) to spreading false information is quantified. The intuition behind the model is that communities are modular structures, where within-group members are highly connected, and across-group members are loosely connected. If boundary nodes of such communities are exposed to false information propagating from neighbor nodes, the likelihood of all core nodes of the community to get infected is high. Through experiments on real world information spreading networks on Twitter, it is showed that the proposed metrics identified the vulnerable nodes for false information networks with higher precision than true news networks, confirming the hypothesis that false information relies strongly on inter-personal trust to propagate while true news does not.

While determining the veracity of information has been widely researched, it is equally important to determine the authenticity of the people who spread information on social media. In *Identification of infected population* phase [52], a novel machine learning based model is built for automatic identification of people spreading false information by leveraging the concept of *believability*, (likelihood of trust formation) i.e., the extent to which the propagated information is likely to be perceived as truthful, based on the trust measures of users in a Twitter network. It is hypothesized that the believability between two users is a function of the trustingness of the retweeter and the trustworthiness of the tweeter. With the retweet network edge-weighted by believability scores, network representation learning is used to generate user embeddings, which is then leveraged to classify users as false information spreaders or not using a recurrent neural network classifier. Based on experiments on a very large real-world rumor dataset collected from Twitter, our method could effectively identify false information spreaders.

An important aspect of preventing false information dissemination is to proactively detect the likelihood of its spreading. In *Risk assessment of population* phase [53], an neural network based model is proposed to identify nodes that are likely to become spreaders of false information. Using the community health assessment model and interpersonal trust an inductive representation learning framework is proposed to predict nodes of densely-connected community structures that are most likely to spread false information, thus making the entire community vulnerable to the infection.

False information and true information refuting it can simultaneously exist in social networks, each competing to influence people in their spread paths. In such scenarios, an efficient strategy for false information containment is to proactively identify if nodes in the spread path are more likely to endorse the false information (i.e. further spread it) or endorse the refuting true information (thereby help quash the false information). In *Infection control and prevention* phase, a graph neural network model using attention mechanism is proposed to predict whether a node will likely endorse false information, endorse refuting true information, or decide to do nothing. Using behavioral data we aggregate interpersonal trust and user credibility features of a node and its neighbors into our deep learning model, network structure and people's sociological and psychological features are effectively integrate to propose an efficient false information suppression strategy.

1.6 Thesis Outline

The outline of this dissertation is as follows: Chapter 2 explains the *Vulnerability Assessment* phase, where I propose the Community Health Assessment model that is used to propose novel node-based and community based vulnerability metrics. Chapter 3 explains the *Identification of infected population* phase, where I propose a recurrent neural network based model to categorize whether a node is a rumor spreader or non-rumor spreader. Chapter 4 explains *Risk assessment of population* phase where I proposed an inductive representation learning based model to identify likely spreaders of a false information. Chapter 5 explains *Infection control and prevention* phase where I proposed a graph neural network model using attention mechanism to predict whether a node will likely endorse false information pr endorse its refutation information. Finally in Chapter 6 I provide concluding remarks, limitations and scope of future work.

Chapter 2

Vulnerability Assessment

2.1 Introduction

The use of social media platforms like Facebook, Twitter and Whatsapp is ubiquitous in modern times, making them powerful tools for information propagation and consumption. However, the good inevitably gets accompanied by the bad, which can be witnessed with the problem of *fake news spreading* [44]. *Fake news* is a recently coined term that refers to fabricated news. *It refers to claims that may have no basis in fact, but are presented as being factually truthful.* It gets spread when someone propagates it via various endorsements such as replying, sharing or re-posting without validating the authenticity of the content.

There is a tremendous amount of interest in the research community to understand fake news spreading, summarized by Sharma et al. [42]. Our approach is orthogonal to these by focusing on *assessing the vulnerability of social networks to false information spreading* Specifically, our focus is on people and the communities they create, with the goal of identifying how vulnerable individuals and communities are to believing false information. We propose the Community Health Assessment model that introduces the ideas of neighbor, boundary and core nodes of a community and proposes novel metrics to quantify the vulnerability of an individual and the community itself. *From a public health perspective, determining whether a piece of news is fake or not is akin to determining whether a virus is injurious to health, while our approach is akin to determining whether an individual or community is vulnerable to being infected by the* *virus*. Thus, the proposed approach provides a complementary perspective, and can be useful in inoculating individuals and communities against spread of fake news.

We propose methods to quantify the likelihood of a boundary node of a community to believe a news item sent from its immediate neighbors, and also quantify the likelihood of a community's entire boundary node set to believing its neighborhood, i.e. the set of nodes outside the community that are connected to at least one member of the community. It is important to note that the method used to quantify vulnerability of a boundary node can be generalized to any node. Intuitively, if an external node infects a member of a community, the likelihood of the entire community to get infected increases due to high connectivity among community members. Thus while assessing vulnerability of community, we focus on examining the influence of information propagated from external nodes into the community rather than considering the internal propagation of the news within the community. We evaluate our model on the propagation networks on multiple real-world information spreading networks from Twitter.

The following novel contributions are made in this chapter:

- We propose the Community Health Assessment model that initiates the ideas of neighbor, boundary and core nodes for a community structure in a social network.

- We propose metrics that help us quantify the vulnerability of node and community to fake news spreading from outside. Health analogy here is that fake news is akin to infection, and quantifying vulnerability is akin to assessing immunity to infection spread.

- We present evaluation of the proposed metrics using two datasets $DS1$ and $DS2$. $DS1$ contains networks for three kinds of information (based on ratings by snopes.com): news which is partially inaccurate (rated as *mixture*), news which is completely inaccurate (rated as *false*) and news whose claim is demonstrably true (rated as *true*). $DS2$ contains 10 news events ($N1 - N10$) from fact checking websites based in India: each having a false information network (F_N), refutation information network(T_N) debunking it and a network obtained by combining F_N and T_N ($F \cup T_N$). We demonstrate that our proposed metrics can much better assess the vulnerability of social networks to false information. To the best our knowledge, this is the first work to measure the vulnerability to fake news spreaders.

Rest of the chapter is organized as follows: We first discuss the related work, followed by explanation of the Community Health Assessment model and the preliminary ideas that it builds upon. We then explain the algorithm to quantify the vulnerability metrics. Next in the Experiments and Results section we explain the data collection process, the datasets used, the metrics used for evaluation and the results. Finally with provide concluding remarks and summarize scope of future work.

2.2 Related Work

We describe briefly prior literature in three broadly related domains of *Misinformation Detection*, *Rumor Spreading Models* and *Computational Trust*.

Misinformation Detection

There has been a surge in interest among researchers over the past few years to build models to detect misinformation. Most approaches in literature model content-based and network-based characteristics of the misinformation. These methods include approaches to capture the style and the language of articles [4], hyperpartisan news content [54] and cues that map language to perceived levels of credibility [55]. Many classification models distinguishing true and fake news have also been proposed. Perez-Rosas et al. [37] proposed a fake news detection model using linguistic features. Yang et al. [56] proposed a classification model using client- and location- based features extracted from micro-blogging websites. Network-based approaches that try to model the propagation structures of false information have also been proposed [57, 58, 52, 20]. The use of neural networks has gained strong attention recently. Use of convolution neural networks [59] and recurrent neural networks [60] have shown promising results.

Rumor Spreading Models

Infection spread models from epidemiology, namely SIR (Susceptible, Infected, Recovered) [61], SIS (Susceptible, Infected, Susceptible) [62], SEIZ (susceptible, exposed, infected, skeptic) [23] and SIHR (Spreaders, Ignorants, Hibernators, Removed) [63] have been widely used to model rumor spreading. Modelling rumor spreading as cascade structures in social networks is also well studied [58, 40]. Other models proposed have tried to identify the rumor spreading source [64, 65]. Fan et al. [66] proposed a model to maximize rumor containment within a fixed number of initial protectors and a given time deadline. Social networks are naturally composed of disjoint communities with relations formed within communities stronger than relations formed across communities. Focusing on such communities to understand rumor spread is a domain with a lot of research potential. Fan et al. [67] proposed an approach to identify a minimal set of boundary nodes that would prevent spread of rumors from neighboring communities, and Nguyen et al. [68] proposed a community-based heuristic method to find the smallest set of highly influential nodes whose decontamination with good information would contain rumor spreading. Vosoughi et al. [40] is another closely related work that tried to empirically investigate the spread of true and false news online.

Computational Trust

Computational social scientists have been interested to quantify the concept of trust in various domains [46] with online social networks being one of them [69]. One of the first works in the area of trust propagation in networks was by Ziegler and Lausen [70]. Some researchers have been interested to understand the role of trust in message propagation during time critical situations [71]. Others have worked to assign scores to nodes in a trust network based on various structural aspects. Kamvar et al. [47] proposed *Eigentrust* to rate trust scores of peers in a P2P network. Mishra and Bhattacharya [48] proposed an iterative matrix algorithm to compute the bias and prestige of nodes in a network. Inspired from the HITS algorithm, Roy et al. [50] proposed the Trust in Social Media (TSM) algorithm to compute a pair of complementary trust scores for every node in a social network, on which our work builds upon.

2.3 Community Health Assessment model

A social network has the characteristic property to exhibit community structures which are formed based on inter-node interactions. Communities tend to be modular groups where within-group members are highly connected, and across-group members are

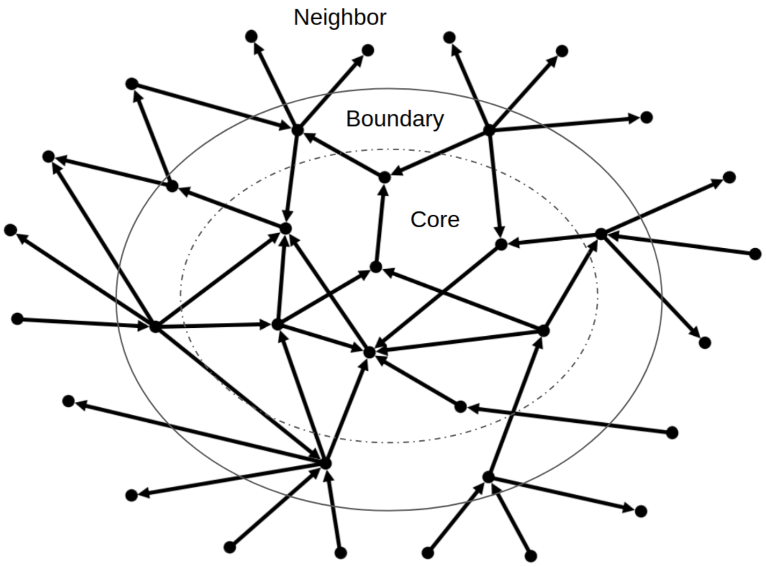


Figure 2.1: Community Health Assessment model.

loosely connected. *Modularity* refers to the ratio of density of edges inside a community to edges outside the community. Thus, based on the edge density, members within a community would tend to have a higher degree of trust among each other than between members across different communities. Also, there is variation in the level of inter-member trust across different communities due to varying modularities. If such communities are exposed to false information being propagated from neighboring nodes, the likelihood of the whole community getting infected would be high. Thus it is important to identify vulnerable communities that lie in the path of false information spreading in order to protect them and thus limit the overall influence of false information in the network. Motivated by this idea we propose the Community Health Assessment model. As part of the modeling, we first propose the ideas of neighbor, boundary and core nodes of a community and then propose metrics to quantify vulnerability of nodes and communities based on the fundamental measures of trust.

Figure 2.1 explains the three groups of nodes with respect to a community which are affected during the process of information spreading, namely:

1. Neighbor nodes: These nodes are directly connected to at least one node of the community. The set of neighbor nodes is denoted by $\mathcal{N}$. They are not a part of the community.

2. Boundary nodes: These are community nodes that are directly connected to at least one neighbor node. The set of boundary nodes is denoted by $\mathcal{B}$. Edges connecting neighbor nodes to boundary nodes are the boundary edges

3. Core nodes: These nodes are only connected to members within the community. The set of core nodes is denoted as $\mathcal{C}$.

2.3.1 Preliminaries

Trustingness and Trustworthiness

In the context of social media, researchers have used social networks to understand how trust manifests among users. A recent work is the Trust in Social Media (TSM) algorithm which assigns a pair of complementary trust scores to each actor, called *Trustingness* and *Trustworthiness* scores [50]. *Trustingness* quantifies the propensity of an actor to trust its neighbors and *Trustworthiness* quantifies the willingness of the neighbors to trust the actor. The TSM algorithm takes a user network, i.e., a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, as input together with a specified convergence criteria or a maximum permitted number of iterations. In each iteration for every node in the network, trustingness and trustworthiness are computed using the equations mentioned below:

$$ti(v) = \sum_{\forall x \in out(v)} \left(\frac{w(v,x)}{1 + (tw(x))^s} \right) \tag{2.1}$$

$$tw(u) = \sum_{\forall x \in in(u)} \left(\frac{w(x,u)}{1 + (ti(x))^s} \right) \tag{2.2}$$

where $u, v, x \in \mathcal{V}$ are user nodes, $ti(v)$ and $tw(u)$ are trustingness and trustworthiness scores of v and u, respectively, $w(v,x)$ is the weight of edge from v to x, $out(v)$ is the set of outgoing edges of v, $in(u)$ is the set of incoming edges of u, and s is the involvement score of the network. Involvement is basically the potential risk an actor takes when creating a link in the network, which is set to a constant empirically. Once the trust scores are calculated for each node in the network, TSM normalizes the scores by adhering to the normalization constraint so that both the sum of trustworthiness and the sum of trustingness of all nodes in the network equals to 1. However, a salient problem of such normalization method lies in that the scale of the scores is dependent on the size of the network. When the network is very large, the resulting scores will become extraordinarily small. To deal with the issue, min-max normalization based on the logarithm of the scores output by TSM can be used to normalize the scores into the range of (0,1]. Details about the TSM algorithm can be found in [45].

Believability

Believability is an edge score derived from Trustingness and Trustworthiness scores [52]. It helps us to quantify the potential or strength of directed edges to transmit information by capturing the intensity of the connection between the sender and receiver. Believability for a directed edge is computed as a function of the trustworthiness of the sender and the trustingness of the receiver.

More specifically, given users u and v in the context of microblogs such as Twitter, a directed edge from u to v exists if v reads a tweet from u. The believability quantifies the strength that v trusts on u when v decides to retweet a news endorsed by u. Therefore, v is very likely to believe in u if:

I. u has a high trustworthiness score, i.e., u is highly likely to be trusted by other users in the network, or

II. v has a high trustingness score, i.e., v is highly likely to trust others.

So, the believability score is supposed to be proportional to the two values above, which can be jointly determined and computed as follow:

$$Believability(u \to v) = tw(u) * ti(v) \tag{2.3}$$

The idea has been previously applied in [52] where a classification model was built to identify rumor spreaders in Twitter user network based on believability measure.

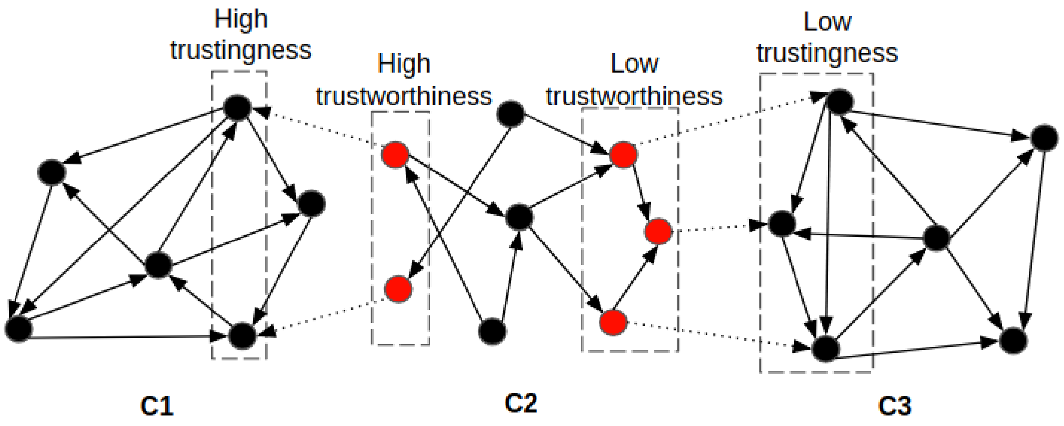


Figure 2.2: Illustration of vulnerability to false information spreading.

2.3.2 Vulnerability metrics

Motivation

False information generally gets no coverage from mainstream news platforms (such as press or television), so an important factor contributing to a user's decision to spread a fake news on social media is its inherent trust on its neighbor endorsing it. On the other hand, a user would most likely endorse a true news since it is typically endorsed by multiple credible news sources. *We hypothesize that the less credible nature of false information makes it much more reliant on user trust for spreading than true news does.* Thus, we propose our vulnerability metrics based upon the idea of computational trust, particularly believability, for assessing the health of individuals and communities encountering false information.

Illustrative Example

We first illustrate the idea of our proposed vulnerability metrics through figure 2.2. Red nodes in community C2 represent fake news spreaders. C1 and C3 are two other communities having identical structure. We see that C3 and C1 have 3 and 2 boundary nodes respectively that are directly connected to the fake news spreaders (represented through dotted lines). Based on edge count one would believe that C3 is more vulnerable to fake news spreading than C1. But the boundary nodes of C3 having low trustingness scores are connected to spreaders having low trustworthiness scores, while the boundary nodes of C1, which have high trustingness scores, are connected to spreaders having high trustworthiness scores. Therefore, our metric should be able to identify C1 as more vulnerable than C3.

With the believability (Eq. 2.3) which is based on trustingness and trustworthiness derived from the TSM algorithm, we now derive the metrics to quantify vulnerability of nodes and communities to false information spreading. *Vulnerability Metrics* help us quantify the likelihood of boundary nodes and communities to believe an information spreading from their neighbors. In healthcare terminology, this analysis tries to model the immunity of the system. We assume that the information spreading is widespread outside of the community, i.e., at least some of the neighbor nodes of the community are spreaders. We define the node- and community-level metrics as follows:

I. *Vulnerability of boundary node,* $V(b)$: This metric measures the likelihood of a boundary node b to become a spreader. The metric is derived as follows: The likelihood of node b to believe an immediate neighbor n is a function of the trustworthiness of the neighbor n ($n \in \mathcal{N}_b$, where $\mathcal{N}_b$ is the set of all neighbor nodes of b) and the trustingness of b, and is quantified as $bel_{nb} = tw(n) * ti(b)$, that is, $Believability(n \to b)$. Thus, the likelihood that b is *not* vulnerable to n can be quantified as $(1-bel_{nb})$. Generalizing this, the likelihood of b *not* being vulnerable to all of its neighbor nodes is $\prod_{\forall n \in \mathcal{N}_b}(1 - bel_{nb})$. Therefore, the likelihood of b to believe any of its neighbors, i.e. the vulnerability of the boundary node b is computed as:

$$V(b) = 1 - \prod_{\forall n \in \mathcal{N}_b} (1 - bel_{nb}) \tag{2.4}$$

II. *Vulnerability of community,* $\widetilde{V}(C)$: This metric measures likelihood of the boundary node set of a community C ($\mathcal{B}_C$) to believe an information from any of its neighbors. The metric is derived as follows: Going forward with the idea in 1), the likelihood that boundary node b is *not* vulnerable to its neighbors can be quantified as $(1 - V(b))$. Generalizing this to all $b \in \mathcal{B}_C$, the likelihood that none of the boundary nodes of a community are vulnerable to their neighbors can be quantified as $\prod_{\forall b \in \mathcal{B}_C}(1 - V(b))$. Thus, the likelihood of community C being vulnerable to any its neighbors, i.e., the vulnerability of the community, is defined as:

$$\widetilde{V}(C) = 1 - \prod_{\forall b \in \mathcal{B}_C}(1 - V(b)) \tag{2.5}$$

The pseudo-code of algorithm to generate the vulnerability metrics is provided in Algorithm 1 and pictorially represented in Figure 2.3.

Algorithm 1 Vulnerability Metrics Computation

```
Input: $\mathcal{G}(\mathcal{V}, \mathcal{E})$: Spreader's follower-following network
Output: $V(b)$: Vulnerability of each boundary node, and
$\widetilde{V}(C)$: Vulnerability of each community
$(ti, tw)_{\forall v \in \mathcal{V}} \leftarrow$ Trust scores using TSM($\mathcal{G}$)
$\phi \leftarrow$ Disjoint communities in $\mathcal{G}$
$C \leftarrow$ A community s.t. $C \in \phi$
$\mathcal{B}_C \leftarrow$ Set of Boundary nodes for community $C$
$\mathcal{N}_b \leftarrow$ Set of Neighbor nodes for boundary node $b$
for each $C \in \phi$ do
    for each $b \in \mathcal{B}_C$ do
        for each $n \in \mathcal{N}_b$ do
            $bel_{nb} = tw(n) * ti(b)$
        end
        $V(b) = 1 - \prod_{\forall n \in \mathcal{N}_b}(1 - bel_{nb})$
    end
    $\widetilde{V}(C) = 1 - \prod_{\forall b \in \mathcal{B}_C}(1 - V(b))$
end
```

2.4 Experiments and Results

2.4.1 Dataset and setup

We collected two sets of network datasets $DS1$ and $DS2$, summarized in Figure 2.4 for tweets associated with news articles with confirmed ground truth from various fact checking websites. $DS1$ contains 12 news networks categorized the news into three types: News $M1$, $M2$, $M3$ and $M4$ are labelled as *Mixture* which indicates that the news has significant elements of both truth and falsity in it, news $F1$, $F2$, $F3$ and

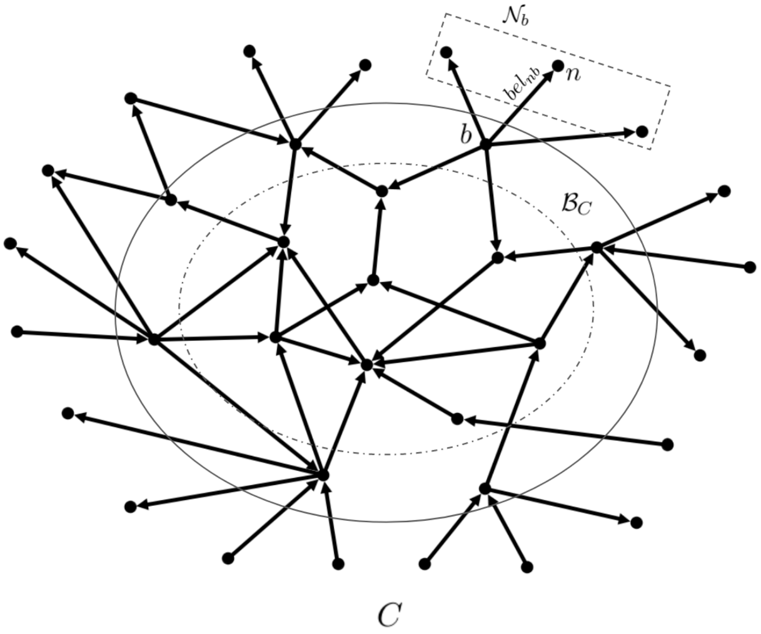


Figure 2.3: Vulnerability metrics using Community Health Assessment model.

$F4$ are labelled as *False* which indicates that the primary elements of the news are basically false, and news $T1$, $T2$, $T3$ and $T4$ are labelled as *True* which indicates that the primary elements of a claim are basically true. $DS2$ contains news events $N1 - N10$ with each news event containing false information network: F_{N1}-F_{N10}, its corresponding refutation information network: T_{N1}-T_{N10} . Refutation information can be defined as true information that fact checks a specific item of false information. It is created soon after a false information is identified and tends to co-exist with the false information. $F \cup T_{N1}$-$F \cup T_{N10}$ denotes network obtained by combining false and refutation networks for specific news events. The metadata of $DS1$ and $DS2$ is described in Table 2.1 and Table 2.2 respectively.

We identified the specific source tweet related to each information in question. For evaluation of metrics, we then identified all the spreaders of the source tweet associated with the news, which comprised of the source tweeter (identified using Twitter API) and the list of retweeters (accessible through *twren.ch* or the Twitter search API). We considered the follower-following network of the spreaders obtained from Twitter

Table 2.1: Metadata for $DS1$.

Type	ID	Snopes link
Mixture	*M1*	www.snopes.com/fact-check/nike-workers-pay-kaepernick/
	M2	www.snopes.com/fact-check/virginia-prisons-tampons/
	M3	www.snopes.com/fact-check/sheriff-nike-shirt-mugshots/
	M4	www.snopes.com/fact-check/opportunity-rovers-final-words/
False	*F1*	www.snopes.com/fact-check/were-hate-charges-blm-kidnappers-dropped/
	F2	www.snopes.com/fact-check/german-news-trump-nato/
	F3	www.snopes.com/fact-check/jussie-smollett-cnn-job/
	F4	www.snopes.com/fact-check/kamala-harris-jussie-smollett/
True	*T1*	www.snopes.com/fact-check/eva-ramon-gallegos-hpv/
	T2	www.snopes.com/fact-check/nz-prime-minister-massacre-aid/
	T3	www.snopes.com/fact-check/betsy-devos-special-olympics/
	T4	www.snopes.com/fact-check/texas-governor-tweet-rapist/

API, as a proxy for trust. Code implementation and sample dataset is also provided[1]. To evaluate our proposed metrics we used the collection of the twelve different news spreading networks. We ran the TSM algorithm [50] on follower-following network to compute the trustingness and trustworthiness scores for every node in the network. We then identified disjoint communities by trying three popular community detection algorithms on large networks: Louvain [72] solves an optimization problem that tries to maximize modularity of communities. Infomap [73] algorithm is based on the principles of Information Theory. In contrast to maximizing modularity, the fundamental approach of Infomap is to utilize flows in the graph. It uses the map equation framework, which characterizes community detection as a problem of finding a description of minimum information of a random walk process. Label Propagation [74] starts off by assigning a unique label to each node, and then iteratively assigns each node the label most common amongst its neighbors. As a greedy algorithm, Label Propagation is more efficient which is linear to the number of edges in the graph. For each of the communities generated we identified the sets of boundary and neighbor nodes and then computed vulnerability metrics (see Algorithm 1).

The network statistics based on Community Health Assessment model for $DS1$ shown in Table 2.3, and for false, refutation and the combined network in $DS2$ is

[1] https://github.com/BhavtoshRath/Vulnerability-Metrics

Table 2.2: Metadata for $DS2$.

ID	Link of debunked article
N1	www.altnews.in/bjp-mla-raja-singh-plagiarises-pakistan-army-song-dedicates-it-to-indian-army/
N2	www.altnews.in/amit-malviya-targets-yogendra-yadav-via-edited-video-clip-after-tv-debate-face-off/
N3	www.altnews.in/shivraj-singh-chouhan-tweets-clipped-video-to-portray-gaffe-by-rahul-gandhi-in-poll-speech/
N4	www.boomlive.in/pragya-thakur-was-not-4-years-old-at-the-time-of-babri-masjid-demolition/
N5	www.altnews.in/2017-video-from-gujarat-shared-as-pm-narendra-modis-rally-in-hyderabad/
N6	www.altnews.in/no-a-bjp-candidate-from-west-bengal-did-not-dress-up-as-hanuman/
N7	www.boomlive.in/did-sp-workers-jump-the-gun-with-a-pm-akhilesh-billboard-not-quite/
N8	https://navbharattimes.indiatimes.com/viral-adda/fake-news-buster/news-about-former-srilankan-cricketer-sanath-jayasuriyas-death-is-a-hoax-24858/
N9	https://smhoaxslayer.com/%E2%80%8Bimported-dogs-stone-pelters-for-kasmir-or-imported-entire-video-for-inciting-communal-hatred/
N10	www.altnews.in/hindi/no-mohammad-barkat-ali-is-not-a-regular-audience-of-ndtv/

shown in Tables 2.4, 2.5 and 2.6 respectively. Community size plots are shown in 2.5. We observe that the datasets contain varying number of communities, ranging from as low as 4/2/7 to as high as 99/2497/1637 with respect to Louvain (L)/Infomap (I)/Label Propagation (LP).[2] A general observation is that Label Propagation algorithm tends to generate more number of communities while Infomap generates fewer number of communities. Louvain gives more balanced results in terms of size and count of communities.

2.4.2 Evaluation of metrics

To measure how good the proposed metrics are able to quantify the vulnerability of nodes and communities, we evaluate the quality of ranking on boundary nodes and communities based on vulnerability scores in comparison with the ground-truth ranking of nodes and communities derived from the news spread in the network. We adopt the ranking evaluation measures widely used in Information Retrieval literature [75].

[2] For the rest of the chapter, L: Louvain, I: Infomap, LP: Label Propagation for all tables.

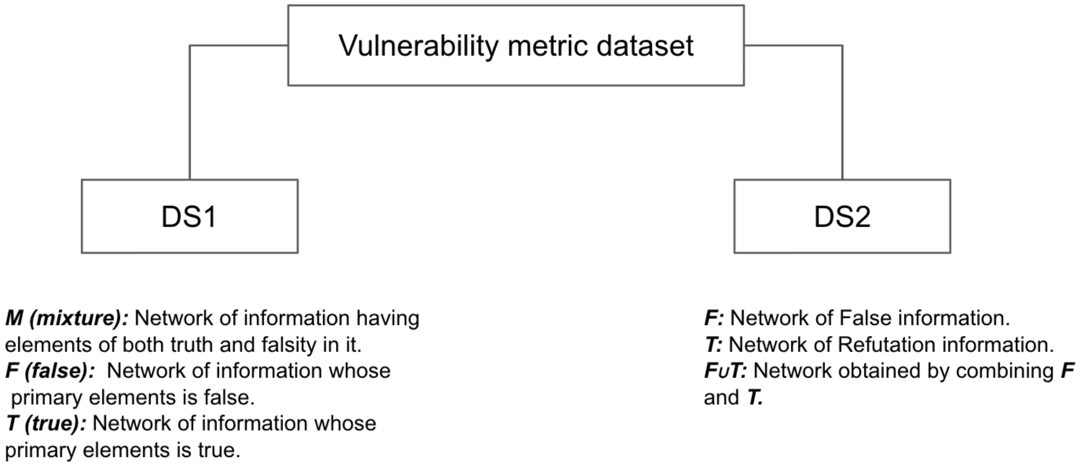

Vulnerability metric dataset
DS1
DS2
M (mixture): Network of information having elements of both truth and falsity in it.
F (false): Network of information whose primary elements is false.
T (true): Network of information whose primary elements is true.
F: Network of False information.
T: Network of Refutation information.
F∪T: Network obtained by combining **F** and **T.**

Figure 2.4: Dataset summary of $DS1$ and $DS2$.

Evaluation of $V(b)$

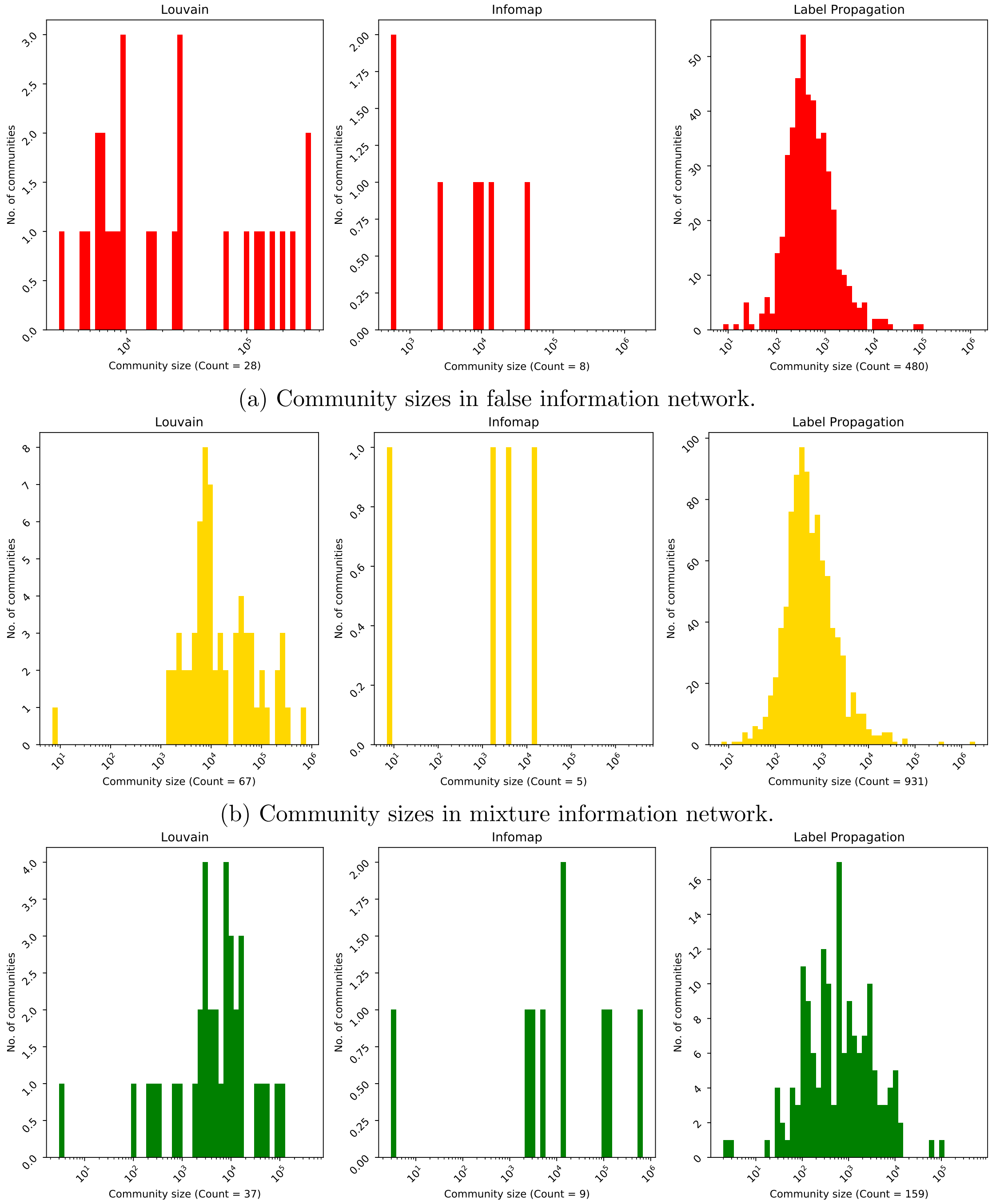
A vulnerable boundary node is highly likely to have strong believability with its neighbors. We thus consider the ground truth of a vulnerable node as a node which retweets. The ground truth vulnerability of boundary nodes is binary as we only have information of whether the node retweets or not. We thus evaluate this metric using *Average Precision@k* and *Mean Average Precision*.

Average Precision@k (AP@k): We first compute Precision@k (viz. top-k vulnerable boundary nodes based on the metric as a percentage of spreader boundary nodes in a community) and then compute the Average Precision@k ($AP@k$) (viz. the average of Precision@k values over all communities in a network).

Mean Average Precision (MAP): Mean Average Precision is computed as the mean of the average precision scores for the top-k boundary nodes over all communities in a network. The formula to compute MAP is given by $\sum_{k=1}^{K} AP(k)/K$, where K denotes total number of communities in the network.

Evaluation of $\widetilde{V}(C)$

A community with more number of spreader boundary nodes is more vulnerable to news penetration. As most communities of a network have at least few spreader boundary nodes, it is not feasible to use node ranking metrics above for evaluating community



(a) Community sizes in false information network.

(b) Community sizes in mixture information network.

(c) Community sizes in true information network.

Figure 2.5: Frequency distribution of community sizes in information spreading networks.

vulnerability. We thus rank the communities by their vulnerability scores and compare with the ground-truth ranking given by the relative count of spreader boundary nodes in the community. We use Kendall's tau, which is a correlation measure for ordinal data, as evaluation metric. Kendall's tau close to 1 indicates strong agreement, and that close to -1 indicates strong disagreement between evaluated and ground-truth rankings.

Kendall's tau (τ): Let $rel = [rel_1, rel_2, \ldots, rel_n]$ represent the 'relevant' ranked list of n communities based on ground-truth vulnerability (quantified as the fraction of boundary nodes that are spreaders), and $ret = [ret_1, ret_2, \ldots, ret_n]$ represent the 'retrieved' ranked list of communities based on our proposed vulnerability metric. Let P represent the # of concordant pairs, Q the # of discordant pairs, T the # of ties only in rel, and U the # of ties only in ret. If a tie occurs for the same pair in both rel and ret, it is not added to either T or U. Then we calculated $\tau = (P - Q)/sqrt((P + Q + T) * (P + Q + U))$.

2.4.3 Results for $DS1$

Table 2.7 shows the evaluation results for the proposed metric assessing the vulnerability of boundary nodes for $DS1$. For the twelve networks we show the Average Precision for k = 1, 5, 10 and 15 and compute the MAP for the top-15 results.

AP@1 shows how well we are able to identify the first spreader boundary node based on our metric. Our metric is able to identify the most vulnerable boundary node in AP of 0.712 averaged over the mixture news networks, 0.91 averaged over the false news networks and 0.471 averaged over the true news networks for Louvain; 0.695 averaged over the mixture news networks, 0.923 averaged over the false news networks and 0.459 averaged over the true news networks for Informap, and 0.811 averaged over the mixture news networks, 0.915 averaged over the false news networks and 0.74 averaged over the true news networks for Label Propagation. Thus, we are able to identify the most vulnerable boundary node of communities in false news networks with average precision of over 90%. As expected, our metrics show better performance particularly for fake news networks, followed by mixture and then true news networks. Average precision for rest of the k-values also shows similar trend.

Metrics for Louvain-/Infomap-based communities follow a similar trend for the remaining k values. However, Label Propagation communities for k=3 evaluate with AP

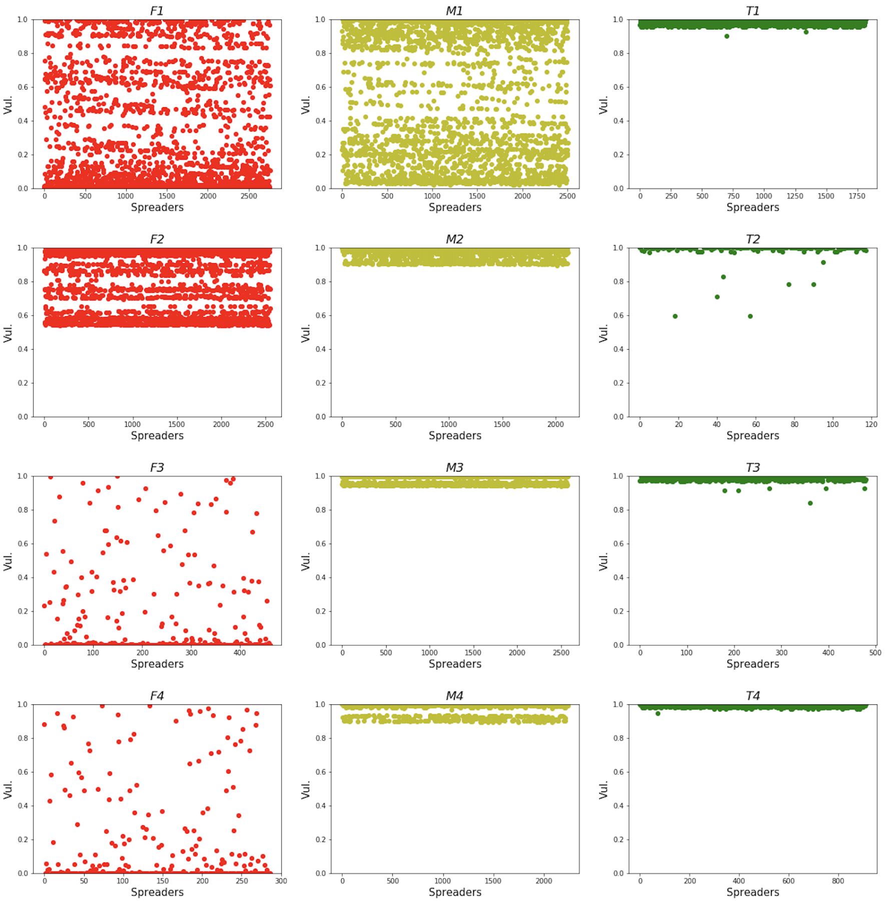


Figure 2.6: Distribution of vulnerability score of spreaders in $DS1$.

of 90.25% averaged over the false news networks, which is over 35% and 20% better than the mixture and the true networks, respectively. In this case, true news networks are ranked better than mixture news networks. While k=5 also shows a similar trend, for the rest of the k values Label Propagation-based communities show better performance for the mixture than the true news networks. This insensitivity in evaluation could be attributed to the fact that label propagation algorithm tends to generate more number of communities. Thus, the average community size is much smaller, causing the communities to have sparser boundary and neighbor node sets.

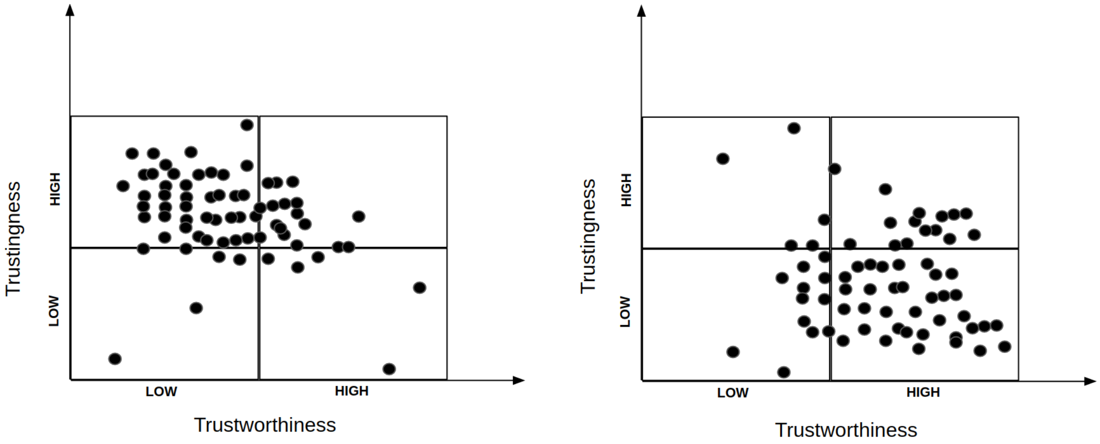
We also observe that the MAP averaged over the false news networks is 47.86% better than the mixture and 150% better than the true news networks for Louvain-based communities; and 25.94% better than the mixture, and 139.9% better than the true news networks for Infomap-based communities; and 33.72% better than the mixture and 37.14% better than the true news networks for Label Propagation-based communities. Therefore, we are able to identify most vulnerable boundary nodes of communities in false news networks with an average MAP of over 75%.

Table 2.8 shows the evaluation results for proposed metric to compute the vulnerability of a community for $DS1$. For the twelve networks the table shows Kendall's tau value (τ) for communities generated using the three algorithms. We observe that the τ for mixture and true news networks tend to have a negative correlation with the ground truth community ranking. False news networks on the other hand show a positive correlation, with high values of 0.642 for F1, 0.667 for F2, 0.457 for F3 and 0.714 for F4. For modular communities generated using Louvain heuristics, our proposed metrics evaluate all false news networks with a positive correlation (average correlation: 0.208) while all true news networks are evaluated with a negative correlation (with average correlation -0.105). Three of the four mixture news networks also have a negative correlation (with average correlation -0.0095). Therefore, our proposed metrics are confirmed to produce better performance on fake news networks, compared to the true and mixture ones.

Figure 2.6 show the vulnerability scores of news spreaders of false, mixture and true news networks respectively. We observe that the scores of spreaders in false news networks have more variance (points are more spread out between 0 and 1) than spreaders in mixture news networks. Mixture news networks (except M1) have lesser variance, while true news spreaders have least variance. Thus we can conclude that trust-based vulnerability metrics are able to distinguish between spreaders with high (1) and low (0) vulnerability better than true news spreaders (where most spreaders are assigned similar scores). This in turn affects the performance of community vulnerability metrics in a similar way.

Case study of *mixture* news spreaders

On observing the trustingness and trustworthiness scores of the spreaders of mixture news networks as shown in Figure 2.7 we notice that most spreaders of *M1* have high trustingness and low trustworthiness scores compared to *M2*, *M3* and *M4* that have



a) Spreaders in *M1*

b) Spreaders in *M2*, *M3*, *M4*.

Figure 2.7: Case study of spreaders in $Mixture$ networks.

low trustingness and high trustworthiness scores. Source of *M1* was tweeted by a conservative with political undertones and it is known that conservatives are more likely to share fake news [76]. The information shows spreading pattern similar to fake news, as spreaders with high trustingness score shared *M1* without fact checking the claim, unlike the source and spreaders of *M2*, *M3* and *M4* who are not political conservatives.

2.4.4 Results for $DS2$

Table 2.9 shows the evaluation results for vulnerability assessment of boundary nodes for DS1. For the thirty networks (three each for the ten news events) we show the Average Precision for k = 1, 5, 10 and 15 and compute the MAP for the top-15 results. Based on the AP@1, we show that our metric is able to identify the most vulnerable boundary node with average precision (aggregated over all news events) of 0.735, 0.672, 0.694 for false, refutation and combined networks repectively when communities are generated using Louvain; 0.705, 0.501, 0.628 when communities are generated using Infomap and 0.744, 0.501, 0.577 when communities are generated using Label propagation method. As in $DS1$, we observe that our propsoed metrics are able to identify spreaders in false information network with higher precision than spreaders in refutation information networks. This can be attributed to the fact that a person's motivation to spread refutation information (whose validity is more certain) is driven more by the nature of

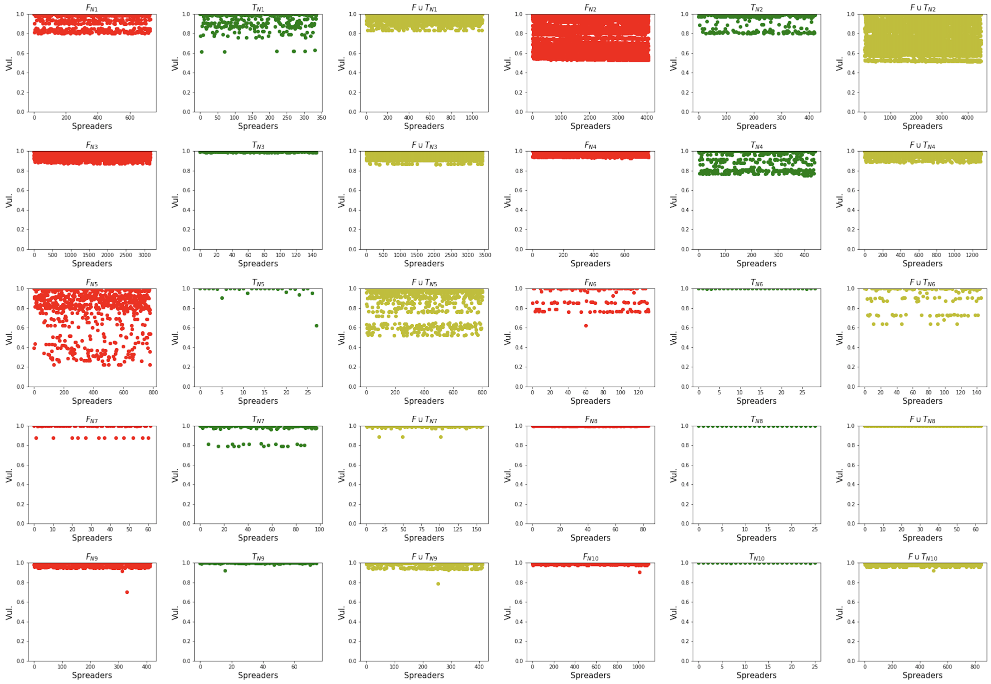


Figure 2.8: Distribution of vulnerability score of spreaders in $DS2$.

the content; unlike false information (whose content is not validated) which is driven less by the content on more by the trust dynamics with the endorser. Metric's performance in identifying false information spreaders in combined network affected slightly due to the presence of refutation information spreading dynamics, but is still better than only refutation information network.

Trends do not drastically vary for other values of k, with Label Propagation performing slightly better than Louvain while Infomap with lowest performance. ALos we observe than certain vulnerability scores are drastically low. This can be attributed to the quality of disjoint communities generated by the community detection algorithm. In scenarios where the number of communities is too low or too large, this causes large variation in the boundary and neighbor node count for the community thus affecting the metric score computation.

Through MAP we aggregate the precision scores for top-15 spreader boundary nodes. We observe precison scores of 0.626. 0.366, 0.766 for fale information network; 0.449/

0.152/ 0.51 for refutation information network; 0.652/ 0.457/ 0.759 for combined network using L/ I/ LP.

Table 2.10 shows the evaluation results for proposed metric to compute the vulnerability of a community for $DS2$. Similar to Table 2.8, τ for false information networks tend to have more values greater than zero (i.e. positive correlation) compared to refutation information networks. Figure 2.8 shows a similar trend as Figure 2.6, with spreaders in false information networks showing more variance than refutation information networks and comparable variance with combined networks for most news events.

2.5 Conclusions and Future Work

We propose novel metrics based on the concept of believability derived from computational trust measures to compute vulnerability of nodes and communities to news spread and show that the metrics is much more sensitive to false information. We confirm our hypothesis that false information have to rely on strong trust among spreaders to propagate while true or refuting information does not. Through experiments on two datasets of large information spreading networks on Twitter we show that our proposed metrics can identify the vulnerable nodes and communities with high precision. While detection of fake news spreading is a widely studied problem, its containment is not. We believe that the proposed model can be used to identify vulnerable individuals and communities to build content-agnostic fake news spread prevention models. We thus propose the *Community Health Assessment* model as a preliminary idea that exploits the structural characteristics of social networks to identify nodes and communities that are most vulnerable to news spreading.

As part of future work we would like to extend the proposed ideas to understand the dynamics of news spreading within a community (i.e. through core nodes). We would also like to include temporal features of news spreading into our model.

Table 2.3: Community statistics for $DS1$.

Information	Community Detection	# of communities (C)	Avg. # of nodes / C	Avg. # of infected nodes / C	Avg. # of $\mathcal{B}$ edges	Avg. # of $\mathcal{B}$	Avg. # of neighbor nodes	Avg. # of infected $\mathcal{B}$ nodes	Avg. # of infected $\mathcal{N}$ nodes
M1	Louvain	54	45,004	53	69,040	7,107	14,401	47	774
	Infomap	36	68,148	81	5,594	1,778	1,408	59	376
	Label Propagation	786	3,038	4	603	215	266	3	38
M2	Louvain	67	54,764	34	28,250	3,300	13,717	32	494
	Infomap	5	733,843	459	1274	716	453	74	120
	Label Propagation	931	3,941	2	1,080	264	620	2	50
M3	Louvain	72	89,756	39	20,406	2,878	11,371	36	412
	Infomap	14	461,604	202	49,791	7,848	20,097	186	558
	Label Propagation	1,341	4,819	2	1,150	240	702	2	60
M4	Louvain	99	35,477	27	10,606	2,285	2,996	23	484
	Infomap	37	94,924	72	16,081	3,933	3,764	66	480
	Label Propagation	1,637	2,146	2	709	191	292	2	50
F1	Louvain	28	67,262	103	218,939	14,547	34,442	99	1,028
	Infomap	8	235,416	360	1,482	775	616	81	143
	Label Propagation	480	3,924	6	933	340	455	5	40
F2	Louvain	50	99,626	57	51,664	5,793	21,101	51	660
	Infomap	4	1,245,330	708	1,454	760	637	89	118
	Label Propagation	677	7,358	4	2,318	396	1,542	4	84
F3	Louvain	15	52,147	31	417,933	16,382	52,259	31	365
	Infomap	133	5,881	3	6,722	1,075	3,225	3	157
	Label Propagation	15	52,147	31	5,227	2,285	2,514	24	83
F4	Louvain	15	33,544	19	338,248	13,848	56,711	19	246
	Infomap	38	13,241	8	11,255	2,182	5,484	8	171
	Label Propagation	7	71,880	41	1,779	992	744	22	64
T1	Louvain	47	232,538	59	47,189	2,171	42,783	39	246
	Infomap	34	321,450	82	5,792	1,390	2,261	52	189
	Label Propagation	1,283	8,519	2	2,151	202	1,724	2	54
T2	Louvain	37	25,758	5	4,150	509	3,095	3	36
	Infomap	9	105,893	22	5,650	1,418	1,777	17	60
	Label Propagation	159	5,994	1	1,102	189	752	1	25
T3	Louvain	27	79,849	26	10,135	1,942	5,251	18	180
	Infomap	629	3,428	1	1,266	161	641	1	124
	Label Propagation	209	10,315	3	1,138	303	584	3	46
T4	Louvain	89	17,202	12	4,511	908	1,502	10	205
	Infomap	1,206	1269	1	544	92	271	1	99
	Label Propagation	797	1,921	1	723	164	279	1	53

Table 2.4: Community statistics for false information in $DS2$.

Information	Community Detection	# of communities (C)	Avg. # of nodes / C	Avg. # of infected nodes / C	Avg. # of $\mathcal{B}$ edges	Avg. # of $\mathcal{B}$	Avg. # of neighbor nodes	Avg. # of infected $\mathcal{B}$ nodes	Avg. # of infected $\mathcal{N}$ nodes
F_{N1}	Louvain	37	23,935	25	9,654	1,482	3,116	20	163
	Infomap	3	295,199	314	17,466	4,786	3,224	159	373
	Label Propagation	220	4,025	4	1,299	322	623	4	46
F_{N2}	Louvain	66	39,510	69	35,877	2,274	16,655	62	562
	Infomap	6	434,605	759	1	1	1	1	1
	Label Propagation	280	9,313	16	2,148	250	1,571	9	40
F_{N3}	Louvain	53	44,215	65	23,464	2,774	8,280	59	443
	Infomap	2497	956	1	926	67	558	1	117
	Label Propagation	313	7,628	11	1,519	347	955	7	40
F_{N4}	Louvain	37	55,031	24	8,758	1,102	6,539	20	144
	Infomap	2	1,018,081	447	3,744	1,123	1,959	75	84
	Label Propagation	214	9,515	4	1,918	299	1,356	4	43
F_{N5}	Louvain	47	11,037	17	10,685	1,617	3,319	17	234
	Infomap	738	703	1	1,107	107	587	1	155
	Label Propagation	119	4,359	7	1,646	426	848	5	49
F_{N6}	Louvain	26	10,653	6	2,140	422	1,085	5	50
	Infomap	4	69,246	40	1,734	584	564	17	64
	Label Propagation	97	2,856	2	768	152	379	2	29
F_{N7}	Louvain	20	7,230	4	1,261	381	324	3	27
	Infomap	117	1,236	1	337	74	92	1	29
	Label Propagation	35	4,131	2	724	245	207	2	16
F_{N8}	Louvain	17	23,188	7	1,479	308	911	5	49
	Infomap	4	98,551	30	3,439	1,060	1,253	17	60
	Label Propagation	83	4,749	1	494	117	200	1	22
F_{N9}	Louvain	43	11,092	11	3,673	802	1,219	10	135
	Infomap	487	979	1	538	77	224	1	90
	Label Propagation	162	2,944	3	830	221	356	3	32
F_{N10}	Louvain	55	19,506	22	5,681	853	2,455	20	200
	Infomap	1,045	1,027	1	570	52	281	1	98
	Label Propagation	216	4,967	6	1,066	220	641	5	33

Table 2.5: Community statistics for refutation information in $DS2$.

Information	Community Detection	# of communities (C)	Avg. # of nodes / C	Avg. # of infected nodes / C	Avg. # of $\mathcal{B}$ edges	Avg. # of $\mathcal{B}$	Avg. # of neighbor nodes	Avg. # of infected $\mathcal{B}$ nodes	Avg. # of infected $\mathcal{N}$ nodes
T_{N1}	Louvain	40	11,338	10	5,856	856	3,018	8	96
	Infomap	2	226,769	200	1,260	606	151	37	58
	Label Propagation	154	2,945	3	1,564	274	1,019	2	39
T_{N2}	Louvain	47	9,226	10	3,562	540	1,648	9	103
	Infomap	472	919	1	581	58	327	1	80
	Label Propagation	167	2,597	3	1,042	169	641	3	34
T_{N3}	Louvain	15	86,491	32	7,305	987	5,160	10	67
	Infomap	457	2,839	1	757	64	437	1	80
	Label Propagation	84	15,445	6	1,497	260	1,032	5	29
T_{N4}	Louvain	45	23,522	11	5,399	590	3,950	10	102
	Infomap	523	2,024	1	740	58	502	1	77
	Label Propagation	214	4,946	2	1,211	167	827	2	39
T_{N5}	Louvain	15	17,513	6	4,895	305	4,376	2	28
	Infomap	2	131,346	46	5,650	936	2,874	5	45
	Label Propagation	40	6,567	2	1,769	159	1,449	2	19
T_{N6}	Louvain	9	7,458	4	772	220	333	3	10
	Infomap	103	652	1	106	25	48	1	16
	Label Propagation	26	2,582	1	376	112	99	1	13
T_{N7}	Louvain	20	3,067	6	1,280	267	478	5	38
	Infomap	2	30,666	57	1,636	871	370	26	22
	Label Propagation	49	1,252	2	648	149	290	2	21
T_{N8}	Louvain	4	310,826	23	2,152	233	1,723	7	31
	Infomap	2	621,653	47	1,968	601	943	15	42
	Label Propagation	64	19,427	1	465	98	208	1	21
T_{N9}	Louvain	20	13,821	5	1,482	324	836	4	37
	Infomap	3	92,143	32	233	81	132	9	16
	Label Propagation	78	3,544	1	564	120	244	1	28
T_{N10}	Louvain	5	29,757	7	1,098	283	643	5	20
	Infomap	49	3,036	1	214	54	84	1	14
	Label Propagation	31	4,800	1	347	119	90	1	13

Table 2.6: Community statistics for false and refutation information network combined in $DS2$.

Information	Community Detection	# of communities (C)	Avg. # of nodes / C	Avg. # of infected nodes / C	Avg. # of $\mathcal{B}$ edges	Avg. # of $\mathcal{B}$	Avg. # of neighbor nodes	Avg. # of infected $\mathcal{B}$ nodes	Avg. # of infected $\mathcal{N}$ nodes
$F \cup T_{N1}$	Louvain	40	30,764	33	11,340	2,005	4,302	27	216
	Infomap	5	246,112	267	18,909	4,486	2,997	177	496
	Label Propagation	287	4,288	5	1,718	353	982	4	53
$F \cup T_{N2}$	Louvain	61	47,556	82	42,135	2,893	18,601	74	603
	Infomap	6	483,488	836	1	1	1	1	1
	Label Propagation	321	9,037	16	2,362	284	1,759	10	47
$F \cup T_{N3}$	Louvain	48	51,030	79	29,521	3,331	10,170	72	574
	Infomap	2,647	925	1	952	68	549	1	105
	Label Propagation	316	7,751	12	1,488	344	929	7	41
$F \cup T_{N4}$	Louvain	41	64,961	33	16,483	1,784	10,743	32	230
	Infomap	1,240	2,148	1	964	67	645	1	99
	Label Propagation	419	6,357	3	2,044	242	1,387	3	54
$F \cup T_{N5}$	Louvain	35	21,636	25	14,600	2,047	4,522	23	219
	Infomap	807	938	1	1,075	105	572	1	148
	Label Propagation	142	5,333	6	2,023	418	1,210	5	51
$F \cup T_{N6}$	Louvain	31	10,574	6	2,278	398	1,333	5	5
	Infomap	217	1,511	1	545	73	284	1	56
	Label Propagation	115	2,850	2	834	150	442	2	31
$F \cup T_{N7}$	Louvain	27	7,188	7	1,976	404	664	6	41
	Infomap	3	64,692	61	6,504	1,590	1,535	52	79
	Label Propagation	78	2,488	2	882	217	359	2	27
$F \cup T_{N8}$	Louvain	14	114,353	15	3,801	371	3,106	5	28
	Infomap	3	533,649	70	9,649	1,620	6,213	33	95
	Label Propagation	123	13,016	2	755	122	465	2	26
$F \cup T_{N9}$	Louvain	40	18,008	14	4,912	964	2,089	10	109
	Infomap	3	240,101	192	397	174	194	17	27
	Label Propagation	192	3,752	3	1,006	237	479	3	39
$F \cup T_{N10}$	Louvain	50	23,956	25	6,125	898	2,915	17	161
	Infomap	1,096	1,093	1	576	51	291	1	98
	Label Propagation	228	5,253	5	1,152	231	709	5	36

Table 2.7: Evaluation of vulnerability of boundary nodes for $DS1$.

	AP@1			**AP@5**			**AP@10**			**AP@15**			**MAP**		
	L	**I**	**LP**	**L**	**I**	**LP**	**L**	**I**	**LP**	**L**	**I**	**LP**	**L**	**I**	**LP**
M1	0.759	0.676	0.712	0.736	0.548	0.519	0.606	0.543	0.533	0.661	0.505	0.566	0.672	0.546	0.555
M2	0.818	0.749	0.907	0.769	0.733	0.799	0.821	0.699	0.999	0.733	0.666	0.999	0.785	0.733	0.875
M3	0.805	0.642	0.878	0.567	0.509	0.749	0.590	0.512	0.674	0.524	0.586	0.833	0.596	0.577	0.751
M4	0.468	0.714	0.750	0.366	0.674	0.633	0.323	0.523	0.659	0.325	0.454	0.799	0.350	0.569	0.660
F1	0.892	0.749	0.855	0.824	0.679	0.999	0.922	0.499	0.799	0.899	0.422	0.999	0.876	0.552	0.905
F2	0.819	0.999	0.874	0.727	0.499	0.839	0.741	0.399	0.924	0.706	0.266	0.999	0.714	0.518	0.900
F3	0.933	0.945	0.933	0.955	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.972	0.985	0.995
F4	0.999	0.999	0.999	0.955	0.999	0.999	0.979	0.999	0.999	0.999	0.999	0.999	0.991	0.999	0.999
T1	0.222	0.531	0.868	0.424	0.492	0.716	0.439	0.349	0.479	0.377	0.344	0.533	0.450	0.424	0.644
T2	0.548	0.374	0.482	0.299	0.399	0.999	0.049	0.299	0.699	0.033	0.033	0.466	0.173	0.264	0.726
T3	0.666	0.470	0.913	0.519	0.499	0.999	0.299	0.499	0.899	0.266	0.433	0.799	0.391	0.479	0.900
T4	0.449	0.464	0.699	0.399	0.000	0.479	0.409	0.000	0.499	0.362	0.000	0.366	0.399	0.106	0.500
M_{avg}	0.712	0.695	0.811	0.609	0.616	0.675	0.585	0.569	0.716	0.560	0.552	0.799	0.600	0.606	0.710
F_{avg}	0.910	0.923	0.915	0.865	0.794	0.959	0.901	0.724	0.930	0.900	0.671	0.999	0.888	0.763	0.949
T_{avg}	0.471	0.459	0.740	0.410	0.347	0.798	0.299	0.286	0.644	0.259	0.202	0.541	0.353	0.318	0.692

Table 2.8: Evaluation of vulnerability of communities in $DS1$.

	$\tau_{\mathbf{M1}}$	$\tau_{\mathbf{M2}}$	$\tau_{\mathbf{M3}}$	$\tau_{\mathbf{M4}}$	$\tau_{\mathbf{F1}}$	$\tau_{\mathbf{F2}}$	$\tau_{\mathbf{F3}}$	$\tau_{\mathbf{F4}}$	$\tau_{\mathbf{T1}}$	$\tau_{\mathbf{T2}}$	$\tau_{\mathbf{T3}}$	$\tau_{\mathbf{T4}}$
L	-0.027	0.003	-0.149	-0.035	0.050	0.164	0.457	0.161	-0.045	-0.255	-0.090	-0.030
I	0.072	0.000	0.274	0.138	0.642	0.667	0.117	0.146	-0.037	-0.222	-0.025	-0.031
LP	0.039	-0.014	0.019	0.018	0.039	0.029	0.381	0.714	0.003	0.005	-0.110	-0.036

Table 2.9: Evaluation of vulnerability of boundary nodes in $DS2$.

	AP@1			AP@5			AP@10			AP@15			MAP		
	L	**I**	**LP**	**L**	**I**	**LP**	**L**	**I**	**LP**	**L**	**I**	**LP**	**L**	**I**	**LP**
F_{N1}	0.729	0.999	0.502	0.866	0.533	0.999	0.785	0.333	0.799	0.644	0.222	0.766	0.78	0.48	0.825
T_{N1}	0.624	0	0.571	0.819	0	0.999	0.766	0	0.999	0.799	0	0.999	0.766	0	0.872
$F \cup T_{N1}$	0.799	0.799	0.577	0.799	0.599	0.899	0.789	0.299	0.899	0.752	0.244	0.999	0.789	0.44	0.897
F_{N2}	0.728	0.999	0.728	0.599	0	0.933	0.735	0	0.899	0.776	0	0.933	0.711	0.133	0.881
T_{N2}	0.702	0.314	0.62	0.616	0	0.499	0.516	0	0.999	0.733	0	0.999	0.565	0.079	0.831
$F \cup T_{N2}$	0.745	0.999	0.669	0.614	0	0.699	0.737	0	0.899	0.747	0	0.933	0.697	0.133	0.806
F_{N3}	0.666	0.49	0.541	0.584	0.799	0.999	0.774	0.899	0.999	0.745	0.733	0.999	0.691	0.808	0.874
T_{N3}	0.733	0.302	0.607	0.949	0.199	0.999	0.799	0.099	0.999	0.933	0.066	0.999	0.916	0.174	0.973
$F \cup T_{N3}$	0.76	0.532	0.555	0.592	0.879	0.999	0.723	0.849	0.999	0.683	0.866	0.999	0.667	0.846	0.885
F_{N4}	0.599	0.999	0.556	0.699	0.299	0.999	0.585	0.099	0.899	0.866	0.066	0.999	0.637	0.282	0.878
T_{N4}	0.622	0.363	0.523	0.516	0	0.699	0.419	0	0.599	0.666	0	0.999	0.531	0.046	0.722
$F \cup T_{N4}$	0.707	0.369	0.579	0.687	0.999	0.799	0.662	0.499	0.966	0.59	0.399	0.866	0.652	0.62	0.786
F_{N5}	0.914	0.711	0.957	0.899	0.199	0.999	0.924	0.099	0.999	0.895	0.133	0.999	0.907	0.228	0.997
T_{N5}	0.599	0.999	0.824	0	0.399	0.399	0	0	0.199	0	0	0.133	0.073	0.279	0.347
$F \cup T_{N5}$	0.857	0.57	0.666	0.89	0.399	0.699	0.957	0.299	0.599	0.893	0.266	0.999	0.911	0.362	0.726
F_{N6}	0.769	0.499	0.762	0.519	0.533	0.999	0.499	0.599	0.899	0.466	0.533	0.666	0.549	0.581	0.867
T_{N6}	0.666	0.562	0.923	0	0	0.599	0.099	0	0	0	0	0	0.08	0.037	0.301
$F \cup T_{N6}$	0.612	0.349	0.565	0.599	0	0.699	0.533	0	0.899	0.866	0	0.599	0.599	0.173	0.733
F_{N7}	0.749	0.499	0.914	0.399	0	0.399	0.099	0	0.199	0.066	0	0.133	0.285	0.033	0.314
T_{N7}	0.649	0.999	0.833	0.633	0.499	0.999	0.749	0.099	0.899	0.533	0.066	0.666	0.688	0.321	0.836
$F \cup T_{N7}$	0.481	0.999	0.538	0.519	0.733	0.999	0.433	0.749	0.899	0.355	0.533	0.666	0.467	0.74	0.809
F_{N8}	0.705	0.999	0.724	0.533	0.733	0	0.499	0.566	0	0.333	0.199	0	0.517	0.592	0.097
T_{N8}	0.499	0.499	0.721	0.499	0.499	0	0	0.349	0	0	0	0	0.218	0.352	0.048
$F \cup T_{N8}$	0.499	0.999	0.442	0.733	0.933	0.199	0.499	0.599	0.099	0.333	0.422	0.066	0.546	0.713	0.2
F_{N9}	0.72	0.377	0.849	0.672	0	0.999	0.599	0	0.999	0.483	0	0.933	0.582	0.048	0.952
T_{N9}	0.631	0.666	0.558	0.499	0.199	0.199	0.249	0.099	0.099	0.333	0.066	0	0.409	0.215	0.133
$F \cup T_{N9}$	0.724	0.333	0.526	0.619	0.199	0.899	0.539	0.099	0.699	0.516	0.066	0.999	0.568	0.154	0.817
F_{N10}	0.773	0.475	0.912	0.576	0.666	0.899	0.622	0.399	0.999	0.552	0.366	0.999	0.605	0.474	0.97
T_{N10}	0.999	0.31	0.599	0.199	0	0	0.199	0	0	0.133	0	0	0.253	0.02	0.039
$F \cup T_{N10}$	0.759	0.332	0.657	0.599	0.599	0.899	0.614	0.349	0.999	0.599	0.266	0.999	0.627	0.389	0.937
F_{avg}	0.735	0.705	0.744	0.635	0.376	0.822	0.612	0.299	0.769	0.583	0.225	0.742	0.626	0.366	0.766
T_{avg}	0.672	0.501	0.678	0.473	0.179	0.539	0.379	0.065	0.479	0.413	0.013	0.479	0.449	0.152	0.51
$F \cup T_{avg}$	0.694	0.628	0.577	0.665	0.534	0.779	0.649	0.374	0.796	0.633	0.306	0.813	0.652	0.457	0.759

Table 2.10: Evaluation of vulnerability of communities in $DS2$.

	τ_F			τ_T			$\tau_{F \cup T}$		
	L	**I**	**LP**	**L**	**I**	**LP**	**L**	**I**	**LP**
N1	0.009	0.333	0.075	0.171	1	-0.008	0.128	0	0.027
N2	0.03	0.999	0.044	-0.063	0.015	-0.03	0.066	0.999	0.003
N3	-0.351	-0.001	0.012	0.2	-0.04	0.06	-0.411	-0.012	0.044
N4	0.078	-1	-0.009	-0.222	0.065	-0.011	-0.051	-0.007	-0.022
N5	-0.073	0.003	0.075	-0.238	-1	0.01	-0.055	0.02	0.051
N6	-0.113	1	0.039	-0.055	0.017	0.052	-0.092	0.033	-0.038
N7	-0.284	0.052	0.109	0.157	-1	-0.062	-0.065	0.333	0.066
N8	-0.088	0.333	-0.035	-0.333	-1	-0.089	-0.076	-0.333	0.011
N9	0.08	0.007	0.006	-0.147	0.333	-0.018	0.076	0.333	0.047
N10	-0.019	0.017	0.067	-0.399	-0.022	-0.027	-0.025	-0.028	0.001

Chapter 3

Identification of Infected Population

3.1 Introduction

In recent years, social media platforms have been increasingly witnessed as an emerging sphere for generating and spreading false or unverified information. For example, the news about Facebook CEO Mark Zuckerberg offering money to Facebook users who do not share social media hoaxes is itself a parody of social media hoaxes[1]. False rumors can be potentially detrimental, triggering serious repercussions or consequence in our society. A rumor circulating on Facebook and Twitter since December 5, 2015 claimed that Muslim residents of Dearborn, Michigan, held a pro-ISIS march, where protesters were carrying ISIS flags[2]. This rumor was circulated following the mass shooting in San Bernardino, California, by a U.S.-born Muslim who became radicalized while living in the U.S. and whose wife was from Pakistan. On a daily basis, such misinformation originates and propagates within social media outlets, rendering the quality and credibility of social media content seriously inferior.

Social psychology literature defines rumor as a story or a statement whose truth value is unverified or deliberately false [77]. Differentiating rumor from fact, or measuring the truthfulness of information directly is technically very challenging. While one

[1] `https://www.snopes.com/zuckerberg-dont-share-hoaxes/`

[2] `http://www.factcheck.org/2015/12/dearborns-anti-isis-rally/`

way to address this is by debunking the false information using rumor detection and classification approach [27, 78, 25, 60, 20, 79, 56, 21, 80], another way is by estimating whether the spreader of concerned information is trusted or not and to what extent by their peers so as to identify the “high-risk” users who are more likely to spread false information online. These high-risk users are effectively labeled so that other online users can be drawn appropriate attention or be altered against the credibility of information posted by the labeled users. To the best of our knowledge, there is no existing approach that has fallen into this second category based on computational trust for rumor spreader detection in social media sphere except ours [52], where we have conducted a pilot study by utilizing the retweet network of Twitter users. This chapter is a natural extension of the original idea by taking into account other types of trust proxies such as reply relationship among the users as well as conducting more thorough experimentations and analyses.

The concept of *Trust* in Twitter’s retweet network can be described generally as follows: a user which is referred to as **A**, who receives a post tweeted from user **B**, may intend to share the post with his/her followers with the action of propagating the information, i.e., *retweet*. **A** might also decide to *reply* the tweeter, which can contain some additional information or comment in the reply beyond a mere retweet. There are two essential factors that can influence the decision of user **A**, who may choose to act on an original post or not: 1) The *trustworthiness* of user **B**, i.e., the willingness of the network to trust **B**; and 2) the *trustingness* of **A**, i.e., the propensity of **A** to trust the other users in the network. According to the prior research of Computational Trust such as [45, 50], Trustingness and Trustworthiness are characterized as a pair of complementary measures of user trust in social network and both of them are associated with each network user. A person having higher trustingness contributes to the trustworthiness of its neighbors to a lower degree, while a higher trustworthiness is a result of lots of neighbors linked to the actor having lower trustingness.

Intuitively, users with high trustingness are more likely to spread information online than those with low trustingness since they are more likely to believe what someone else tweets. When the circulated message is false, such users tend to be more likely to become rumor spreaders. On the other hand, users with high trustworthiness are generally less likely to inject or spread false information than those who have low trustworthiness in the sense that the tweets of users who have high trustworthiness are historically retweeted more extensively and they are subjectively more cautious on what they tweet for maintaining their own reputation. As a result, the properties of users in terms of information veracity they are involved in propagating can be inferred somehow based on the nuance of trust relationships among the users.

In this chapter, we propose a novel approach for the identification of rumor spreaders based on the concept of *Believability*, which is a measure defined on the basis of trustingness and trustworthiness metrics. Specifically, Believability represents the strength of a directed edge between the tweeter **B** and the responder **A**, indicating how strong the potential is for the information from **B** to be spread through **A**. The basic idea is that the Believability of the retweeted message is proportional to the trustingness of the responder **A** and the trustworthiness of the tweeter **B**. To this end, we construct the trust network among users, using retweets, and additionally replies, as proxies of trust relationship, for automatically learning the user representation as embeddings in a low-dimension space. More specifically, the representation is inferred from the re-weighted user network with the believability on its edges by employing a state-of-the-art network embedding algorithm called LINE [81]. Finally, based on the generated user embeddings, we apply supervised machine learning algorithms such as neural networks or other kind of classifiers to categorize the given user spreading the specific information as a rumor spreader or not.

In a nutshell, the contributions of the chapter are three-fold:

- To the best of our knowledge, this is the first attempt to identify rumor spreaders on Twitter by exploring the nuance of concepts in Computational Trust, i.e., trustingness and trustworthiness, for creating a novel measure of believability which quantifies the potential of a message being spread from one user to the others.

- We propose a novel technical framework that strengthens the representation of user properties in consideration of information veracity using network feature learning based on a large-scale believability re-weighted trust network. Experimental results demonstrate the superiority of the proposed method over technically more straightforward approaches.

- We build three Twitter datasets using different trust proxies (i.e., retweet-only, reply-only, retweet+reply) based on a set of real-world rumorous and non-rumorous events gathered from rumor debunking websites, which are made publicly available to research community[3].

3.2 Related Work

The task of rumor detection can be classified into two categories: rumor information detection and rumor spreader detection. Most of prior research focused on rumor information detection. Little work has been done, however, for rumor spreader detection.

Automatic detection of rumorous information from social media is based on traditional classifiers stemming from the pioneering study of information credibility on Twitter [27]. In the subsequent studies [56, 78, 25, 60, 20, 21, 80], different sets of hand-crafted features were proposed and incorporated to determine whether a claim about some event is credible or not. However, feature engineering in these methods is painstakingly labor intensive. Ma et al. [60] proposed a RNN-based method that automatically learns the representations to capture the hidden implications and dependencies of complex signals over time, and achieved better performance due to the effective representation learning capacity of deep neural models. In addition, other neural models such as Convolutional Neural Networks (CNN) and tree-structure Recursive Neural Networks (RvNN) have also been attempted by exploiting either social media content or propagation structures of information in the recent studies [82, 22]. A comprehensive survey focusing on rumor information detection can be found in [43], and two others focusing on detection on fake news and false information in general can be found in [36, 44]. In our work, we focus on the rumor spreader detection instead of rumor information detection. To the best of our knowledge, there is no concrete study conducted so far for identifying rumor spreaders via predictive analytics except for a few other works considering spreader characteristics as features for rumor information detection [83, 25].

Computational trust has been studied extensively in recent years. Many researchers have tried to assign trust scores [46, 47, 48] to the nodes in a network to accomplish

[3] `https://github.com/BhavtoshRath/RNN-Trust/blob/master/data/snam2018.zip`

different type of tasks. Trust scores can be defined as scores that an algorithm puts on a node in a trust network based on various structural aspects of the node. Eigentrust [47] proposes to rate trust scores of peers in a P2P network. These scores can help an ordinary user in the network to identify the trustworthy peers and initiate content download from them. Eigentrust, like Pagerank [49] calculates a single score for each node in the network. In this algorithm, however, one's reputation does not play a part in the weight of the node's trust vote. Other researchers have proposed measures to rank bias and deserve of a node in a network [48], in which they used an iterative matrix algorithm to calculate bias and deserve of nodes which reinforce each other.

Roy [45, 50] proposed a pair of complementary measures that can be used to measure trust scores of actors in a social network using involvement of social networks. Based on the proposed measures, an iterative matrix convergence algorithm based on HITS [84] was developed that calculates the trustingness and trustworthiness of each actor in the network. The algorithm runs in $O(k \times |E|)$ time where k denotes the number of iterations and $|E|$ denotes the number of edges in the network. In this chapter, we propose a novel measure called *believability* based upon these two complementary measures for assessing the potential of the message being spread from one user to the other, which is used to re-weight the edges of the user trust network. Note that the believability is in essence different from commonly known concept of credibility studied in many papers [27, 85, 86, 87], where credibility is primarily used to measure the quality of content being believed in or that of a user being trusted, but believability here is a measure of "spreadability" of information between *a pair of users* instead of an individual user.

Network-based representation learning is an emerging area in machine learning. DeepWalk [88] learns node embeddings by exploring local neighborhood of the nodes using truncated random walks. Since the strategy of the random walk is uniform following Depth-First-Search (DFS) style, it gives no control over the explored neighborhoods. Also, it works only for unweighted, undirected graphs. LINE model [81] proposes a breadth-first strategy to explore neighborhoods. Specifically, it learns a feature representation in two separate phases: first, it learns half of the dimensions by Breadth-First-Search (BFS) style simulations over immediate neighbors of nodes, then it learns the other half of dimensions by sampling nodes strictly at a 2-hop distance from the source nodes. This model works for all types of graphs. Node2vec [89] explores diverse network neighborhoods which designs a sampling strategy that smoothly interpolates between BFS and DFS. The assumption is that BFS and DFS are extreme sampling paradigms suited for structural equivalence (i.e., nodes sharing similar roles) and homophily (i.e., nodes from the sample network community), respectively. Node2vec's sampling strategy accommodates for the fact that these notions of equivalence are not competing or exclusive, and real-world networks commonly exhibit a mixture of both. Considering the weighted, directed nature of our network and the complexity of the learning algorithm, in this chapter, we employ LINE algorithm with the 2-hop distance for generating user embeddings from the trust network, where the edges are re-weighted by the believability scores.

3.3 Believability

Trustingness and trustworthiness, from different perspectives, are used to measure the level of trust of each individual user. But they do not quantify the strength of *inter-user* trust, i.e., the trust between two specific users who have retweet or reply relationship. The intensity of inter-user trust is very important to indicate the potential or capacity of user network edges for transmitting messages. When a message is propagated, the strength of inter-user trust along the propagation path would largely determine how fast and how far the message could be transmitted over the network. In the original trust model in Figure 1.3, edges weighed by the frequency of retweets between two users cannot reflect such kind of "spreadability" of network edges. In this regard, a new method of re-weighting the edges is very much desirable.

We propose the new concept called *Believability*, a quantitative figure that is computed for a directed edge between two nodes used to measure the potential of messages being transmitted through the edge based on the strength of belief between two neighbors on that edge. In the context of retweet, a directed edge from **B** to **A** exists if a tweet of **A** is retweeted by **B**. The believability quantifies the strength that **B** trusts on **A** when **B** decides to retweet **A**. Therefore, **B** is more likely to believe in **A** if:

I. **A** has a high trustworthiness score, i.e., **A** is highly likely to be trusted by other users in the network, or

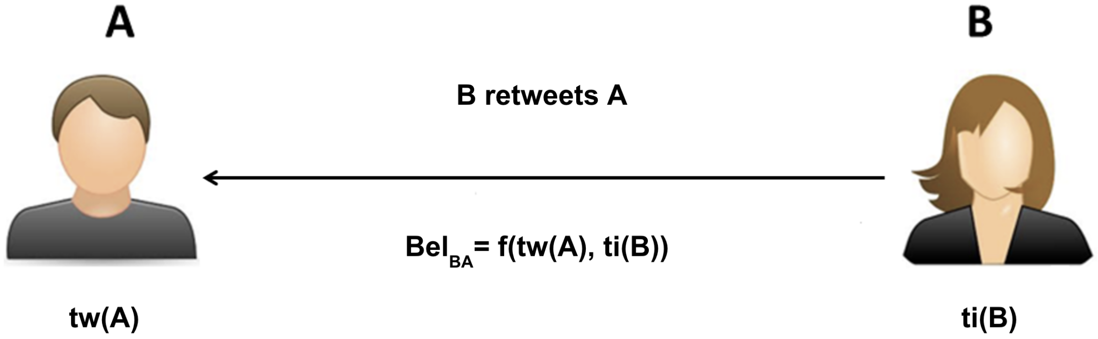


Figure 3.1: An illustration of Believability.

II. **B** has a high trustingness score, i.e., **B** is highly likely to trust others.

The same applies to the case of reply. So, the believability score is supposed to be proportional to the two values above, which can be jointly determined and computed as follow:

$$Believability(B \rightarrow A) = tw(A) * ti(B) \tag{3.1}$$

Figure 3.1 illustrates the relationship between the believability and the trust measures given a retweet or reply edge in the user network. The believability score will be used to re-weight the edges so that the representation of users can be reasonably learned with the differentiation of variable spreadability of different edges. The key reason why this can result in better user representation learning is that the inter-user believability score will lead to the random walk being biased to favorably travel towards nodes via high believability edges (see Section 3.4), thus potentially maximizing the transmission capacity of information over the network.

3.4 User Representation Learning

In this section, we will discuss how to automatically represent the users based on the re-weighted network using believability scores as the edge weights.

3.4.1 Rumorous users and context

We define the network as $G = (V, E)$, where $V = \{u_1, u_2, \ldots, u_n\}$ refers to a set of nodes each representing a user, and $E = \{w_{ij}\}$ is a set of directed edges corresponding to the relationship (retweet or reply) among the nodes in V, which are weighted by the believability scores.

Figure 3.2 illustrates the contexts of network where similar users should be represented closely to each other in the embedding space. Without loss of generality, we illustrate three basic cases of context where two users u and u' reside, which should be considered similar users, and how their similarity is related to rumor propagation:

- **a)** u and u' act mutually as the context of one another and the u-u' weight is strong, suggesting that u may be a "hard-core fan" of u'. If u' is a frequent rumor spreader, so potentially very likely is u because of the generally low veracity of information provided by u'; and conversely, u' is more likely to be rumorous if u is often rumorous since most of the information u spreading is coming from u'.

- **b)** u and u' share many common neighbors (like u_1, u_2, u_3) with in-links, implying that they have a large overlapping group of fans who trust them. If u often pollutes his fans with hearsays while still not losing its audiences, it is very likely that u' is similar to u in terms of information spreading behaviors, because otherwise they could not own many common followers.

- **c)** Similar but different as (b), u and u' share many neighbors with out-links, indicating that both of them trust in a common group of sources of messages. If u is a frequent receiver of rumors, it is reasonable to infer that u' is inclined to be similar as u because of substantial overlap of their information source or context.

As such, by considering the commonality of context, similar users will be projected closely in the representation space for yielding better classification effectiveness.

3.4.2 User embeddings

We adopt the second-order proximity between a pair of nodes in a network-based representation learning method [81] which is called LINE, to learn user embeddings based on the retweet and reply user network depicted above. The goal is to embed each user

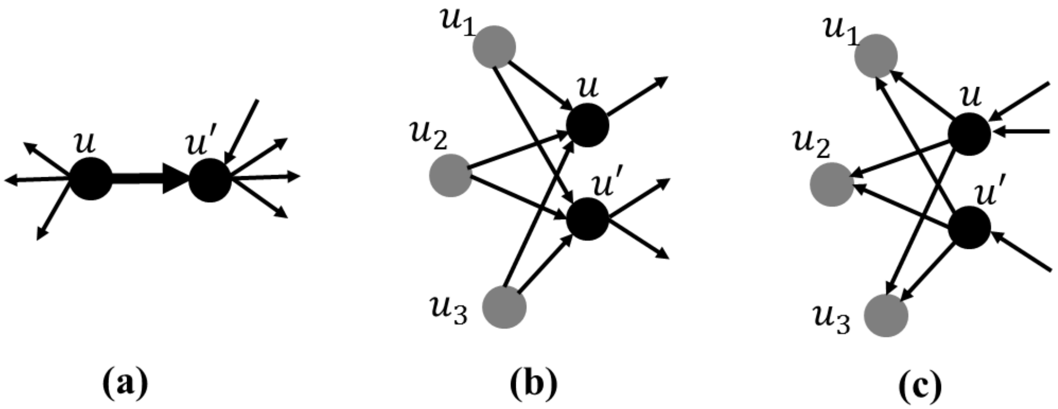


Figure 3.2: An illustration of similar users in a network.

$u_i \in V$ into a lower-dimensional space $\mathbb{R}^d$ by learning a function $f_G : V \rightarrow \mathbb{R}^d$, where d is the dimension of the projected vector. Specifically, for each u_i, let $\overline{v}_i$ denote the embedding of u_i as a node and $\overline{v}'_i$ be the representation of u_i when treated as a specific context of other nodes. For each edge $u_i \rightarrow u_j$, the conditional probability of u_j being generated by u_i as context is defined as follow:

$$p(u_j|u_i) = \frac{exp(\overline{v}'^{\intercal}_j \cdot \overline{v}_i)}{\sum_{k=1}^{|V|} exp(\overline{v}'^{\intercal}_k \cdot \overline{v}_i)} \tag{3.2}$$

Given this definition, the nodes sharing similar contexts will have similar conditional distributions over the entire set of nodes. To preserve the context proximity, the objective is to make $p(u_j|u_i)$ be close to its empirical distribution $\hat{p}(u_j|u_i)$, where the empirical distribution can be observed from the weighted social context network. Thus, the objective function is defined as:

$$\min \sum_{(i,j)\in E} \lambda_i * d\big(\hat{p}(u_j|u_i), p(u_j|u_i)\big) \tag{3.3}$$

where $d(\cdot, \cdot)$ is the distance between two probabilities based on KL-Divergence, λ_i is the prestige of u_i which is set to u_i's out-degree d_i following [81], and the empirical distribution is computed as $\hat{p}(u_j|u_i) = w_{ij}/d_i$ where w_{ij} is the weight of the edge (i, j).

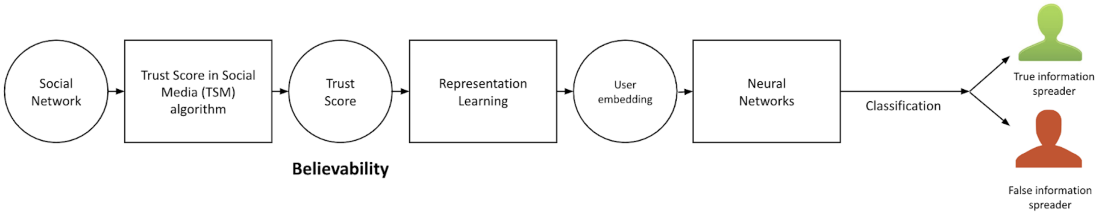


Figure 3.3: Spreader identification framework.

In the learning, we use LINE for optimizing equation 3.3, which provides an efficient solution based on negative sampling of edges and asynchronous stochastic gradient descent over the mini-batches of the sampled edges for parameter update.

3.5 Identifying Spreaders of Rumors

The overall framework of the spreader identification model is summarized in Figure 3.3. To identify the rumor spreaders out of a large number users, we use Recurrent Neural Network (RNN) to model the classification process. RNN is used as the classification model for two reasons: Firstly, our data is based on time sequence, i.e. retweets/replies are sequential in nature, where RNN is naturally suitable for the structure of data. Secondly, the training data is of variable length, i.e., the source tweets can have different number of retweets/replies, for which RNN also fits. It is important to note that there is no fixed time interval between two successive actions. Therefore, we can safely consider that the data is a time sequence instead of time series.

In our experiments (see Section 3.6), we adopt two variants of RNN model: Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) model in order to leverage long-distance dependencies of units in the sequence.

3.5.1 RNN-based user models

An RNN is a type of feed-forward neural network that can be used to model variable-length sequential information. A basic RNN is formalized as follows: given an input sequence $(x_1, \ldots, x_T)$, for each time step, the model updates the hidden states $(h_1, \ldots, h_T)$

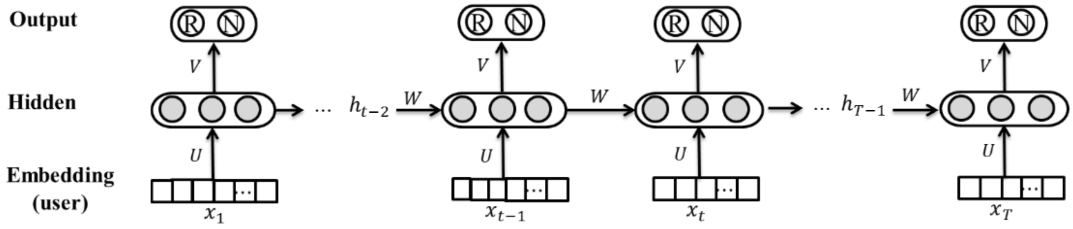


Figure 3.4: Proposed recurrent neural network based spreader classification model.

and generates the output vector $(o_1, \ldots, o_T)$, where T depends on the length of the input. From $t = 1$ to T, the algorithm iterates over the following equations:

$$\begin{aligned} h_t &= tanh(Ux_t + Wh_{t-1} + b) \\ o_t &= Vh_t + c \end{aligned} \tag{3.4}$$

where U, W and V are the input-to-hidden, hidden-to-hidden and hidden-to-output weight matrices, respectively, b and c are the bias vectors, and $tanh(.)$ is a hyperbolic tangent nonlinearity function.

Typically, the gradients of RNNs are computed via back-propagation through time [90]. In practice, because of the vanishing or exploding gradients [91], the basic RNN cannot learn long-distance temporal dependencies with gradient-based optimization. One way to deal with this is to make an extension that includes “memory” units to store information over long time periods, commonly known as Long Short-Term Memory (LSTM) unit [92, 93]. Gated Recurrent Unit (GRU) [94] is another simpler RNN-model.

LSTM networks were designed to address the vanishing gradients through a gating mechanism. They are basically an alternative way of computing the hidden state.

LSTMs use the following equations to compute the hidden state [92, 93]:

$$
\begin{aligned}
i_t &= \sigma(x_t\,U_i + h_{t-1}\,W_i)\\
f_t &= \sigma(x_t\,U_f + h_{t-1}\,W_f)\\
o_t &= \sigma(x_t\,U_o + h_{t-1}\,W_o)\\
g_t &= tanh(x_t\,U_g + h_{t-1}\,W_g)\\
c_t &= c_{t-1} \cdot f_t + g_t \cdot i_t\\
h_t &= tanh(c_t) \cdot o_t
\end{aligned}
$$

For a basic RNN model, the inputs to the unit were x_t (the current input at time step t) and h_{t-1} (the hidden state in previous time step), and the output was a new hidden state h_t and the output state o_t at current time step. In LSTM, however, the hidden state is computed based on the states of some internal gates using the above equations, which are explained as follows: i_t, f_t, o_t are the *input*, *forget* and *output* gates computed for time step t. They have the same equations but different parameter matrices. They are called *gates* because the sigmoid function σ converts these into vectors in range between 0 and 1. Multiplying these gates element-wise lets us decide how much of the other vector is let through to the next hidden unit. g_t is a *candidate* hidden state that is computed based on the current input and the previous hidden state. The internal memory c_t is computed as previous memory c_{t-1} multiplied by the $forget$ gate, and the newly computed hidden state g, multiplied by the *input* gate. This mechanism allows LSTM to not ignore the old memory completely. We finally compute the output hidden state h_t by multiplying the *hyperbolic tan* of internal memory with the output gate.

The model of GRU is very similar to that of LSTM layer however more simplified. A GRU has gating units that modulate the flow of the content inside the unit, but a GRU is simpler than LSTM with fewer parameters. The following equations are used for a GRU unit in hidden layer [94]:

$$
\begin{aligned}
z_t &= \sigma(x_t\,U_z + h_{t-1}\,W_z)\\
r_t &= \sigma(x_t\,U_r + h_{t-1}\,W_r)\\
\tilde{h}_t &= tanh(x_t\,U_h + (h_{t-1} \cdot r_t)\,W_h)\\
h_t &= (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t
\end{aligned}
$$

where a reset gate r_t determines how to combine the new input with the previous memory, and an update gate z_t defines how much of the previous memory is cascaded into the current time step, and $\tilde{h}_t$ denotes the candidate activation of the hidden state h_t.

The major difference between LSTM and GRU can be summarized as follows: 1) While an LSTM unit consists of three gates, GRU unit consists of two gates; 2) LSTM unit have an internal memory unit while the GRU unit does not; 3) It is easier to train GRUs than LSTMs as they have fewer parameters. On large datasets however, LSTMs tend to give better results.

We use the recurrent units of LSTM and GRU to fit the time steps as the basic identification framework. For each source tweet, all of its retweeting or replying users are ordered in terms of the time stamps that indicate when the different users retweet or reply it. In each time step, we input the embedding of the user who retweets or replies the message at that time step. Suppose that the dimensionality of the generated user embedding is K. The structure of our RNN model is illustrated in Figure 3.4. Note that an output unit is associated with each of the time steps, which uses *sigmoid* function for the probabilistic output of the two classes indicating the input user is a rumor spreading user or not.

Let g_c, where c denotes the class label, be the ground-truth 2-dimensional multinomial distribution of a user. Here, the distribution is of the form $[1, 0]$ for rumor spreading users and $[0, 1]$ for non-rumor spreading users. For each training instance (i.e., each source tweet), our goal is to minimize the squared error between the probability distributions of the prediction and ground truth:

$$min \quad \sum_c (g_c - p_c)^2 + \sum_i ||\theta_i||^2$$

where g_c and p_c are the gold distribution and predicted distribution, respectively, θ_i represents the model parameters to be estimated, and the L2-regularization penalty is used for trading off the error and the scale of the problem.

3.5.2 Naive User models

Instead of using a RNN-based user model presented in Section 3.5.1, one may come up with some naive and more straightforward models based upon the property of trust.

Trustingness-only model

Intuitively, users with high trustingness, who easily trust others, are more likely to spread rumors. Our trustingness-only model simply learns a threshold based on the correlation between the trustingness score and ground truth of users in the training data. With the threshold, the model can easily predict user class given the trustingness of a new user. The model is described as follows:

$$prediction(u) = \begin{cases} \textbf{true} & \text{if } trustingness(u) \geq \mathcal{T}_{ti}; \\ \textbf{false} & \text{otherwise} \end{cases} \tag{3.5}$$

where $\mathcal{T}_{ti}$ is the threshold of trustingness score to be learned from training.

Trustworthiness-only model

In contrast, the users with high trustworthiness who are more trustworthy are less likely to spread rumors. The trustworthiness-only model similarly learns a threshold from the training data capturing the relationship between the trustworthiness score and ground truth label of users. Similar to Eq. 3.5, the trustworthiness-only model is given as below:

$$prediction(u) = \begin{cases} \textbf{false} & \text{if } trustworthiness(u) \geq \mathcal{T}_{tw}; \\ \textbf{true} & \text{otherwise} \end{cases} \tag{3.6}$$

where $\mathcal{T}_{tw}$ is the threshold of trustworthiness score to be learned from training data.

Interpolation model

The interpolation model linearly combines the trustingness and trustworthiness scores in such a way that they are interpolated with the appropriate weights to give an optimal prediction on its trust score. The trust score of a given user can be predicted as:

$$T(u) = \alpha * trustingness(u) + (1 - \alpha) * trustworthiness(u)$$

where α is the weight that can be fixed during training stage. With the similar thresholding strategy above, we can obtain the threshold $\mathcal{T}_{tr}$ of the interpolated trust score, and the class of user can be predicted as follows:

$$prediction(u) = \begin{cases} \textbf{true} & \text{if } T(u) \geq \mathcal{T}_{tr}; \\ \textbf{false} & \text{otherwise} \end{cases} \tag{3.7}$$

Logistic regression classifier with user embeddings

For machine learning based approach, in addition to use RNN-based model, we can also utilize other type of classifiers such as logistic regression or support vector machines for classifying the users. The user embeddings generated from the LINE algorithm can be straightforwardly treated as features of the classifiers.

Here we use the L2-regularized logistic regression (L2_LR) classifier [95]. The L2_LR solves the following unconstrained optimization problem:

$$\min_{w} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \log(1 + \exp(-y_i w^T x_i)) \tag{3.8}$$

where x_i and y_i are the input vector of an instance and its prediction, w is the vector of model parameters to learn. The reasons why we have chosen L2_LR are that (a) it has consistently delivered state-of-the-art performance in several applications [96, 97], and is thus a strong contender, and (b) logistic regression, like RNN, natively outputs posterior probabilities [98], which is comparable.

3.6 Experiments and Results

In this section, we will describe the collection of datasets, comparative experiments and the results achieved.

3.6.1 Data collection

We constructed our datasets based on two reference datasets, namely Twitter15 [78] and Twitter16 [60], which were previously used for binary classification of rumor and non-rumor with respect to a given event that contains its relevant tweets. The two Twitter datasets were originally constructed by first gathering a set of rumorous and non-rumorous events from rumor debunking websites such as `www.snopes.com`. For every event on the debunking websites, there is a main claim about the specific event which is associated with it. Then they gathered the relevant tweets of each event via keyword search on Twitter's site, for which a set of keywords were manually composed based upon the claim of each event [78, 60].

In our work, given the main claim of each event in the two datasets, we extracted from them the popular source tweets that are highly retweeted or replied. We then constructed the propagation threads (i.e., retweets and replies) for these source tweets. Because Twitter API cannot retrieve the retweets and replies, we gathered the retweet users for a given tweet from Twrench[4]. We also crawled the replies to the source tweets through Twitter's web interface. Note that each main claim is associated with a gold-standard veracity label, i.e., rumor or non-rumor, derived directly from the rumorous and non-rumorous events on the rumor debunking websites. The label of the claim associated with event will be used to generate the ground truth of users for rumor spreader detection (see Section 3.6.1).

User trust network

Based on the users appearing in these events, we constructed three networks (i.e., retweet-only, reply-only and retweet+reply) using the following steps:

I. We merged the two reference datasets into one large corpus;

II. We obtained the follow relationships among the users that have appeared across all the events for getting an initial user network[5]. In particular, we treat the follow relationship on Twitter as a special yet rudimentary form of retweet with

[4] `https://twren.ch`

[5] We used Twitter API for getting maximum 5k friends of each user, and obtained more friends by requests via Twitter's Web interface.

Table 3.1: Retweet and reply based trust network dataset statistics.

	Retweet	Reply	Combined
Total # of nodes	1,292,708	1,122,797	1,321,872
Total # of edges	19,533,330	18,535,341	19,645,380
Avg in-degree	15.1	16.5	14.9
Max in-degree	95,303	95,302	95,303
Min in-degree	0	0	0
Avg out-degree	15.1	16.5	14.9
Max out-degree	52,519	6,740	58,274
Min out-degree	0	0	0

frequency of 1 (i.e., a follow relationship is counted as one-time retweet) in order to alleviate the sparsity of the generated retweet network.

III. From each of the events, we extracted popular source tweets with more than 50 retweets and replies altogether[6] that are highly responded;

IV. We collected all the retweet users for each source status[7]. These retweet relationships are added as edges into the initial users network above to form the retweet-only network.

V. We then collected all the reply users for each source status[8]. These reply relationships are added as edges into the initial user network to form the reply-only network.

VI. We finally generate the retweet+reply network by joining the two networks obtained above.

The statistics for the three networks are shown in Tables 3.1. These three user networks will be used to compute the user trust scores for then generating the user embeddings (see Section 3.4).

[6] Though unpopular tweets could be fake, we ignore them as they do not draw much attention and are hardly impactful

[7] Since Twitter API cannot retrieve over 100 retweets, we gathered the retweet users for a given tweet from Twrench (`https://twren.ch`)

[8] We generated the replies of the source tweets using PHEME toolkit (`https://github.com/azubiaga/pheme-twitter-conversation-collection`)

Table 3.2: User classification dataset statistics.

	Retweet	Reply	Combined
# of unique users	902,806	98,373	969,857
# of users spreading rumors	417,569	49,671	415,846
# of users not spreading rumors	485,237	48,702	554,011
Total # of source tweets	3,098	3,068	3,098
# of rumor source tweets	1,716	1,690	1,716
# of non-rumor source tweets	1,382	1,378	1,382
Avg # length of source tweet sequences	413.98	52.48	464.9
Max # length of source tweet sequences	2,915	814	3,145
Min # length of source tweet sequences	49	2	56

User classification dataset

We also built user classification dataset for our RNN models (see Section 3.5.1) based on the source tweets, where each source tweet is associated with a sequence of retweeters/repliers ordered by the time stamp when the retweet/reply occurs.

The ground-truth label for each user is determined by the nature of the source tweet by following these rules:

I. If the main claim of event is reported as rumorous and the source tweet supports it, the ground truth of the users participating in spreading it are labeled as rumor spreaders; if the source tweet denies the claim, the users participating in spreading it are labeled as non-rumor spreaders.

II. If the main claim is reported as a non-rumor, the ground truth labels of the users responding to the source tweet are assigned in an opposite way as the rumor case above depending on the specific stance of the source tweet that the users are participating in spreading.

III. When a user appear both in rumor and non-rumor cases, we check the frequency that the user has appeared in the two cases: if it appears more often in rumor than in non-rumor cases, it is labeled as a rumor spreader; otherwise labeled as a non-rumor spreader.

According to these rules, the procedure of ground truth assignment was carried out semi-automatically, in which only the stance of source tweets need to be determined manually.

The statistics on the user classification datasets are shown in Tables 3.2. As we can observe, there are few major distinctions between the retweet and reply dataset generated, which may contribute to the differences in the experimental results reported in later section. These differences are described as follows:

- The reply dataset is much smaller than the retweet dataset (both in terms of network size and average sequence length). This is due to the fact that the most popular interactions among users on Twitter are retweet rather than reply, which is reflected properly in our data.
- While each sequence in the retweet dataset has different users, this need not be the case with the reply dataset, as few users in the dataset can participate in the conversation chain via reply multiple times.
- Reply and retweet are both indicators of active engagement with information exchange. We believe that retweet is however a stronger indication of trust than reply. This is because: 1) Through a retweet, a user basically causes information broadcasting to all its followers, and in contrast, a reply is not broadcasted, and remains confined to the conversation chain of the source tweet. 2) A retweet contains nothing but the original post, implying a supportive stance on source tweet. A reply, however, can contain additional text expressing a different attitude toward the source message.

3.6.2 Settings and protocols

We ran TSM to get the trust scores based on our networks, which is then re-weighted by the believability scores. We adopted the generic setting of TSM involvement parameter $s = 0.391$ by referring to [45]. Then, we learned the user embeddings in the networks by running LINE algorithm, where we empirically set the size of embeddings as 200 and kept other parameters as the default settings.

For user classification, we fed the sequence of users of each source tweet into RNN's input layer one at a time and trained the RNN model by employing the derivative of the loss with respect to all the parameters via back propagation [99]. We used gradient descent for parameter update. The size of the hidden units is set as 100 and the learning rate as 0.005, and the number of epoch as 100 for ensuring the convergence of RNN. In prediction, the probabilities of the same users if they appear across different source tweets are averaged for predicting the final class labels.

We made systematic comparisons among the following eight models:

- **Trustingness**: The trustingness-only user model (section 3.5.2);
- **Trustworthiness**: The trustworthiness-only user model (section 3.5.2);
- **Interpolation**: The model that interpolates the trustingness-only and trustworthiness-only models (section 3.5.2);
- **L2_LR**: The L2-regularized logistic regression classifier using user embeddings as features (section 3.5.2);
- **GRU_noweight**: The GRU-based RNN user model using user embeddings as features which were obtained from the unweighted user network whose edge weights are all set equal to 1;
- **GRU_notrust**: The GRU-based RNN user model using user embeddings as features which were obtained from the initial user network (with only follow relationship) without considering other trust relationship;
- **GRU_trust**: The GRU-based RNN user model using user embeddings as features which were obtained from the user network whose edges are re-weighted with believability scores.
- **LSTM_trust**: The LSTM-based RNN user model using user embeddings as features which were obtained from the user network whose edges are re-weighted with believability scores.

For evaluation purpose, we ran experiments based on 10-fold cross validation and used five common evaluation metrics: Accuracy, Area Under the Receiver Operating Characteristic Curve (AUROC), Precision, Recall and F1 measure. The Accuracy is defined over the two classes as: $Accuracy = \frac{\text{\# of correctly predicted users}}{\text{Total \# of users}}$. AUROC is the area under the graphical plot which combines True Positive rate and False Positive rate into one metric and provides a measure of performance (between 0 and 1) across all possible classification thresholds. In all our models we have used the classification threshold of 0.5. The rest of the three metrics are defined for each individual class: For the positive class, i.e., rumor spreading users, the Precision is defined as $Precision(+) = \frac{FP}{TP+FP}$, the Recall is defined as $Recall(+) = \frac{TP}{TP+FN}$, and F_1 is defined as $F_1 = \frac{2*Precision*Recall}{Precision+Recall}$, where TP, FP and FN are true positive rate, false positive rate and false negative rate, respectively. The corresponding metrics for the negative class, i.e., non-rumor spreading users, are defined similarly.

3.7 Analysis of Results

We now provide a detailed description and analysis of the experimental results for the three datasets in the following three sub-sections.

3.7.1 Retweet-only dataset

Table 3.3 shows experimental results run on the retweet-only dataset. It is observed that **Trustingness** model has an accuracy of 0.538. **Trustworthiness** model shows a marginal increase in accuracy 0.56%. AUROC for **Trustworthiness** model also shows a minor increase of 5.2% over **Trustingness** model. While the precision of rumor class noticeably drops by 25.7%, the recall of the same class increases sharply by 800%. Other metric values remain almost identical. This is because the trustworthiness scores are much better scaled than the trustingness scores. The network is very sparse with a large number of star-shaped sub-graphs. This means that a large portion of the nodes have no out-going edges, which results in lots of identical trustingness scores. However, the two models are basically comparable, which is attributed to the fact that the two scores are complementary measures derived from a global user interaction network which are essentially the reciprocal sides of trust. So, overall they contribute equally to the spreader detection.

The **Interpolation** model which combines the above two models shows similar results. Since the interpolation method just adopts a linear combination of the trustingness and trustworthiness scores, it is cannot capture any feature-level correlation,

Table 3.3: Results comparison of different models on the retweet-only dataset.

Method	Model	Class	Accuracy	AUROC	Precision	Recall	F_1
Naive	**TN**	+	0.538	0.498	0.847	0.002	0.003
		-			0.538	0.999	0.699
	TW	+	0.541	0.524	0.629	0.018	0.036
		-			0.540	0.990	0.699
	TNnTW	+	0.539	0.513	0.609	0.016	0.032
		-			0.538	0.990	0.697
	L2_LR	+	0.537	0.500	0.891	0.000	0.001
		-			0.537	0.999	0.698
RNN	**GRU_NW**	+	0.618	0.655	0.555	0.478	0.513
		-			0.654	0.721	0.686
	GRU_NT	+	0.592	0.661	0.536	0.249	0.340
		-			0.606	0.842	0.705
	GRU_TR	+	0.664	0.716	0.590	0.664	0.625
		-			0.731	0.663	0.695
	LSTM_TR	+	0.698	0.731	0.626	0.706	0.664
		-			0.764	0.692	0.726

and as expected its performance lies in-between, with an accuracy of 0.539 and an AUROC of 0.513. In comparison with **Trustworthiness** model, while the metrics for the non-rumor class show almost identical results, rumor class shows slight decrease in all metrics.

L2_LR, which uses as features the embeddings based on believability-weighted edge scores also shows similar results. The AUROC score suggests that the model acts as a random predictor. This is a bit surprising since it appears that L2_LR which is a learning-based model fails to learn useful patterns from the generated embeddings. There could be reasons from two aspects: 1) A point to notice in baseline results is that both recall and F_1 scores are very low for rumor class. It shows a bias towards predicting most classes as non-rumor spreaders, resulting in very low *True Positives* and *False Positives*, which could be attributed to the fact that the edge-density ratio of the network is very low due to the sparsity of the network. This caused the trust scores to be less scaled. 2) Using such embeddings resulting from a sparse network, L2_LR, although being learning-based, does not consider higher level abstractions or correlations among these features, and also cannot naturally take into account the dependencies among the users in the sequence, rendering model's generalizability as low as the simple **Interpolation** model.

The **GRU_noweight** model employs the RNN algorithm for user classification but does not take into consideration the different strength of the retweet edges. The accuracy for the model is 0.618, which gives around 15.1% improvement over **L2_LR** model. AUROC for the model is 0.655, a significant improvement of 31% over the L2_LR model. More importantly, recall and F_1 scores on both classes show a significant improvement over baseline results. Thus we can conclude that the user representation learning based on even unweighted retweet relationships together with RNN classification improves the ability to identify rumor spreaders. This can be attributed to the strong learning capacity of RNN-based models by the use of hidden units capturing complex feature correlations and the consideration of long-distance temporal dependencies among the users.

The **GRU_notrust** model takes into account the basic following relationship while the **GRU_noweight** considers the unweighted retweet network. The accuracy is improved over the **L2_LR** by 10.2% to achieve 0.592. This can be attributed to the fact that the follow relationship network is sparser than the retweet network, which is a sub-graph of the retweet relationship network. However, **GRU_notrust** shows a minor decrease of 4.2% in accuracy and a slight increase of 0.91% in the AUROC score over **GRU_noweight** model.

The **GRU_trust** and **LSTM_trust** models consider the believability scores for the retweet edges based on the complementary trust measures derived from the overall topology of the network. The accuracy for **GRU_trust** is further improved over **GRU_notrust** by 12.1% and reaches 0.664. Also, its accuracy improves over the **GRU_noweight** and **L2_LR** models by 7.4% and 23.6%, respectively. In addition, the AUROC of **GRU_trust** increases by 8.3% over GRU_notrust to reach 0.716, and the AUROC for **LSTM_trust** further increases to 0.731. In terms of the *Precison*, *Recall* and F_1 measure, the performances on both classes also demonstrate consistent improvement. **LSTM_trust** performs better than **GRU_trust** with an improvement of 5.1% in terms of accuracy. Compared to the GRU model, it also shows improvement in all other metric values for both classes. This is because the LSTM model uses an additional memory unit which captures the temporal dependencies better than the

Table 3.4: Results comparison of different models on the reply-only dataset.

Method	Model	Class	Accuracy	AUROC	Precision	Recall	F_1
Naive	**TN**	+	0.506	0.501	0.506	0.999	0.672
		-			0.499	0.000	0.001
	TW	+	0.534	0.524	0.519	0.977	0.678
		-			0.790	0.087	0.156
	TNnTW	+	0.532	0.524	0.518	0.979	0.678
		-			0.797	0.08	0.145
	L2_LR	+	0.502	0.499	0.502	0.998	0.668
		-			0.526	0.002	0.004
RNN	**GRU_NW**	+	0.544	0.591	0.696	0.124	0.210
		-			0.529	0.948	0.679
	GRU_NT	+	0.546	0.594	0.646	0.164	0.262
		-			0.532	0.913	0.672
	GRU_TR	+	0.612	0.689	0.561	0.146	0.232
		-			0.618	0.923	0.741
	LSTM_TR	+	0.649	0.709	0.581	0.438	0.499
		-			70.678	0.789	0.729

GRU model. The higher performance of these two RNN-based models indicate that not only using as input the user embeddings derived from the believability-reweighted retweet network is advantageous, but also when the embeddings are used in a RNN framework that takes into consideration temporal dependencies, we can improve the final classification effectiveness on users.

3.7.2 Reply-only dataset

The reply dataset show some interesting results compared to the retweet dataset. In table 3.4 the **Trustworthiness** model shows an increase of 5.5% in accuracy and 4.6% in AUROC score than the **Trustingness** model, which suggests that trustworthiness score is a better metric than trustingness to classify users based on reply behavior. This can be explained as the fact that the reply network, similar to retweet network being very sparse, is also relatively smaller. In such a sparse, relatively small network, many star-shaped sub-graphs exist, with many nodes having no outgoing edges. As a result those nodes have the same trustingness scores rendering trustworthiness scores are better scaled than trustingness scores.

The **Interpolation** model, not surprisingly, shows a similar effect on accuracy as the retweet-only case, i.e., its performance lies between the two individual models. In terms of other metrics, it performs similarly as the **Trustworthiness** model, which indicates that the simple interpolation without deeper integration cannot boost the performance further.

Noticeably, the **L2_LR** model performs even worse than all other baselines with some larger margin than the retweet-only case, similar to a random guessing. This indicates that, being contrary to the general perception, learning is not helpful in this particular case. We attribute the poor performance of **L2_LR** to two reasons: 1) The reply-only network is sparser and relatively smaller, which results in relatively weaker user embeddings; 2) More importantly, unlike RNN model, L2_LR learns the model based on individual user embeddings and cannot easily consider other users in the context of information propagation, rendering a weak predicative model.

An interesting observation in the baseline model statistics is the poor recall and F_1 score of the non-rumor class. It shows a bias towards predicting most classes as rumor spreaders, resulting in very low *True Negatives* and *False Negatives*, which is the opposite case of the retweet-only dataset (see Section 3.7.1). This could be attributed to the fact that compared to retweeting, where a user chooses to broadcast a tweet to all its followers, replying may not be a very strong metric of trustingness because it could imply various types of stances such as supporting, against or neutral.

Similar to the results on the retweet-only dataset, using RNN-based models show significant improvement in accuracies and AUROC score. The accuracy of **GRU_noweight** model is 0.544, an increase of 8.3% over previous baseline. The AUROC score also shows an increase of 18.4%. **GRU_notrust** shows similar trend of improving results as the **GRU_noweight** model. Introducing Trust property in **GRU_trust** model improved accuracy by over 12% and AUROC by 15.9%. Interestingly, all the GRU models have low Recall and F_1 score for the rumor class. The **LSTM_trust** model shows the best results with the highest accuracy of 0.649, the highest AUROC score of 0.709, and more balanced metric values for both the classes.

3.7.3 Combined (retweet+reply) dataset

The retweet and reply datasets for user classification are merged to form this dataset, in which the retweeting and replying user sequences of the same source tweets are concatenated according to the sequence of time stamps of the user posting the responses.

Table 3.5: Results comparison of different models on the combined (retweet and reply) dataset.

Method	Model	Class	Accuracy	AUROC	Precison	Recall	F_1
Naive	**TN**	+	0.551	0.612	0.533	0.289	0.376
		-			0.558	0.779	0.650
	TW	+	0.536	0.551	0.584	0.012	0.023
		-			0.535	0.992	0.695
	TNnTW	+	0.553	0.574	0.537	0.295	0.381
		-			0.558	0.778	0.650
	L2_LR	+	0.629	0.647	0.602	0.602	0.602
		-			0.652	0.652	0.652
RNN	**GRU_NW**	+	0.679	0.704	0.667	0.480	0.559
		-			0.684	0.824	0.748
	GRU_NT	+	0.705	0.715	0.702	0.526	0.601
		-			0.707	0.836	0.767
	GRU_TR	+	0.724	0.740	0.745	0.529	0.619
		-			0.716	0.867	0.784
	LSTM_TR	+	0.738	0.795	0.758	0.559	0.643
		-			0.729	0.869	0.793

Experiments performed on this dataset also show interesting results.

Table 3.5 shows that for the first time the **Trustingness** model performs better than the **Trustworthiness** model, with an increase of 2.7% in accuracy and 11% in AUROC score. By examining the precision and recall specifically, we find that the improvement of the **Trustingness** model primarily comes from the significant rise of recall of rumor classes with only some minor drops on the precision of rumor class and the recall of non-rumor class. As shown in Table 3.2, the combined network is less sparse with more inter-connected nodes. It suggests that as most of the retweeter/replier nodes also have outgoing edges, the trustingness scores are better scaled in this case.

The **Interpolation** model also shows an improvement in all metrics. This can be attributed to the fact that both of the trust scores are better scaled due to the improved connectivity of the user network.

The **L2_LR** model shows an improvement of 13.7% in accuracy and 12.7% in AUROC score over the **Interpolation** model. Further observation on the precision and recall unveils that significant improvement lies in the rumor class where the F_1 is boosted by 58%. This indicates that the embeddings are much better due to the improved user network density giving rise to the better scaled trust scores.

Similar to Table 3.3, RNN-based models show significant improvement in performance. The accuracy for **GRU_noweight** model is 67.9%, giving an improvement of 7.9% over previous best baseline model. The model also shows a high AUROC score of 0.704. We can also observe that **GRU_notrust** outperforms **GRU_noweight** with 3.83% improvement in accuracy and 1.6% improvement in AUROC score, suggesting that the initial user network with only follow relationship is helpful since user's following behavior, although not strong, is a kind of, maybe the simplest form of, implication of user trust. In addition, the other two more advanced neural network models, i.e., **GRU_trust** and **LSTM_trust**, unsurprisingly make further improvement on the overall performance. The **LSTM_trust** model gives the best results in our experimental setup with an accuracy of 0.738 and AUROC score of 0.795. This can be attributed to the fact that the combined dataset is larger and the user network is denser.

3.7.4 Effect of change in hyper-parameters

Furthermore, we studied the influence of the hyper-parameters of TSM algorithm, i.e., the involvement score, and the LINE algorithm, i.e., vector length of embeddings. We found, however, that there was no significant variations in any of the evaluation metric values. This indicates that our model is not sensitive to the setting of these hyper parameters. Therefore, we do not intend to present these marginally different results in the section.

In addition, we also studied the effect of sparse user networks by considering alternative proxies such as '1 mention', '1 reply', and '1 retweet' without sparsity relief by excluding follow relationship, which leads to an edge connecting two users if there is one '@' mention, one reply or one retweet between them. However, the results are overall discouraging: The naive approaches were giving close to 50% performance in accuracy, and the RNN-based models gave only a slight improvement of around 55%. Therefore, we did not demonstrate them here in detail.

3.8 Conclusions and Future Work

In this chapter, we conducted a pilot study for the identification of rumor spreading users on Twitter based on computational trust measures. We proposed a machine learning framework by using the novel concept of believability which is defined based upon the trust measures of individual users in a large-scale retweet network. The key hypotheses are that: 1) The believability between two users is proportional to the trustingness of the responder and the trustworthiness of the source, where trustingness and trustworthiness are two complementary trust measures inferred from users' behaviors; 2) In return, using the believability for edge re-weighting on the networks can help enhance the learning of feature representation of users in the network, whereby the users' structural properties can be better preserved in terms of neighborhood similarity, signaling the distinctive roles different types of users play in spreading messages. We proposed LSTM and GRU based RNN models for user classification using user embeddings as input features that are generated from the believability re-weighted retweet network. Experimental results on a large real-world user classification dataset collected from Twitter demonstrate that the proposed RNN-based method outperformed four more straightforward methods with large margin.

The research work could be used to build Social Media Reputation framework (similar to how feedback scores for eBay users is calculated). We can associate a trust score to the users in social media that would let service providers to authenticate the veracity of information. Low trust users when detected can be monitored to prevent any future occurrence of rumor propagation. Thus this research can be used to make social media a more veracious source of information.

Overall, the performance on detecting rumor spreaders is not very high, indicating the task is difficult. In the future, we plan to extend our model by incorporating additional proxies of trust more than just retweets and replies. The stance of a user who replies the message is an important indicator as to whether the user is a rumor spreader or not. We will try to directly model user stances with deeper understanding of the concerned user's profile and interactions in the past for inferring the hidden intent and motivation of users retweeting a message.

We would also like to make a distinction between regular, non-regular users and bots to study their rumor spreading characteristics. We shall enhance our data collection to alleviate the sparsity of user trust networks which seems an important issue. In addition, we propose to study rumor information detection based on user trust networks and compare it with state-of-the-art detection systems. Meanwhile, we would be interested to investigate how to perform multiple detection tasks in rumorous environment such as detecting rumors and their spreaders at the same time.

Chapter 4

Risk Assessment of Population

4.1 Introduction

People use social networking platforms like Twitter, Facebook and Whatsapp not only to consume and share information but also their opinions about it. Ease of sharing has made it possible to spread information quickly, often without verifying it, resulting in fake news spreading. This has led to increase in interest among social media researchers to propose *early fake news spreading detection* models. In this context, it is not only important to detect false information early but also identify people in the early stages who are most likely to believe and spread the false information. This is so because the earlier the detection can be performed, the better the chance to stop fake news from pervading the system. While most of the related work in early detection systems has modeled content of the news itself, we propose a complementary approach that takes the network topology and historical user activity into account. As the CoViD-19 virus spread rapidly around the world in 2020, so has false information regarding various aspects pertaining to it[1]. The need for an early spreader detection model for fake news has never been more evident. Thus in this chapter, we propose a novel early spreader detection model using an inductive representation learning framework. The model quickly identifies spreaders before the false information penetrates deeper into a densely connected community and infects more nodes. The main contributions of the chapter are as follows:

[1] https://en.wikipedia.org/wiki/Misinformation_related_to_the_2019-20_coronavirus_pandemic

1. We propose an early fake news spreader detection framework using the Community Health Assessment model [51] and inter-personal trust [50]. To the best of our knowledge, this is the first early fake news spreader detection model proposed that relies on features extracted from underlying network structure and historical behavioral data instead of the content.

2. We implement our framework using inductive representation learning [100] where we sample neighborhood of nodes in a weighted network and aggregate their trust-based features.

3. We evaluate our proposed interpersonal trust based framework using multiple real Twitter networks and show that trust based modeling helps us identify false information spreaders with high accuracy, which makes the technique useful as a fake news detector.

4. We further observe that our model's accuracy when detecting true information spreaders is not as high as that for false information spreaders. This indicates that people are usually able to reason about true information from analyzing the content, and thus trust in their neighbors is not a very significant factor. However, determining the truth of *false information that is plausibly true* from content itself is difficult and hence we have to rely on sources we trust to believe in it or not. This makes nodes that are fake news spreaders and at the same time highly trusted by lots of people in the network, especially dangerous. We acknowledge that not all such *uber-spreaders* have ill intentions; some might be just ignorant. They all, nonetheless, have power to spread false information far and wide, with great speed. We believe this phenomenon needs greater study.

The rest of the chapter is organized as follows: We first discuss related work, then describe a motivating example for early spreader detection from a network structure perspective, and summarize past ideas that the proposed research builds upon. We then explain the proposed framework and how we model interpersonal trust with it followed by experimental analysis and finally give our concluding remarks and proposed future work.

4.2 Related Work

In this section we highlight related work from three domains our proposed research explores. They are: 1. Early fake news detection, 2. Graph neural networks, 3. Computational trust in social networks.

Early fake news detection

Research in the domain of early detection and containment of false information has a lot of potential. Zhao et al's. [80] work was seminal which used people's enquiry behavior to detect rumors early. Liu et al. [78] proposed a method for automatic identification of a rumor as conflicting microblogs. Sampson et al. [101] proposed a model to identify and analyze conversation links to detect rumors during their formative stages. Nguyen et al. [102] proposed a CNN-based early detection model for rumor classification which captured their temporal dynamics. Wu et al. [103] proposed the CERT framework which clustered data using emotional cues. Liu et al. [13] proposed to model the propagation path of a news as multivariate time series. Chen et al. [104] proposed a deep attention neural network that captured contextual variations of relevant posts over time.

Graph Neural Networks

Graph Neural Networks (GNNs) have received wide attention recently because they can be used to apply deep learning models to non-euclidean space such as social networks. Numerous mathematical models fo GNNs have been proposed but we highlight only their applications. They have been applied in multiple domains including computer vision [105, 106] where image and video data can be represented as grid structures, natural language processing [107, 108] for text classification and word-relation extraction, large scale recommender systems [106, 109] and transport networks [110] for traffic forcasting. Bian et al. [111] is a recent work that proposed a graph convolution network based fake news detection model. While GNNs are powerful models, they have not been applied much in the context of information spread in social networks.

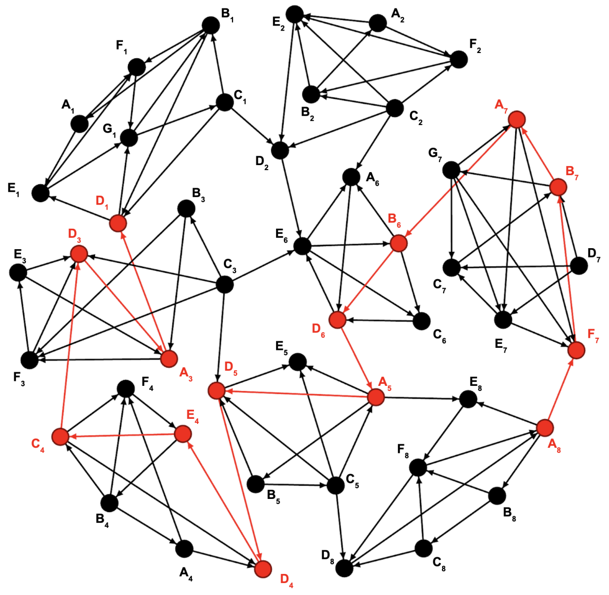


Figure 4.1: Motivating example for community perspective false information spreader prediction model.

Computational Trust in social networks

Computational Trust in social networks is a widely studied domain in which researchers have tried to assign trust scores to nodes of a network. Mui et al. [112] proposed a computational model for trust and reputation in social networks based on history of past interactions. Eigentrust by Kamvar et al. [47] assigned global trust value to people sharing and distributing files in a P2P network which helped an ordinary user in the network to identify malicious peers and isolate them from the network. Mishra et al. [48] proposed a model to compute the bias and prestige of nodes in social networks which used an iterative matrix algorithm using edge weights. Roy et al. [50] proposed the Trust in Social Media algorithm that assigned a pair of complementary trust scores called trustingness and trustworthiness to nodes. Our proposed research builds upon Roy's work.

4.3 Motivation and Preliminaries

To understand the role of network structure in early fake news spreader detection, consider the scenario illustrated in Figure 4.1. The network contains 8 communities. Subscript of a node denotes the community it belongs to. In the context of Twitter, directed edge $B_1 \rightarrow A_1$ represents B_1 follows A_1. Thus, a tweet flows from A_1 to B_1. If B_1 decides to retweet A_1's tweet, we say that B_1 has endorsed A_1's tweet, and that B_1 trusts A_1. Communities in social networks are modular groups, where within-group members are tightly connected, and intra-community trust is higher, compared to trust between members in different communities, who are at best loosely connected. The more B trusts A, the higher the chance that B will retweet A's tweet, and thus propagate A's message, whether it is true or false. The figure illustrates the spread of fake news starting from D_1 as it spreads across the network through A_3 till A_8. We consider two scenarios for early spreader detection:

1. Information reaches neighborhood of a community

Consider the scenario when a message is propagated by D_1, a neighborhood node for community 3. Node A_3 is exposed and is likely to spread the information, thus beginning spread of information into a densely connected community. Thus it is important to predict nodes in the boundary of communities that are likely to become information spreaders.

2. Information penetrates the community

Consider the scenario where A_3 decides to propagate a message. Nodes B_3, D_3 and E_3, which are immediate followers of A_3 are now exposed to the information. Due to their close proximity, they are vulnerable to believing the endorser. The remaining nodes of the community (C_3, F_3) are two steps away from A_3. Similarly for community 8 when the message has reached node A_8, nodes D_8 and F_8 are one step away and remaining community members (E_8, C_8, B_8) are two steps away. Intuitively, in a closely-knit community structure if one of the nodes decides to spread a piece of information, the likelihood of it spreading quickly within the entire community is very high. Thus it is important to detect nodes within a community that are likely to become information spreaders in the early stages to protect the health of the entire community.

4.4 Preliminaries

4.4.1 Community health assessment model

Consider the scenario as explained in the motivating example where the fake news spreading outside the community has reached its neighboring nodes. If one of the community node believes the news and becomes spreader, the likelihood of other community members becoming spreaders would be high, due to high connectivity among all members. Through the Community Health Assessment model [51] we proposed the ideas of neighbor, boundary and core nodes of a community.

The three types of nodes with respect to a community which are affected during the process of news spreading are explained below:

1. Neighbor nodes: These nodes are directly connected to at least one node of the community. The set of neighbor nodes is denoted by $\mathcal{N}$. They are not a part of the community.

2. Boundary nodes: These are community nodes that are directly connected to at least one neighbor node. The set of boundary nodes is denoted by $\mathcal{B}$. It is important to note that only community nodes that have an outgoing edge towards a neighbor nodes are in $\mathcal{B}$.

3. Core nodes: Community nodes that are only connected to members within the community. The set of core nodes is denoted by $\mathcal{C}$.

The neighbor, boundary and core nodes for Figure 4.1 are listed in Table 4.1.

4.4.2 Trustingness and trustworthiness

In the context of social media, researchers have used social networks to understand how trust manifests among users. A recent work is the Trust in Social Media (TSM) algorithm which assigns a pair of complementary trust scores to each actor called *Trustingness* and *Trustworthiness*. *Trustingness (ti)* quantifies the propensity of an actor to trust its neighbors and *Trustworthiness (tw)* quantifies the willingness of the neighbors to trust the actor. The TSM algorithm takes a user network, i.e., a directed graph

Table 4.1: Neighbor, boundary and core nodes for communities in Figure 4.1.

com	$\mathcal{N}_{com}$	$\mathcal{B}_{com}$	$\mathcal{C}_{com}$
1	D_2	C_1	A_1,B_1,E_1,D_1,F_1,G_1
2	A_6,E_6	C_2,D_2	A_2,B_2,E_2,F_2
3	D_1,D_5,E_6	A_3,C_3	B_3,D_3,E_3,F_3
4	D_3	C_4	A_4,B_4,D_4,E_4,F_4
5	D_4,D_8,E_8	D_5,A_5,C_5	E_5,B_5
6	A_5	D_6	A_6,B_6,C_6,E_6
7	B_6	A_7	B_7,C_7,D_7,E_7,F_7, G_7
8	$F7$	A_8	B_8,C_8,D_8,E_8,F_8

$\mathcal{G}(\mathcal{V},\mathcal{E})$, as input together with a specified convergence criteria or a maximum permitted number of iterations. In each iteration for every node in the network, trustingness and trustworthiness are computed using the equations mentioned below:

$$ti(v) = \sum_{\forall x \in out(v)} \left(\frac{w(v,x)}{1 + (tw(x))^s} \right) \tag{4.1}$$

$$tw(u) = \sum_{\forall x \in in(u)} \left(\frac{w(x,u)}{1 + (ti(x))^s} \right) \tag{4.2}$$

where $u, v, x \in \mathcal{V}$ are user nodes, $ti(v)$ and $tw(u)$ are trustingness and trustworthiness scores of v and u, respectively, $w(v,x)$ is the weight of edge from v to x, $out(v)$ is the set of out-edges of v, $in(u)$ is the set of in-edges of u, and s is the involvement score of the network. Involvement is basically the potential risk an actor takes when creating a link in the network, which is set to a constant empirically. The details of the algorithm are excluded due to space constraints and can be found in [50].

4.4.3 Believability

Believability (bel) is an edge score derived from Trustingness and Trustworthiness scores. It helps us quantify how likely is the receiver of a message to believe its sender. Believability for a directed edge is naturally computed as a function of the trustworthiness of the sender and the trustingness of the receiver. It quantifies the strength that v trusts on u when v decides to retweet a news endorsed by u. Therefore, v is very likely to believe in u if:

1. u has a high trustworthiness score, i.e., u is highly likely to be trusted by other users in the network, or
2. v has a high trustingness score, i.e., v is highly likely to trust others. So, the believability score is supposed to be proportional to the two values above, which can be jointly determined and computed as follow:

$$bel_{uv} = tw(u) * ti(v) \tag{4.3}$$

The idea has been applied in [52] where a classification model was built to identify rumor spreaders in twitter network.

Problem Formulation

Given a directed social network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ comprising disjoint modular communities (ϕ), with each community ($com \in \phi$) having well-defined neighbor nodes ($\mathcal{N}_{com}$), boundary nodes ($\mathcal{B}_{com}$) and core nodes ($\mathcal{C}_{com}$). Aggregating topology-based (top) and activity-based (act) trust properties from nodes sampled from depth K (where $Nbr_{K=1}(b) \subseteq \mathcal{N}_{com}$), we want to predict boundary nodes b that are most likely to become information spreaders (b_{sp}). Similarly, we aggregate nodes sampled from depth K (where $Nbr_{K=1}(c) \subseteq \mathcal{B}_{com}$) to predict core nodes c that are most likely to become information spreaders (c_{sp}).

4.5 Inductive Representation Learning

As fake news spreads rapidly, network structure around the spreaders also evolves quickly. Thus, it is important to have a scalable model that is able to quickly learn meaningful representations for newly seen (i.e. exposed) nodes without relying on the complete network structure. Most graph representation learning techniques, however, employ a *transductive* approach to learning node representations which optimizes the embeddings for nodes based on the entire graph structure. We employ an *inductive* approach inspired from GraphSAGE [100] to generate embeddings for the nodes as the information spreading network gradually evolves, as explained in Figure 4.2. It learns an embedding function that generalizes to unseen node structures in the network which

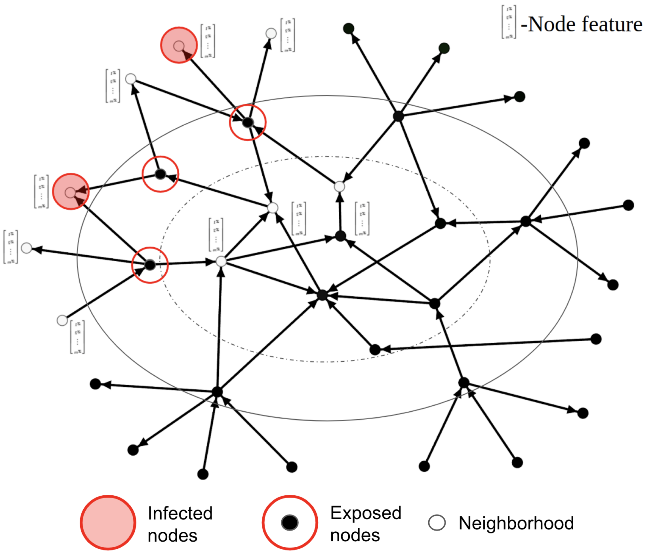
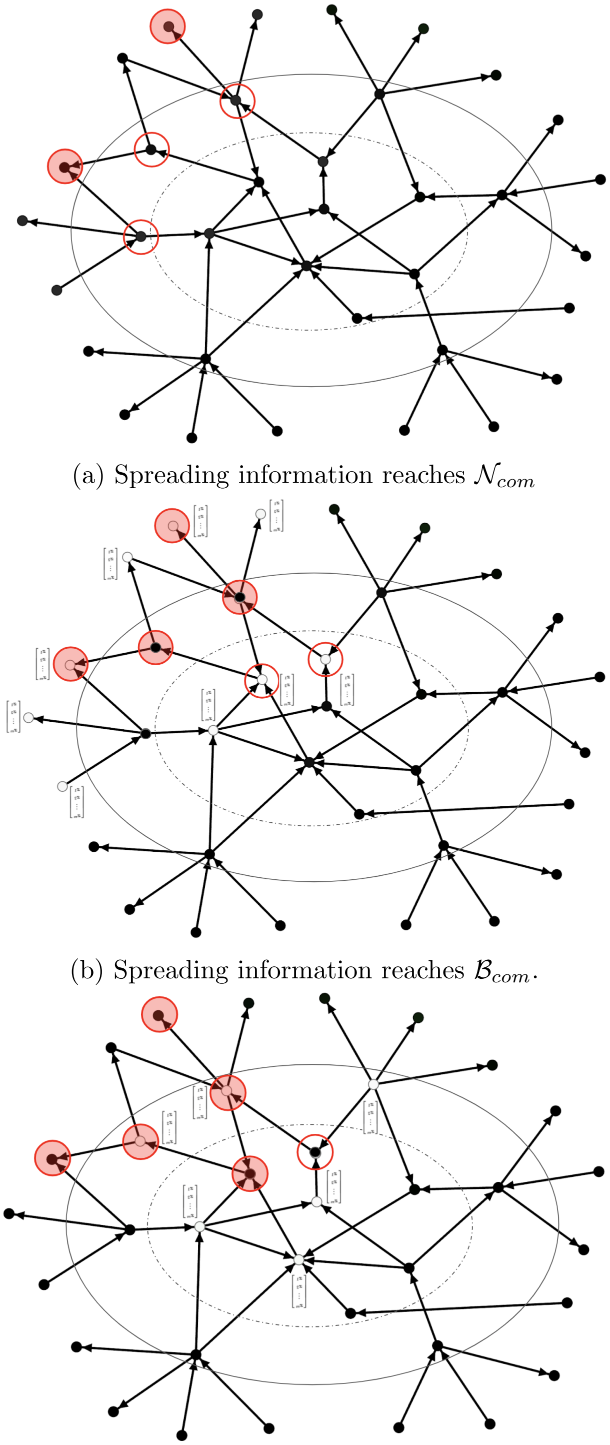


Figure 4.2: Inductive representation learning model for detection of false information spreaders.

could become potential information spreaders. The idea is to simultaneously learn the topological structure and node features from the neighborhood (Nbr) nodes, by training a set of aggregator functions instead of individual node embeddings. Using an inductive representation learning model we learn features of the exposed population (i.e. followers of fake news spreaders) by aggregating trust-based features from their neighborhood nodes. Figure 4.3 shows how we model the proposed approach with community perspective. Nodes outside the solid oval represent $\mathcal{N}_{com}$, between solid and dotted oval represents $\mathcal{B}_{com}$ and within the dotted oval represents $\mathcal{C}_{com}$. (a) shows that false information spread has reached the two neighbor nodes (highlighted in red). Three boundary nodes (circled in red) are exposed to the information. In (b) we learn representations for the exposed boundary nodes by aggregating features of their local neighborhood structure (denoted by white nodes). Two out of the three boundary nodes that become spreaders are highlighted and the exposed core nodes are circled. Similarly, in (c) we learn representations for the exposed core nodes by aggregating their local neighborhood features. One core node becomes a spreader and the community is now vulnerable to fake news spreading.



(a) Spreading information reaches $\mathcal{N}_{com}$

(b) Spreading information reaches $\mathcal{B}_{com}$.

(c) Spreading information reaches $\mathcal{C}_{com}$.

Figure 4.3: False information spreader prediction using Community Health Assessment model

The proposed framework is explained as follows: First we generate a weighted information spreading network based on interpersonal trust. We then sample neighborhood with a probability proportional to the edge weights. For the sampled neighborhood we aggregate their feature representations. Finally we explain the loss function used to learn parameters of the model.

4.5.1 Generating weighted graph

Graph of the information spreading network has edge weights that quantify the likelihood of trust formation between senders and receivers. Once we compute these edge scores using techniques mentioned in Table 4.2, we normalize weights for all out-edges connecting the boundary node.

$$\hat{w}_{bx} = \frac{bel_{bx}}{\sum_{\forall x \in out(b)} bel_{bx}} \tag{4.4}$$

Similarly we normalize weights for all in-edges connecting the boundary node.

4.5.2 Sampling neighborhood

Instead of sampling neighborhood as a uniform distribution, we sample a subset of neighbors proportional to the weights of the edges connecting them. Sampling is done recursively till depth K. The idea is to learn features from neighbors proportional to the level of inter-personal trust. Algorithm 2 explains the sampling strategy.

Algorithm 2 Sample neighborhood (SA)

```
Input: $\mathcal{G}(\mathcal{V}, \mathcal{E})$: Information spreading network,
$K$: Sampling depth, $\mathcal{B}_{com}$: Boundary nodes of community.
Output: $Nbr_K(b)$: Sampled neighborhood for $b$ till depth $K$.
$\phi \leftarrow$ Disjoint modular communities in $\mathcal{G}$ for each $com \in \phi$ do
    for each $b \in \mathcal{B}_{com}$ do
        $Nbr_0(b) \leftarrow \{b\}$
        for $k = 1 \ldots K$ do
            $Nbr_k(b) \leftarrow Nbr_{k-1}(b) \cup SA_k(b)_{Eq.\ 4.4}$
        end
    end
end
```

4.5.3 Aggregating features

After sampling neighborhood as an unordered set, we aggregate the embeddings of sampled nodes till depth K recursively for each boundary node. The intuition is that at each depth, the boundary nodes incrementally learn trust-based features from the sampled neighborhood. Three aggregation architectures namely mean, LSTM and pooling explained in [100] can be used. For simplicity, we only apply the mean aggregator, which takes the mean of representations h_u^{k-1} where $u \in Nbr_{k-1}(b)$. The aggregators (Mean, LSTM and Pooling) is represented below: 1. Mean aggregator: The aggregator takes the mean of h_u^{k-1} where $u \in Nbr_{k-1}(b)$ and is represented below:

$$h_b^k \leftarrow \sigma(W_b^k.Mean(\{h_b^{k-1}\} \cup \{h^{k-1}_{u(\forall u \in Nbr(b))})\}) \tag{4.5}$$

2. LSTM aggregator: The aggregator is based on the LSTM architecture, applied on neighborhood nodes considered as an unordered set.

$$h_b^k \leftarrow LSTM(\{h^{k-1}_{u(\forall u \in Nbr(b))})\} \tag{4.6}$$

3. Pooling aggregator: Each neighborhood node is fed into a fully-connected neural network and then elementwise max-pool operation is applied on the obtained transformation.

$$h_b^k \leftarrow max(\{\sigma(W_{pool} h^{k-1}_{u(\forall u \in Nbr(b))} + b)\}) \tag{4.7}$$

Algorithm 3 explains the aggregation strategy.

4.5.4 Learning parameters

The weight matrices in Algorithm 3 are tuned using stochastic gradient descent on a loss function in order to learn the parameters. We train the model to minimize cross-entropy.

$$Loss(\hat{y}, y) = -\sum_{\forall b \in \mathcal{B}_{com}} \sum_{i \in \{b_{Sp}, b_{\bar{Sp}}\}} y_i log \hat{y}_i \tag{4.8}$$

Algorithm 3 Aggregate features (GE)

```
Input: $\mathcal{G}(\mathcal{V}, \mathcal{E})$: Information spreading network,
$K$: Sampling depth, $\mathcal{B}_{com}$: Boundary nodes of community, $x_{v(\forall v \in \mathcal{V})}$: Node features.
Output: $z_b^k$: Embedding vector for $b$.
$\phi \leftarrow$ Disjoint communities in $\mathcal{G}$  for each $com \in \phi$ do
    for each $b \in \mathcal{B}_{com}$ do
        $h_b^0 \leftarrow x_b$
        for $k = 1 \dots K$ do
            $h_{Nbr(b)}^k \leftarrow GE_k(h_{u(\forall u \in Nbr(b))}^{k-1})$
            $h_b^k \leftarrow \sigma(W_b^k.Concat(h_b^{k-1}, h_{Nbr(b)}^k))_{Eq.\ 4.5}$
        end
        $h_b^k \leftarrow h_b^k / ||h_b^k||_2$
    end
    $z_b^k \leftarrow h_b^k$
end
```

Table 4.2: Trust based strategy for sampling and aggregating features.

		Topology (top)	**Activity (act)**
Sample	w_{xv}	bel_{xv}	RT_{xv}
Aggregate	*trusting others*	$ti(x)$	$\sum_{\forall i \in t}\{1 \text{ if } i = RT_x \text{ else } 0\}/n(t)$
	trusted by others	$tw(x)$	$\sum_{\forall i \in t} i_x^{n(RT)}/n(t)$

The loss function is modeled to predict whether the boundary node is an information spreader (b_{Sp}) or a non-spreader ($b_{\bar{Sp}}$). y represents the actual class (2-dimensional multinomial distribution of [1,0] for spreader and [0,1] for non-spreader) and $\hat{y}$ represents the predicted class.

We extend the model for $\mathcal{C}_{com}$ to identify the core node spreaders (c_{Sp}) and non-spreaders ($c_{\bar{Sp}}$). Considering boundary nodes have denser neighborhood compared to core nodes, we later analyze whether the proposed model is more sensitive to density of neighborhood structure or the aggregated features. The implementation code is made publicly available[2].

[2] https://github.com/BhavtoshRath/Proactive_Spreader_Detection

4.6 Modeling Interpersonal Trust

As explained in the preliminaries section, interpersonal trust has been applied successfully in the past to model fake news spreading. Thus we model our node representation learning problem using interpersonal trust to predict whether a node is a spreader or not. We first apply a non-uniform neighborhood sampling strategy using weighted graph (where edge weights quantify the likelihood of trust formation). We then aggregate two trust features: 1) The likelihood to 'trust others' and 2) The likelihood to be 'trusted by others'. We use two kinds of interpersonal-trust: *Topology-based* computed from the social network topology and *Activity-based* computed using timeline activity data collected for every node using twitter API. We use trustingness ($ti(x)$) and trustworthiness ($tw(x)$) scores of node x obtained from TSM as proxy for topology-based trust features and the average number of times x retweets (RT_x) denoted by $\sum_{\forall i \in t}\{1$ if $i = RT_x$ else $0\}/n(t)$ and average number of times x is retweeted ($n(RT)$) denoted by $\sum_{\forall i \in t} i_x^{n(RT)}/n(t)$ as activity-based trust features (t represents recent tweets posted in x's timeline[3]). For an edge from x to v, the topology-based edge weight is the believability score (bel_{xv}) and activity-based edge weight is the number of times x retweets v (RT_{xv}). Explained strategy is summarized in Table 4.2.

4.7 Experiments and Results

4.7.1 Ground truth and data collection

We evaluate our proposed model using real world Twitter datasets. We obtained the ground truth of false information and the refuting true information from *altnews.in*, a popular fact checking website. The source tweet related to the information was obtained directly as a tweet embedded in the website or through a keyword based search on Twitter. From the source tweet we generated the source tweeter and the retweeters (proxy for spreaders), follower-following network of the spreaders (proxy for network) and the timeline data for all nodes in the network (to generate trust-based features) using the Twitter API. Besides evaluating our model on false information (F) and the refuting true information (T) networks separately, we also evaluated on network

[3] Due to time restrictions we collected only 10 most recent tweets.

Table 4.3: Network and community dataset statistics.

	F	**T**	**F ∪ T**
No. of nodes	1,709,246	1,161,607	2,554,061
No. of edges	3,770,532	2,086,672	5,857,205
No. of spreaders	2,246	643	2,862
No. of communities	58	39	52
No. of nodes in $\mathcal{N}$	209,311	94,884	276,567
No. of spreaders in $\mathcal{N}$	19,403	5,350	22,868
No. of nodes in $\mathcal{B}$	217,373	136,350	345,312
No. of spreaders in $\mathcal{B}$	2,152	611	2,738
No. of nodes in $\mathcal{C}$	1,278,885	862,778	1,893,493
No. of spreaders in $\mathcal{C}$	94	31	98

Table 4.4: Results comparison of different models for boundary node spreader prediction.

	F				**T**				**F ∪ T**			
	Accu.	**Prec.**	**Rec.**	**F1**	**Accu.**	**Prec.**	**Rec.**	**F1**	**Accu.**	**Prec.**	**Rec.**	**F1**
Trusting others	0.58	0.612	0.329	0.396	0.615	0.697	0.450	0.519	0.510	0.522	0.888	0.603
Trusted by others	0.608	0.631	0.384	0.455	0.646	0.713	0.500	0.585	0.518	0.513	0.916	0.638
Interpolation	0.622	0.635	0.426	0.498	0.661	0.768	0.496	0.588	0.524	0.526	0.846	0.611
LINE	0.709	0.784	0.593	0.669	0.692	0.763	0.567	0.647	0.589	0.602	0.517	0.554
$SA_{rand}GE_{top}$	0.870	0.879	0.862	0.866	0.776	0.858	0.667	0.748	0.599	0.605	0.570	0.583
$SA_{rand}GE_{act}$	0.777	0.845	0.689	0.754	0.728	0.814	0.612	0.688	0.566	0.572	0.539	0.547
GCN_{top}	0.839	0.887	0.784	0.832	0.775	0.921	0.595	0.723	0.592	0.649	0.646	0.647
GCN_{act}	0.807	0.849	0.750	0.796	0.740	0.835	0.591	0.693	0.576	0.640	0.612	0.626
$SA_{top}GE_{top}$	**0.937**	**0.918**	**0.965**	**0.939**	**0.834**	**0.927**	**0.732**	**0.815**	**0.616**	**0.630**	**0.561**	**0.592**
$SA_{top}GE_{act}$	0.912	0.899	0.935	0.915	0.800	0.884	0.699	0.777	0.584	0.601	0.504	0.545
$SA_{act}GE_{top}$	0.838	0.854	0.816	0.833	0.763	0.817	0.686	0.743	0.582	0.589	0.542	0.559
$SA_{act}GE_{act}$	0.804	0.853	0.737	0.786	0.735	0.800	0.634	0.706	0.561	0.570	0.542	0.539

obtained by combining them (F ∪ T). Metadata for the network dataset aggregated for all news events is summarized in Table 4.3.

4.7.2 Settings and protocols

We obtained the topology-based measures by running TSM algorithm on the network to obtained *ti*, *tw* for all nodes and *bel* for all edges. We used the generic settings for TSM parameters (number of iterations = 100, involvement score = 0.391) by refering to [50]. We found the disjoint modular communities using Louvain community detection algorithm [72] and identified the neighbor, boundary and core nodes for every community using Community Health Assessment model. We then generated the activity-based measures from timeline data of the nodes. The embeddings are generated using the forward propagation method shown in Algorithm 3, assuming that the model parameters are learnt using Equation 4.8. Due to class imbalance we undersample the majority class to obtain balanced spreader and non-spreader class distribution. The size of hidden units is set to 128 and the learning rate is set to 0.001. We used rectified linear units as the non-linear activation function. The batch size was adjusted for optimal performance depending on the size of training dataset. Due to the heavy-tailed nature of degree distributions of edges in social networks we downsample before modeling, which ensured that the neighborhood information is stored in dense adjaceny lists. This drastically reduces our run time, which is ideal for early detection of spreaders. We also set sampling depth K=1 because the network constitutes only immediate follower-following nodes of the spreaders. We compared results for the following models, including baselines:

1) $Trusting\ others$: Intuitively, users with high likelihood to trust others tend to be spreaders of false information. This model learns a threshold based on correlation between 'trusting others' features (both topology- and activity- based) and user ground truth.

2) $Trusted\ by\ others$: Intuitively, users with high likelihood to be trusted by others tend to be spreaders of false information. Like the previous model, this model learns a threshold based on correlation between 'trusted by others' features (both topology- and activity- based) and user ground truth.

3) $Interpolation$: This model linearly combines 'trusting others' and 'trusted by others' features to find an optimal threshold.

4) $LINE$: This model applies LINE [81] which serves as transductive learning baseline.

5) $SA_{rand}GE_{top}$: This model applies the inductive learning by sampling neighborhood considered as uniform distribution and aggregating only topology based features.

6) $SA_{rand}GE_{act}$: This model applies the inductive learning by sampling neighborhood considered as uniform distribution and aggregating only activity based features.

7) $SA_{rand}GE_{top+act}$: This model applies the inductive learning by sampling neighborhood considered as uniform distribution and aggregating both topology and activity based features.

8) $SA_{top}GE_{top}$: Instead of random sampling, we sample on the believability (bel)

Table 4.5: Results comparison of different models for core node spreader prediction.

	F				**T**				**F ∪ T**			
	Accu.	**Prec.**	**Rec.**	**F1**	**Accu.**	**Prec.**	**Rec.**	**F1**	**Accu.**	**Prec.**	**Rec.**	**F1**
Trusting others	0.553	0.643	0.298	0.388	0.569	0.585	0.338	0.414	0.521	0.511	0.95	0.659
Trusted by others	0.569	0.628	0.411	0.481	0.614	0.694	0.503	0.508	0.540	0.523	0.952	0.673
Interpolation	0.609	0.730	0.400	0.492	0.640	0.681	0.438	0.521	0.550	0.548	0.764	0.608
LINE	0.721	0.821	0.625	0.681	0.672	0.870	0.467	0.579	0.577	0.572	0.676	0.602
$SA_{rand}GE_{top}$	0.842	0.900	0.802	0.838	0.726	0.880	0.574	0.664	0.656	0.651	0.707	0.665
$SA_{rand}GE_{act}$	0.798	0.893	0.700	0.764	0.658	0.742	0.448	0.523	0.597	0.631	0.512	0.548
GCN_{top}	0.755	0.972	0.524	0.681	0.739	0.698	0.839	0.762	0.683	0.731	0.537	0.619
GCN_{act}	0.731	0.741	0.705	0.722	0.701	0.735	0.641	0.684	0.657	0.691	0.561	0.619
$SA_{top}GE_{top}$	**0.916**	**0.940**	**0.892**	**0.912**	**0.836**	**0.895**	**0.787**	**0.825**	**0.734**	**0.725**	**0.823**	**0.750**
$SA_{top}GE_{act}$	0.891	0.929	0.849	0.884	0.800	0.931	0.684	0.769	0.685	0.703	0.677	0.682
$SA_{act}GE_{top}$	0.868	0.941	0.788	0.854	0.771	0.962	0.598	0.712	0.648	0.688	0.651	0.641
$SA_{act}GE_{act}$	0.846	0.847	0.858	0.846	0.707	0.827	0.581	0.661	0.619	0.694	0.522	0.567

weighted network and aggregate their topology based features.

9) $SA_{top}GE_{act}$: Sampling approach is identical to 8) but we aggregate neighborhood's activity based features.

10) $SA_{act}GE_{top}$: We sample neighborhood non-uniformly on the retweet count (RT) weighted network and aggregate their topology based features.

11) $SA_{act}GE_{act}$: Sampling approach is identical to 10) but we aggregate neighborhood's activity based features.

Baseline models 1) - 3) are inspired from [113] that considers features based on trust. Baseline model 4) considers features based on network structure. Proposed models 5) - 11) integrate both neighborhood structure and node features. We analyze the best combination of sampling and aggregating strategy that predicts spreader node with highest accuracy. For evaluation we did a 70-15-15 train-validation-test split of the dataset. We used 5-fold cross validation and four common metrics: Accuracy, Precision, Recall and F1 score. We only show results for the spreader class.

4.8 Analysis of Results

We evaluated our proposed model on 10 debunked news events. For each news event we obtained three types of networks: network for the false information (F), for the true information (T) refuting it and the network obtained by combining them (F ∪ T). Thus we ran our models on 30 large-scale networks.

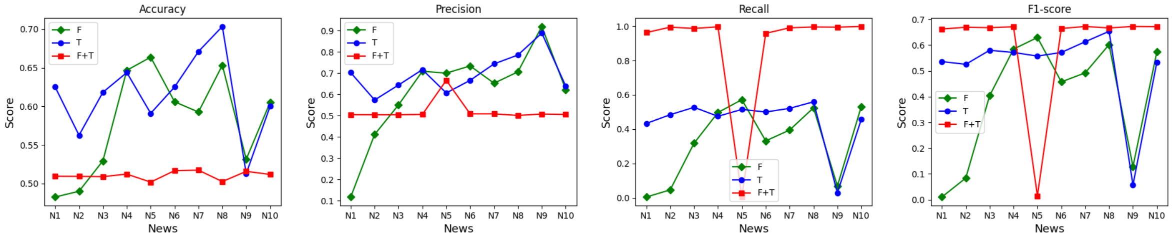
4.8.1 Boundary node analysis

Table 4.4 summarizes results for the boundary node prediction aggregated for all news. The results show that F performs better than T on almost every metrics while F ∪ T performs poorly. The poor performance of F ∪ T networks could be attributed to the fact that there is minimal overlap of nodes in F and T networks (12%) which causes the F ∪ T networks to have sparser communities. Also false and true information spreaders are together considered as spreader class which could be affecting the model performance. While comparing the baseline models, $Trusted\ by\ others$ model performs better than the $Trusting\ others$ model with an improvement in accuracy of 4.8%, 5% and 1.5% for F, T and F ∪ T networks respectively. $Interpolation$ model shows a further improvement of 2.3%, 2.3% and 1.1% for F, T and F ∪ T networks respectively over $trustingness$ model. $LINE$ and GCN baselines show significant improvement on all metrics for F networks compared to T or F ∪ T networks. We see further substantial increase in performance for each type of network using inductive learning models. Comparing the two random sampler models (i.e. $SA_{rand}GE_{top}$, $SA_{rand}GE_{act}$) we see that topology-based features of the neighborhood perform better than activity-based features. Similar trend is observed for topology-based sampler models (i.e. $SA_{top}GE_{top}$, $SA_{top}GE_{act}$) where model using topology-based aggregator performs better than activity-based aggregator. Same is the case for activity-based sampler models (i.e. $SA_{act}GE_{top}$, $SA_{act}GE_{act}$). Integrating *top* and *act* does not show any significant improvement over *top* only models. Thus we can conclude that interpersonal trust based modeling in the inductive learning framework is able to predict false information spreaders better than true information spreaders. We also observe that topology-based sampling and aggregating strategies perform better than activity-based strategies. The low performance of activity-based strategies could be attributed to the fact that many Twitter users are either inactive users or users with strict privacy settings whose timeline data could not be retrieved. Also recent 10 activities on a user's timeline might be insufficient data to capture activity-based trust dynamics. For each type of network, we observe that $SA_{top}GE_{top}$ model performs the best, with F having accuracy of 93.3%, which is higher than 12.3% and 52.1% over T and F ∪ T networks respectively. We observe a clear distinction in performance, with F networks performing better than T, which in turn is better than F ∪ T. An interesting observation is the high precision values for T. This is because the percentage of predicted spreaders which are non-spreaders tends to be lower for T network than for F network.

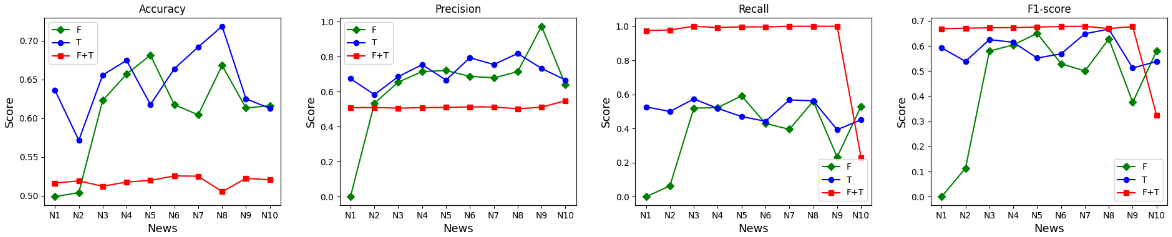
4.8.2 Core node analysis

Table 4.5 summarizes results of the model for predicting core nodes aggregated for all news. The overall performance trend is identical to the results shown for boundary nodes in Table 4.4. Among the baseline models, *Interpolation* model performs better than *Trusted by others* and *Trusting others* models. *LINE* and *GCN* based models show significant improvement over trust feature baselines on all metrics. Among inductive learning models, topology-based trust modeling shows better performance than activity-based trust modeling. Also F networks perform better than T networks, which in turn perform better than F ∪ T networks. Among random sampler models, $SA_{rand}GE_{top}$ has the highest accuracy of 84.2%, 72.6% and 65.6% for F, T and F ∪ T networks respectively. Among topology-based sampler models $SA_{top}GE_{top}$ performs better over $SA_{top}GE_{act}$ with an increase in accuracy of 2.8%, 4.5% and 7.1% for F, T and F ∪ T networks respectively. Activity-based sampler models also show identical trend with $SA_{act}GE_{top}$ performing better than $SA_{act}GE_{act}$ with an increase in accuracy of 2.6%, 9% and 4.6% for F, T and F ∪ T networks respectively. Among all models $SA_{top}GE_{top}$ shows the best overall performance. True information network for N10 is excluded from analysis as it did not have sufficient spreaders to train our model on. Even though the number of core nodes is much higher than boundary nodes, the number of core spreaders is much smaller than boundary node spreaders. Thus the model fails to learn meaningful representations for core nodes due to smaller training dataset.

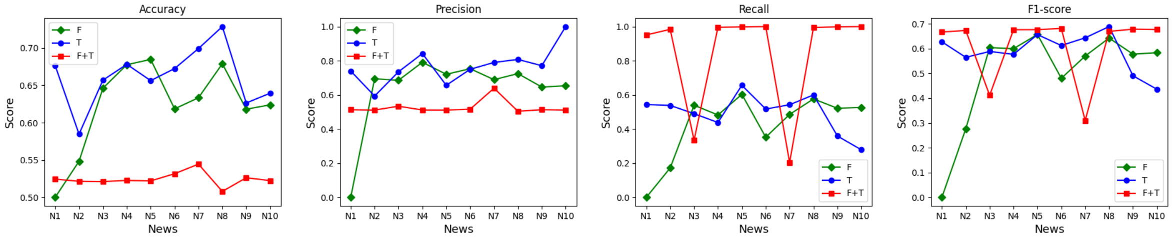
Summary: Comparing the prediction performance of core and boundary spreaders we can conclude that our model's performance is more sensitive to aggregated features and training dataset size compared to density of neighborhood. Metric performance of boundary and node prediction by models for news N1 to N10 is shown in Figure 4.4 and Figure 4.5 respectively.



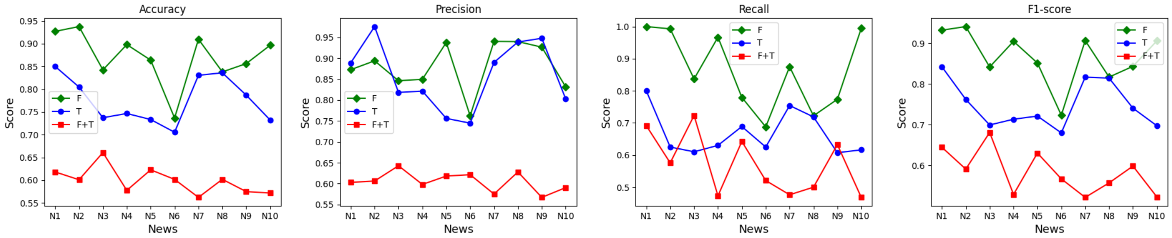
(a) Boundary node prediction using $Trusting\ others$



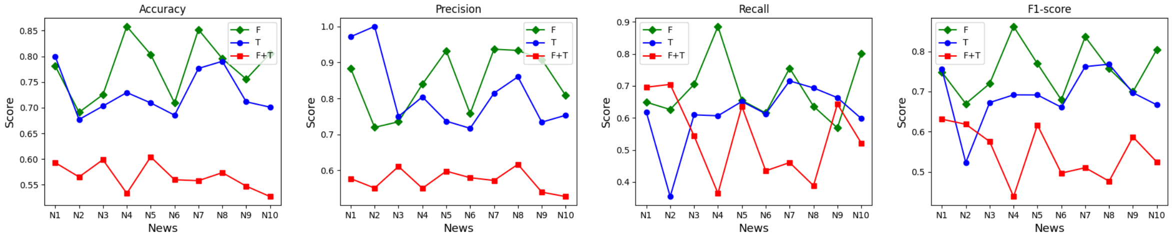
(b) Boundary node prediction using $Trusted\ by\ others$



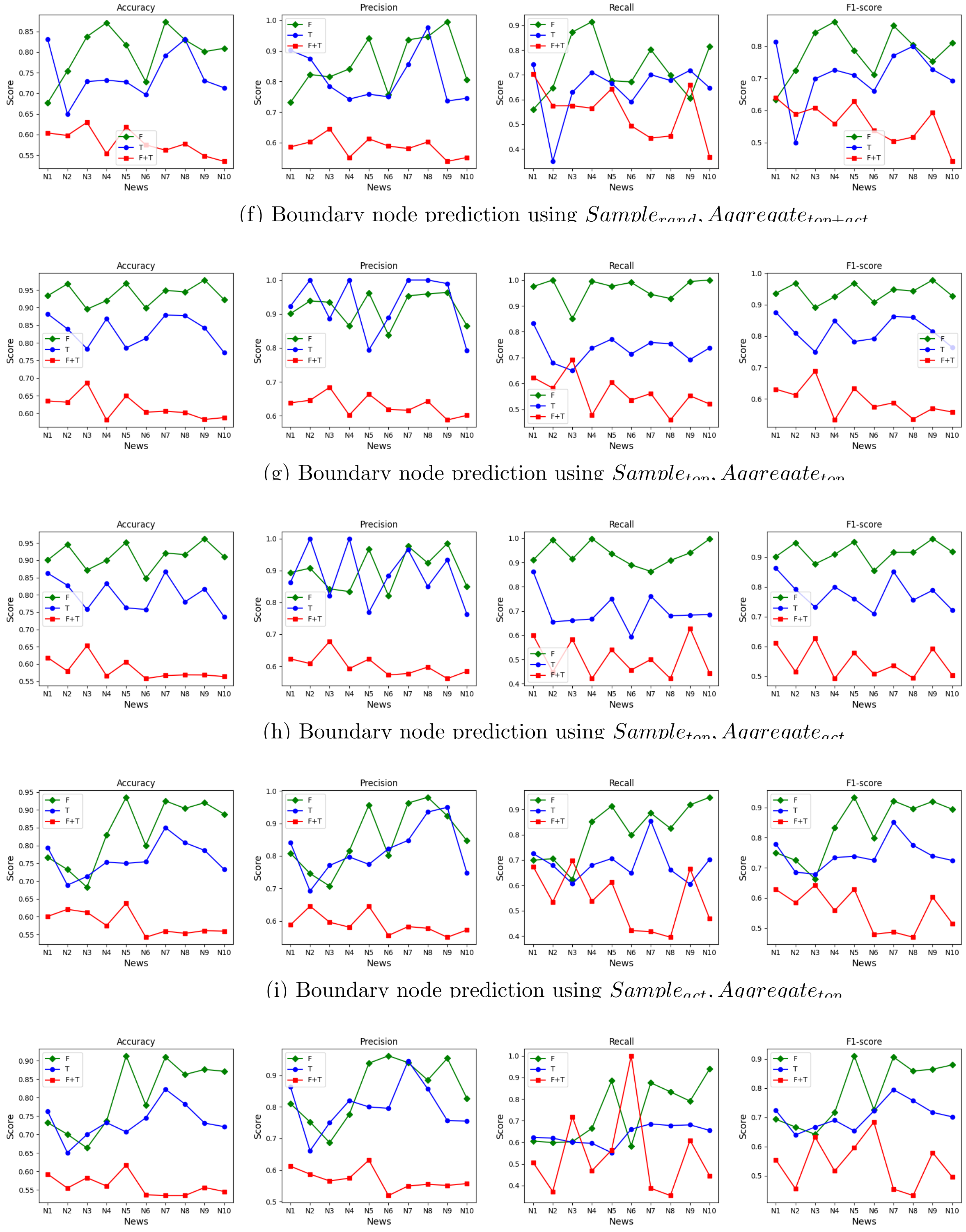
(c) Boundary node prediction using $Interpolation$



(d) Boundary node prediction using $Sample_{rand}, Aggregate_{top}$



(e) Boundary node prediction using $Sample_{rand}, Aggregate_{act}$



(f) Boundary node prediction using $Sample_{rand}, Aggregate_{top+act}$

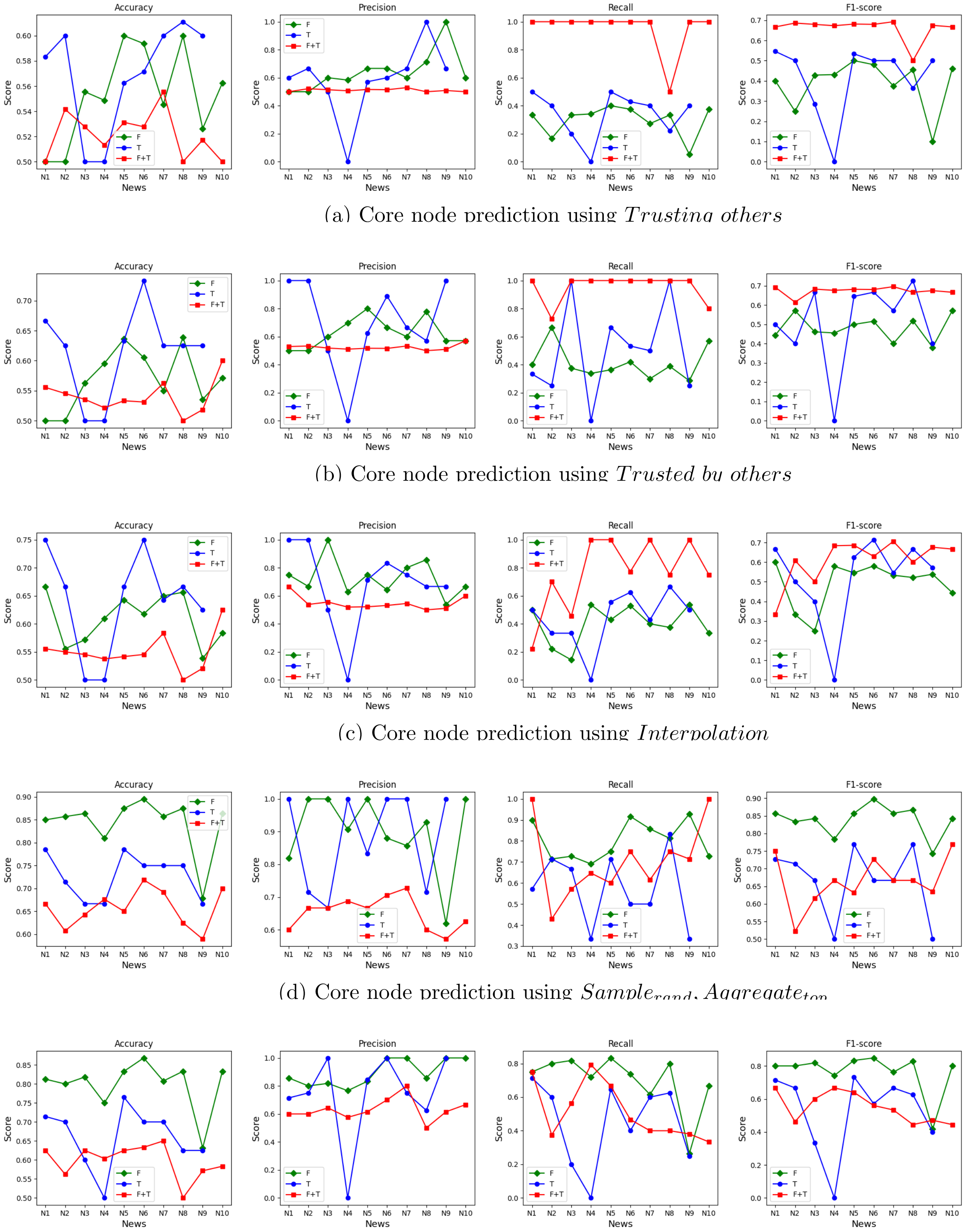
(g) Boundary node prediction using $Sample_{top}, Aggregate_{top}$

(h) Boundary node prediction using $Sample_{top}, Aggregate_{act}$

(i) Boundary node prediction using $Sample_{act}, Aggregate_{top}$

(j) Boundary node prediction using $Sample_{act}, Aggregate_{act}$

Figure 4.4: Metric performance for boundary node prediction for news events (N1-N10).



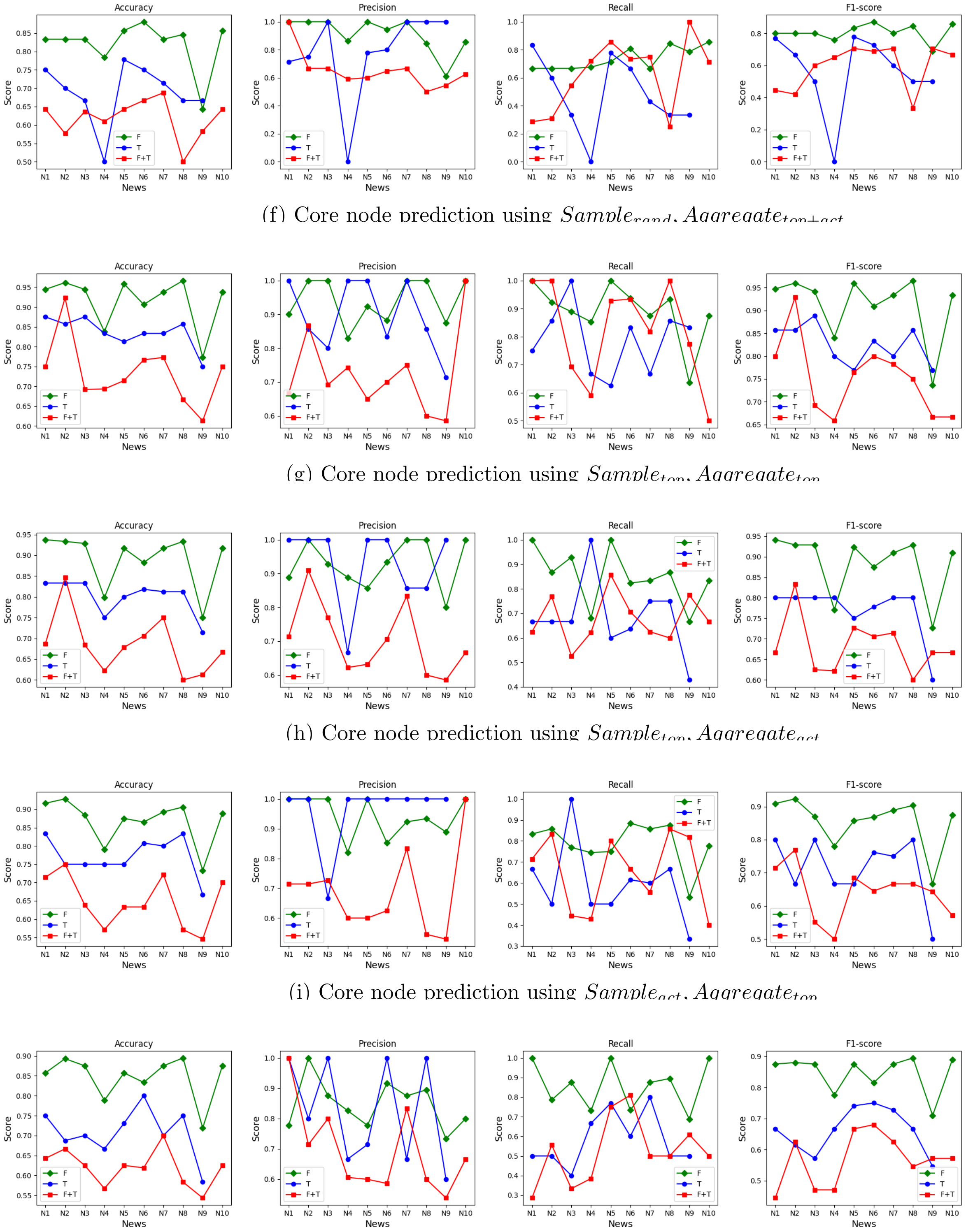
(a) Core node prediction using *Trusting others*

(b) Core node prediction using *Trusted by others*

(c) Core node prediction using *Interpolation*

(d) Core node prediction using $Sample_{rand}, Aggregate_{top}$

(e) Core node prediction using $Sample_{rand}, Aggregate_{act}$



(f) Core node prediction using $Sample_{rand}, Aggregate_{top+act}$

(g) Core node prediction using $Sample_{top}, Aggregate_{top}$

(h) Core node prediction using $Sample_{top}, Aggregate_{act}$

(i) Core node prediction using $Sample_{act}, Aggregate_{top}$

(j) Core node prediction using $Sample_{act}, Aggregate_{act}$

Figure 4.5: Metric performance for core node prediction for news events (N1-N10).

4.9 Conclusions and Future Work

In this chapter, we proposed a novel early false information spreader detection model for communities using GraphSAGE framework. The problem is formulated as early detection model using community health assessment. Using interpersonal trust based properties we could not only identify spreaders with higher accuracy but could also identify a false information spreaders better than a true information spreaders. The key hypotheses we tested is that interpersonal trust plays a more important role in identifying false information spreader than true information spreader. Identified false information spreaders can thus be quarantined and true news spreaders can be promoted. Experimental analysis on twitter data collected for multiple debunked news events showed that topology-based modelling yields better results compared to activity-based modelling. The proposed research can be used to identify people who are likely to become spreaders in real time. It can be applied to viral marketing applications as well.

As part of future work we would want to test our proposed model on higher volume of user timeline activity which could give a better picture of the effectiveness of the activity-based approach. Also we would like to include other proxies of trust such as how long a user has been active on the social media platform, whether the user has a history of refuting false information etc. Also our model has been evaluated on the immediate follower-following network of the information spreaders. We would want to extend the network further in order to sample neighbhood from greater sampling depths.

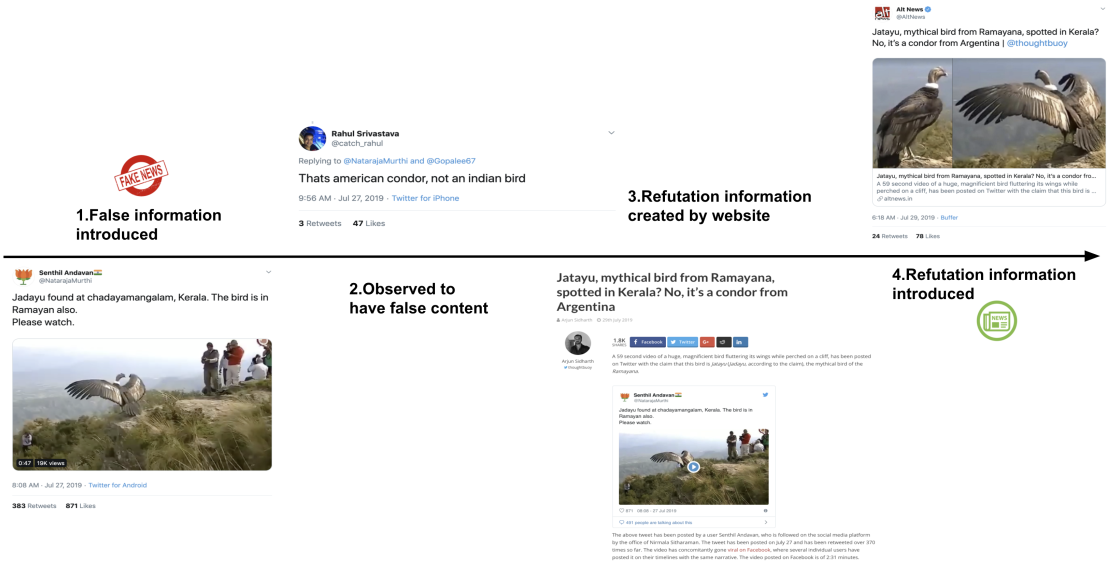
Chapter 5

Infection Control and Prevention

5.1 Introduction

Social network platforms like Twitter, Facebook and Whatsapp are used by millions around the world to share information and opinions. Often, the veracity of content shared on these platforms is not confirmed. This gives rise to scenarios where information having conflicting veracity, i.e. false information and its refutation, co-exist. Refutation can be defined as true information that fact checks a specific item of false information. Content from popular fact checking websites like *altnews.in* can be categorized as refutation information. A typical scenario is that false information originates at time t_1, and starts propagating. Once it is detected, a refutation for it might be created, which starts propagating simultaneously at time t_2 ($t_1 < t_2$). An example scenario in Twitter is shown in Figure 5.1. Many nodes (i.e. people) may lie in the spread paths of both the false information and its refutation. They might either decide to spread false information, its refutation, do nothing, or spread both sets of information, especially if they see the refutation after they have propagated the false information already.

While detecting false information is an important and widely researched problem, an equally important problem is that of reversing the impact of false information spreading. Techniques involve containment/suppression of false information, as well as accelerating the spread of its refutation. An effective strategy in line with such techniques is to proactively identify nodes that are likely to believe false information, and proceed to take appropriate measures. *Being able to predict the likely action of such nodes before they are*


FAKE NEWS
1.False information introduced
2.Observed to have false content
3.Refutation information created by website
4.Refutation information introduced
NEWS

Senthil Andavan
@NatarajaMurthi

Jadayu found at chadayamangalam, Kerala. The bird is in Ramayan also.
Please watch.

0:47 19K views

8:08 AM · Jul 27, 2019 · Twitter for Android

383 Retweets 871 Likes

Rahul Srivastava
@catch_rahul

Replying to @NatarajaMurthi and @Gopalee67

Thats american condor, not an indian bird

9:56 AM · Jul 27, 2019 · Twitter for iPhone

3 Retweets 47 Likes

Jatayu, mythical bird from Ramayana, spotted in Kerala? No, it's a condor from Argentina

Arjun Sidharth 29th July 2019

Arjun Sidharth
thoughtbuoy

1.8K SHARES Facebook Twitter G+ in

A 59 second video of a huge, magnificient bird fluttering its wings while perched on a cliff, has been posted on Twitter with the claim that this bird is *Jatayu* (*Jadayu*, according to the claim), the mythical bird of the *Ramayana*.

Senthil Andavan
@NatarajaMurthi

Jadayu found at chadayamangalam, Kerala. The bird is in Ramayan also.
Please watch.

871 08:08 - 27 Jul 2019

491 people are talking about this

The above tweet has been posted by a user Senthil Andavan, who is followed on the social media platform by the office of Nirmala Sitharaman. The tweet has been posted on July 27 and has been retweeted over 370 times so far. The video has concomitantly gone viral on Facebook, where several individual users have posted it on their timelines with the same narrative. The video posted on Facebook is of 2:31 minutes.

Alt News
@AltNews

Jatayu, mythical bird from Ramayana, spotted in Kerala? No, it's a condor from Argentina | @thoughtbuoy

Jatayu, mythical bird from Ramayana, spotted in Kerala? No, it's a condor fro...
A 59 second video of a huge, magnificient bird fluttering its wings while perched on a cliff, has been posted on Twitter with the claim that this bird is ...
altnews.in

6:18 AM · Jul 29, 2019 · Buffer

24 Retweets 78 Likes

Figure 5.1: Example of co-existing false and refutation information on Twitter.

exposed to false information is an important aspect of such a strategy. Nodes identified as vulnerable to believing false information can thus 1) be cautioned about the presence of the false information so that they do not propagate it, and 2) be urged to propagate its refutation. While optimization models based on information diffusion theories have been proposed in the past for misinformation containment, recent advancements in deep learning on graphs serve as the motivation to explore false information control models using important components that exist even before a specific false information is injected into the network, namely the underlying network structure and people's behavioral data.

Trust and *Credibility* are important psychological and sociological concepts respectively that have subtle differences in their meanings. While trust represents the confidence one person has in another, credibility represents generalized confidence in a person based on his perceived performance record [114]. Thus, trust is a property of a (directed) edge between two nodes, while credibility is a property of an individual node. For Twitter, one can model interpersonal trust using a person's action to follow or retweet someone, while user credibility can be modelled by aggregating credibility of information endorsed by them in the past. Also, as stated by Metzger et al. [87] the interpretation of a neighbor's credibility by a node relies on its perception of the neighbor based on their trust dynamics. Motivated from these ideas we propose a false information spreader detection model based on following two assumptions:

Assumption 1: Based on the property of *Homophily* [115], we hypothesize that credibility of a node in a social network would be the aggregate of its neighbor's credibility.

Assumption 2: Likelihood of a node sharing information endorsed by its neighbor would depend on its perception of the neighbor's credibility, which is a function of their interpersonal trust dynamics.

Most research on controlling false information spreading has focused on using content analysis and propagation paths to identify whether an information is false or not. We propose a complementary approach that analyzes the social network structure and historical behavioral data to identify most likely false information spreaders. More specifically, we propose a graph neural network model that integrates people's credibility and interpersonal trust features in a social network to predict whether a node is likely to spread false information or its refutation.

We make the following contributions in this chapter:

1) We propose a novel user-centric model using a graph neural network with attention mechanism to predict whether a node will most likely spread false information, its refutation or do nothing.

2) We demonstrate that a person's decision to spread a false information is more sensitive to its perception of neighbor's credibility, and this perception is a function of their trust dynamics.

3) To the best of our knowledge, this is the first model being evaluated on not just false and refutation (i.e. true) information, but in a more realistic scenario where false and refutation information co-exists.

The rest of the chapter is organized as follows: First, we discuss the related work. Next in the proposed approach section we explain the graph neural network model with a motivating example. We then explain the trust and credibility features used in the model, followed by detailed analysis of experimental results. Finally, we provide concluding remarks.

5.2 Related Work

Social science research in the past has explored the aspects of people's behavior that cause false information spreading. Jaeger et al. [116] was one of the first to study what makes rumors believable when told by peers instead of authority figures. While it focused on modelling people's anxiety, it served as motivation to explore other sociological features that are relevant to information spreading. Petty and Cacioppo [117] found credibility perception to be an important factor for believing false information. Rosnow et al. [118] proposed that *interpersonal trust* also played an important role in rumor transmission. The idea was further enforced by Morris et al. [119] where they claimed that people assess credibility based on trust relationships with their neighbors in a social network. Motivated by these ideas, there has been much interest in computational models for false information spreader detection using trust, which has shown promising results [113], [51]. Kim's work [120] on applying TSM to study role of trust in rumor suppression is a closely related work. Soroush et al. [40], who conducted an empirical study to understand how people's moods are associated with the propagation of true and false information also served as a motivation for our research.

Many computational techniques to combat false information spreading have been explored over the past decade, as summarized by Sharma et al. [42]. Most models rely on generating relevant features from the information that help distinguish false information from true. Our proposed model is based on recent advances in graph based deep learning research, summarized by Wu et al. [121]. Graph neural network models are a special class of deep learning models that can be applied to non euclidean spaces such as graphs, thus opening up avenues to explore computational models to combat false information spreading by using properties of network structures. Graph Convolution Network (GCN) [122] is a popular graph neural network model that has been successfully applied to domains such as computer vision [105], language modelling [107], recommender systems [109] and transport networks [110]. Another model called Graph Attention Network (GAT) [123] was recently proposed that assigns importance scores to a node's neighbors. It has successfully been applied to recommender systems [124, 125] and network analysis [126] problems. While attention mechanism [127] has been applied to false information detection [128, 16], the approaches were based on content analysis,

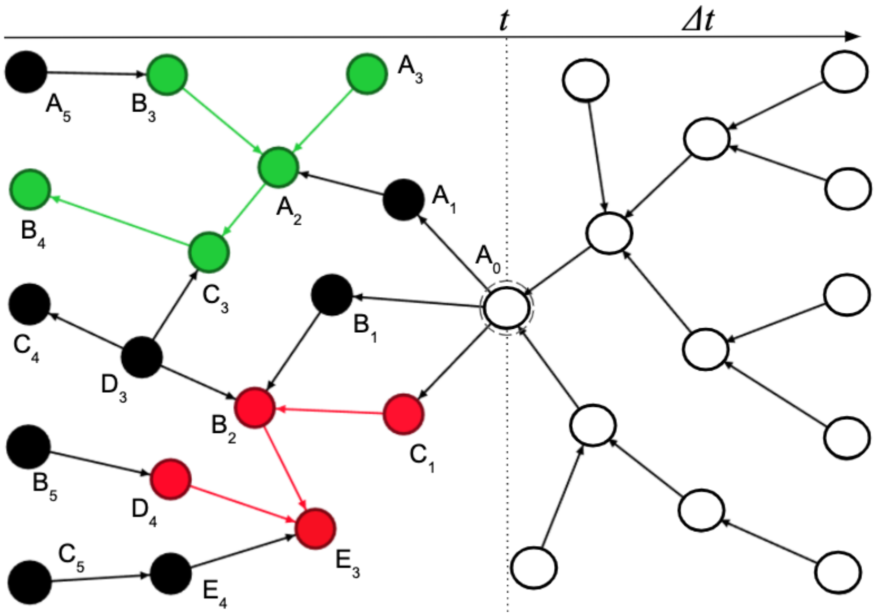


Figure 5.2: Motivating example for spreader prediction in networks where false and refutation information co-exist.

and did not consider underlying network structure. Qui et al.'s [129] is a related work, but while their model focused on influence in general, our model integrates people's psychological and sociological features to identify false information spreaders.

Before false information detection models using machine learning became popular, models inspired by information diffusion models for misinformation containment were proposed. Budak et al. [130] proposed an optimization strategy to identify false information spreaders in a network who, when convinced by its refutation, would minimize the number of people receiving the false information. Nguyen et al. [68] proposed greedy approaches to a similar problem of limiting the spread of false information in social networks. More recently, Tong et al. [131] studied the problem as a multiple cascade diffusion problem. Another closely related problem is rumor blocking that has been studied using two approaches: 1) blocking the spread by removing nodes [132] and 2) blocking the spread by removing edges [133]. Our proposed work is different from these models as we incorporate people's behavior data to predict false information spreaders using a neural network model.

5.3 Interpersonal Trust and User Credibility features

In this section we summarize the trust and credibility features integrated into the proposed model to predict false information spreaders. While these are specific to Twitter, they can be generalized across other social network platforms as well.

5.3.1 Trust-based features

1. Global Trust (Tr^G):

Global trust are trust scores that are computed on the directed follower-followee network around information spreaders. It is called global because an individual's trust score is sensitive to changes in the network structure. Using the Trust in Social Media (TSM) algorithm [50], we quantify the likelihood of *trusting others* and being *trusted by others.* The TSM algorithm assigns a pair of complementary trust scores to each node called *Trustingness* and *Trustworthiness*. *Trustingness (ti)* quantifies the propensity of a node (x) to trust its neighbors, while *Trustworthiness (tw)* quantifies the willingness of the neighbors to trust the node. The TSM algorithm takes a user network — a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ — as input, together with a specified convergence criteria or a maximum permitted number of iterations. In each iteration for every node in the network, trustingness and trustworthiness are computed using the equations below:

$$ti(v) = \sum_{\forall x \in out(v)} \left(\frac{w(v,x)}{1 + (tw(x))^s} \right) \tag{5.1}$$

$$tw(u) = \sum_{\forall x \in in(u)} \left(\frac{w(x,u)}{1 + (ti(x))^s} \right) \tag{5.2}$$

where $u, v, x \in \mathcal{V}$ are nodes, $ti(v)$ and $tw(u)$ are the trustingness and trustworthiness scores of v and u, respectively, $w(v, x)$ is the weight of edge from v to x, $out(v)$ is the set of out-edges of v, $in(u)$ is the set of in-edges of u, and s is the involvement score of the network. The involvement score is basically the potential risk an actor takes when creating a link in the network, which is empirically set to a constant. Details of the algorithm are excluded due to space constraints and can be found in [50].

2. Local Trust (Tr^L):

Local trust is computed based on the retweeting behavior of an individual. It is termed local because the trust score depends solely on an individual node's preferences, and not on the network structure. We consider the proxy for *trusting others* as the fraction of tweets of x that are retweets (RT_x) denoted by $\sum_{\forall i \in t}\{1 \text{ if } i = RT_x \text{ else } 0\}/n(t)$. Meanwhile, we consider the proxy for *trusted by others* as the average number of times x's tweets are retweeted ($n(RT)$) denoted by $\sum_{\forall i \in t} i_{n(RT_x)}/n(t)$. ($t$ represents the most recent tweets posted in x's timeline).

The formulation for global and local trust features is summarized in Table 5.1.

Table 5.1: Summary of interpersonal trust features.

	Tr^G	Tr^L
trusting others	$ti(x)$	$\sum_{\forall i \in t}\{1 \text{ if } i = RT_x \text{ else } 0\}/n(t)$
trusted by others	$tw(x)$	$\sum_{\forall i \in t} i_{n(RT_x)}/n(t)$

5.3.2 Credibility-based features

Credibility for users is generalized based on features extracted from information posted on their timeline, and was empirically studied by Castillo et al. [27]. For Twitter, information includes tweets posted, retweeted and replied to. Borrowing ideas from this work, we generate relevant credibility features for nodes in the network, which can be categorized into two types: User-based and Content-based.

1. User-based Credibility (Cr^U):

User credibility features are extracted from user metadata of nodes in the network. Features used in our model are summarized below:

Table 5.2: Summary of user credibility features.

User-based Credibility (Cr^U)	Content-based credibility (Cr^C)
Registration age	Emotions conveyed in content
Overall activity count	Level of uncertainty
Is Verified	External source citation

A. Registration age (U1): Registration age denotes the time that has transpired since a user created their account. Older accounts tend to be associated with credible users.

B. Overall activity count (U2): Activity or statuses count is the number of tweets issued by a user. Low credibility is associated with users who have not written many messages in the past.

C. Is verified (U3): This label suggests whether a user account is marked as authentic or not by Twitter. Verified accounts are more likely to be credible.

2. Content-based Credibility (Cr^C) :

Content credibility features are obtained by aggregating a user's timeline activity. It is important to note that unlike Castillo's assumption, we do not make a distinction whether information is specifically related to news or not, as that process would involve manually assessing newsworthiness of the tweets. The following relevant features are extracted:

A. Emotions conveyed by user (M1): Emotions represent positive or negative sentiments associated with the tweet. Strong sentiments are usually associated with non-credible users.

B. Level of uncertainty (M2): Level of uncertainty is quantified as the fraction of user's tweets that are questioning in nature (i.e. contain question marks). Tweets with a higher level of uncertainty tend to be less credible.

C. External source citation (M3): External source citation is quantified as the fraction of user's tweets that cite an external URL. Tweets with cited URLs tend to be more credible.

Table 5.2 summarizes the credibility features.

5.4 Proposed Approach

Motivating example

Consider a directed social network as illustrated in Figure 5.2 where false information and its refutation co-exist. Directed edges represent interpersonal trust. For Twitter, directed edge $A_0 \rightarrow B_1$ means that A_0 follows B_1, and thus information flows from B_1

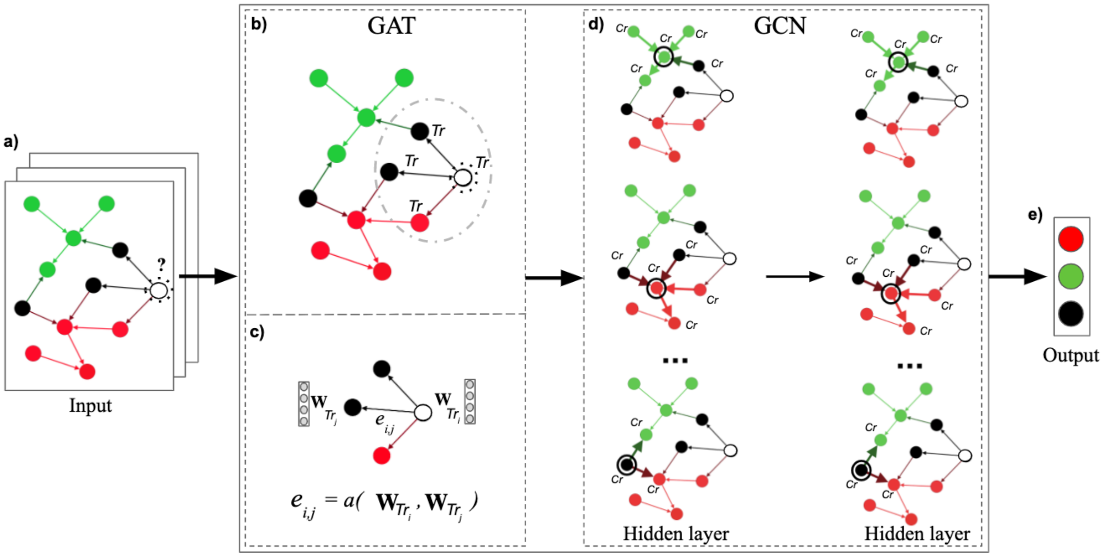


Figure 5.3: Architecture of proposed spreader prediction model in networks where false and refutation information co-exist.

to A_0. If A_0 decides to retweet an information it receives from B_1, it can be interpreted as an endorsement of it. Usually, false information originates at a certain node (E_3, in this case) and gradually spreads through neighboring nodes (D_4, B_2, C_1 sequentially). Soon enough, it is identified and its refutation is created (B_4) and simultaneously begins spreading (C_3, A_2, B_3, A_3 sequentially). Labelled nodes represent all nodes that are exposed to either of these by time t. The circled node (A_0) is at a critical position in the network as many unexposed nodes are prone to believing information it endorses. Thus, it is important to predict whether the node will become a spreader of false information or refutation at a future time $t + \Delta t$. Being able to successfully predict action of such nodes can help us proactively take mitigation measures to restrict further spread of false information.

Based on the assumptions stated earlier, we can say that a person's decision to spread information depends on their perceived credibility in the eyes of neighbors. This section explains how we integrate both credibility and trust features to predict whether a person would likely be a spreader of false information or its refutation; using an attention based graph neural network model. The problem formulation is as follows:

Problem formulation

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a directed social network containing false information spreaders ($\mathcal{V}_F$), refutation spreaders ($\mathcal{V}_R$) and non-spreaders ($\mathcal{V}_{\hat{Sp}}$) at a time instance t ($\{\mathcal{V}_F \cup \mathcal{V}_R \cup \mathcal{V}_{\hat{Sp}}\} \subset \mathcal{V}$). Using global ($Tr^G$) and local ($Tr^L$) trust features ($Tr = Tr^G || Tr^L$), and user-based ($Cr^U$) and content-based ($Cr^C$) credibility features ($Cr = Cr^U || Cr^C$) of node i and its neighborhood nodes (${\mathcal{N}_i}^K$) sampled till depth K, we predict whether i will spread false information, refutation information or do nothing at future time $t + \Delta t$.

To predict whether a node i will be a spreader or not, we propose a graph neural network framework that can be broadly divided into two phases:

1. We assign importance score to neighborhood nodes (${\mathcal{N}_i}^K$) sampled till depth K based on Tr features. This is done using Graph attention mechanism.

2. We learn representations using Graph Convolutional Networks by aggregating Cr features proportional to the importance scores assigned for the neighborhood nodes based on step 1.

An overview of the proposed architecture is shown in Figure 5.3. The following subsections explain the framework in detail.

5.4.1 Importance score using attention

We apply a graph attention mechanism proposed by Veličković et al. [123] which attends over the neighborhood of i and based on their trust features, assigns an importance score to every j ($j \in \mathcal{N}_i$). First, every node is assigned a parameterized weight matrix ($\mathbf{W}$) to perform linear transformation. Then, self-attention is performed using a shared attention mechanism a (a single layer feed-forward neural network) which computes trust-based importance scores. The unnormalized trust score between i,j is represented as:

$$e_{ij} = a(\mathbf{W}_{Tr_i}, \mathbf{W}_{Tr_j}) \tag{5.3}$$

where e_{ij} quantifies j's importance to i in the context of interpersonal trust. We perform masked attention by only considering nodes in $\mathcal{N}_i$. This way we aggregate features based only on the neighborhood's structure. To make the importance scores comparable across all neighbors we normalize them using the softmax function:

$$\alpha_{ij} = softmax(e_{ij}) = \frac{exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} exp(e_{ik})} \tag{5.4}$$

The attention layer a is parameterized by weight vector $\vec{\mathbf{a}}$ and applied using LeakyReLU nonlinearity. Normalized neighborhood edge weights can be represented as:

$$\alpha_{ij} = \frac{exp(\text{LeakyReLU}(\vec{\mathbf{a}}^T[\mathbf{W}_{Tr_i}||\mathbf{W}_{Tr_j}]))}{\sum_{k \in \mathcal{N}_i} exp(\text{LeakyReLU}(\vec{\mathbf{a}}^T[\mathbf{W}_{Tr_i}||\mathbf{W}_{Tr_k}]))} \tag{5.5}$$

α_{ij} thus represents trust between i and j with respect to all nodes in $\mathcal{N}_i$. α_{ij} obtained for the edges is used to create an attention-based adjacency matrix $\hat{A}_{atn} = [\alpha_{ij}]_{|\mathcal{V}| \times |\mathcal{V}|}$ which is used to aggregate credibility features later. We use multi-head attention and concatenate M attention heads running in parallel, enabling us to attend information from M representational subspaces. Further details about muti-head mechanisms can be found in [127].

$$\hat{A}_{atn\prime} = Concat(\hat{A}_{atn_1}, \hat{A}_{atn_1}, \ldots, \hat{A}_{atn_M}) \tag{5.6}$$

5.4.2 Feature aggregation

The Graph Convolution Network by Kipf et al. [122] is a graph neural network model that efficiently aggregates features from a node's neighborhood. It consists of multiple neural network layers where the information propagation between layers can be generalized by equation 5.7. Here H represents the hidden layer and A represents the adjacency matrix representation of the subgraph ($A = \hat{A}_{atn\prime}$). $H^{(0)} = Cr$ and $H^{(L)} = Z$, where Z denotes node-level output during transformation.

$$H^{(l+1)} = f(H^{(l)}, A) \tag{5.7}$$

We implement a Graph Convolution Network with two hidden layers using a propagation rule as explained in [122].

$$H^{(l+1)} = \sigma(\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}H^{(l)}W^{(l)}) \tag{5.8}$$

Here, $\hat{A} = A + I$, where I is the identity matrix of the neighborhood subgraph. This operation ensures that we include self-features during aggregation of neighbor's credibility features. $\hat{D}$ is the diagonal matrix of node degrees for $\hat{A}$, where $\hat{D}_{ii} = \sum_j \hat{A}_{ij}$. $W^{(l)}$ is the layer weight matrix, and σ denotes the activation function. Symmetric normalization of $\hat{D}$ ensures our model is not sensitive to varying scale of the features being aggregated.

5.4.3 Node classification

Using credibility features and network structure for nodes in i's neighborhood, node representations are learnt from the graph using a symmetric adjacency matrix with attention-based edge weights ($\hat{A} = \hat{D}^{-1/2}\hat{A}_{atn}\prime\hat{D}^{-1/2}$). Following forward propagation model is applied:

$$Z = f(X, \hat{A}_{atn}\prime) = softmax(\hat{A}\ \mathrm{ReLU}(\hat{A}XW^{(0)})W^{(1)}) \tag{5.9}$$

X represents the credibility features. $W^{(0)}$ and $W^{(1)}$ are input-to-hidden and hidden-to-output weight matrices respectively, and are learnt using gradient descent learning. Classification is performed using the following cross entropy loss function:

$$\mathcal{L} = \sum_{l \in \mathcal{Y}_L} \sum_{f \in Cr} Y_{lf} ln Z_{lf} \tag{5.10}$$

where $\mathcal{Y}_L$ represents indices of labeled vertices, f represents each of the credibilty features being used in the model, and $Y \in R^{|\mathcal{Y}_L| \times |Cr|}$ is the label indicator matrix.

5.5 Experiment and Results

5.5.1 Data collection

We evaluate our proposed model using real world Twitter datasets. The ground truth of false information and the refuting true information was obtained manually from AltNews[1] and BOOM[2], popular fact checking websites based in India. The source

[1] https://www.altnews.in/

[2] https://www.boomlive.in/

Table 5.3: Co-existing false and refutation information network dataset statistics.

	N1			**N2**			**N3**			**N4**			**N5**		
	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	$\|Sp\|$	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	$\|Sp\|$	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	$\|Sp\|$	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	$\|Sp\|$	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	$\|Sp\|$
F	1,797,059	5,316,114	2,584	885,598	1,824,585	943	1,228,479	2,477,986	1,313	2,607,629	7,146,454	4,552	2,150,820	5,215,120	3,344
T	1,164,162	2,283,160	437	453,537	879,854	403	1,169,681	1,988,576	425	433,616	773,778	467	1,168,820	1,543,513	305
$\mathbf{F} \cup \mathbf{T}$	2,677,924	7,562,503	3,017	1,230,559	2,641,513	1,337	2,198,524	4,458,228	1,738	2,900,925	7,882,019	5,015	3,019,066	6,631,032	3,627
$\mathbf{F} \cap \mathbf{T}$	283,297	8,956	4	108,576	59,912	9	199,636	376	0	140,320	3,273	5	300,574	112,098	22

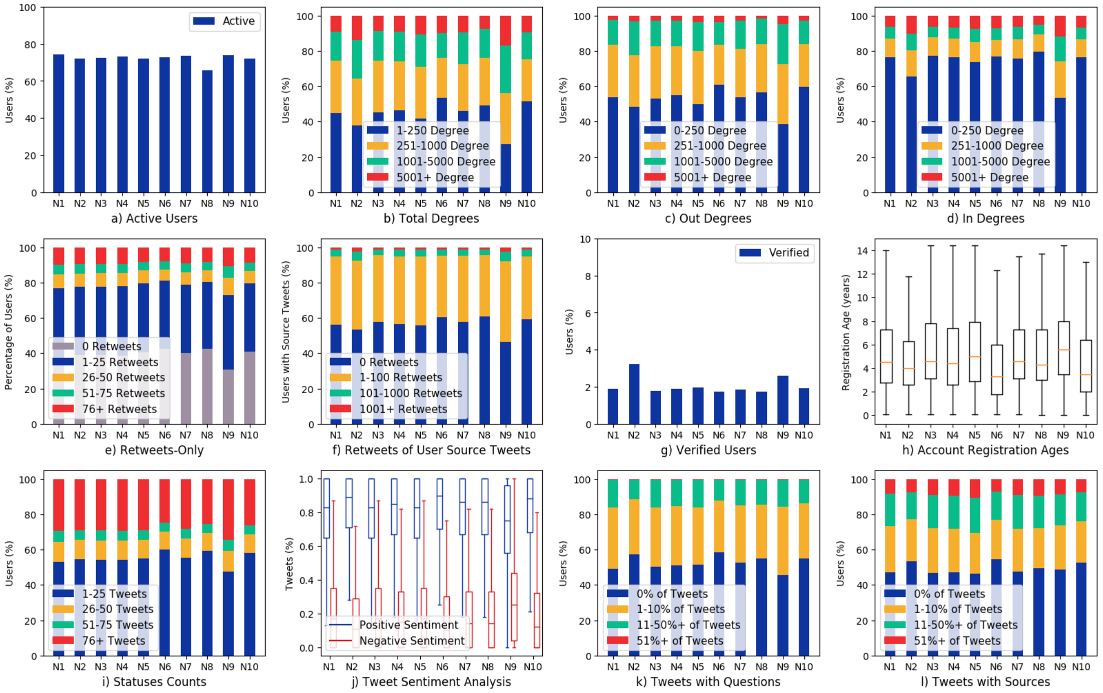
	N6			**N7**			**N8**			**N9**			**N10**		
	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	$\|Sp\|$	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	$\|Sp\|$	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	$\|Sp\|$	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	$\|Sp\|$	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	$\|Sp\|$
F	2,387,610	5,356,288	3,498	627,147	1,071,120	696	2,036,162	2,876,783	894	1,197,935	2,139,912	2,317	2,174,023	4,280,962	2,323
T	1,297,371	1,727,503	481	1,166,528	2,524,907	847	1,058,482	1,513,404	489	2,999,865	6,317,032	1,833	704,006	1,314,996	741
$\mathbf{F} \cup \mathbf{T}$	2,449,434	5,691,728	3,769	1,606,924	3,577,449	1,534	2,663,392	4,082,373	1,365	4,064,545	8,443,888	4,151	2,729,312	5,584,915	3,063
$\mathbf{F} \cap \mathbf{T}$	1,235,547	1,379,510	212	186,751	11,131	9	431,252	305,358	20	133,255	722	1	148,717	699	1

tweet related to the information was obtained directly as a tweet embedded in the website or through a keyword based search on Twitter. We only considered tweets with a large number of retweets (¿ 300). From that source tweet, we used the Twitter API to determine the source tweeter and retweeters (proxy for spreaders), the follower-following network of the spreaders (proxy for social network), and user activity data for all nodes in the network. Besides evaluating our model on the false information (F) and true information (T) spreading networks separately, we also evaluated our model on the combined information spreading networks (F ∪ T). Details regarding the number of nodes ($|\mathcal{V}|$), edges ($|\mathcal{E}|$), and spreaders ($|Sp|$) for the networks of 10 different news events (N1-N10) is shown in Table 5.3.

5.5.2 Feature analysis

Figure 5.4 summarizes trust and credibility based features for the 10 news events. Plot a) shows that over 70% nodes on average were active users (i.e. they had one or more timeline tweets). Plots b), c) and d) shows stacked bar chart for node degrees which follow power law distribution. Plot e) shows the distribution of tweets retweeted by the users. Retweets comprise a low percentage of timeline activity (less than 25% of tweets for over 75% users are retweets). A similar trend is observed in f), with over 95% of users being retweeted less than 100 times. Plot g) shows the percentage of accounts that have been verified by Twitter. The percentages are low (around 2-3% of all accounts) and do not vary much across networks. Plot h) shows the boxplot distribution of user account's registration ages (0-14 years). The majority of accounts are relatively new, created within the past 4 years. Plot i) shows the distribution of statuses counts. We note that almost 50% of all accounts had a count of between 1 and

25 tweets. Additionally, our data showed that only a little under 20% of accounts had 100 or more tweets. Thus, setting the tweet limit to 100 did not significantly affect the quality of the data collected. Thus the quantified features must be close to a realistic estimate. Plot j) shows sentiment analysis results, done using [134] that classified tweets as having either a positive or a negative sentiment. The boxplots show that tweets of most users in all networks have more positive sentiment than negative sentiment. Plot k) shows the distribution of the percentage of a user's tweets that contained a question. The data follows a power law distribution (very few users with > 51% tweets). The stacked bar chart representation shows that around 50% of all users had tweets that contain question. Meanwhile, 30% of users had only 1-10% of all tweets with questions, while nearly 20% of users had a substantial number of questioning tweets (> 10% of all tweets). Plot l) shows the percentage of a user's tweets containing external source URLs. The data has a power law distribution similar to the questioning tweets data.



a) Active Users b) Total Degrees c) Out Degrees d) In Degrees

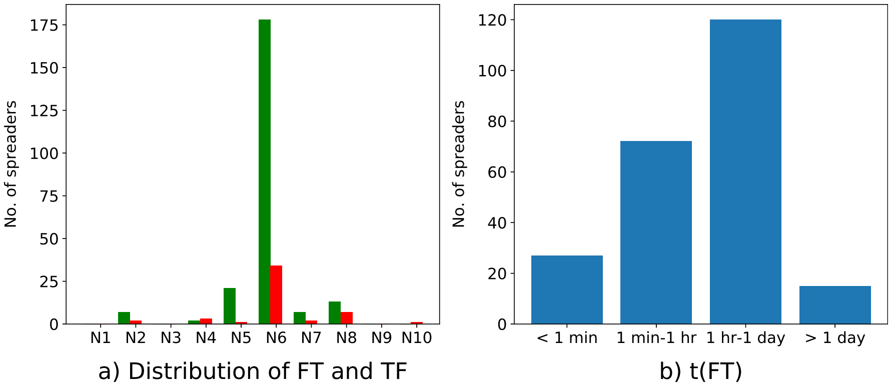
e) Retweets-Only f) Retweets of User Source Tweets g) Verified Users h) Account Registration Ages

i) Statuses Counts j) Tweet Sentiment Analysis k) Tweets with Questions l) Tweets with Sources

Figure 5.4: Credibility and trust feature data for networks of news events (N1-N10).

5.5.3 Analysis of F ∩ T

F ∩ T in Table 5.3 denotes the section of the network that was exposed to both the false and refuting information. An interesting observation is the spreaders who decided to spread both types of information. Figure 5.5 (a) denotes the distribution of spreaders in F ∩ T who spread false information followed by its refutation (FT) and those whose spread refutation followed by the false information (TF). N1 and N9 is excluded from the analysis as our dataset did not have the spreaders' timestamp information. An interesting observation is that the majority of spreaders belong to FT. Intuitively, these are spreaders who trusted the endorser without verifying the information and later corrected their position, thereby implying that they did not intentionally want to spread false information. Consequently, the proposed model can help identify such people proactively in order to take measures to prevent them from endorsing false information. While spreaders belonging to TF are comparatively fewer — whose intentions are not certain — the proposed model can help identify them and effective containment strategies can be adopted. Figure 5.5 (b) shows the time that transpired between spreading refutation and false information for FT spreaders. Once the false information is endorsed, large portions of the network must have already been exposed to false information before the endorser corrected themselves after a significant amount of time (¿ 1 hour). This serves as a strong motivation to have a spreader prediction model which proactively identifies likely future spreaders.



a) Distribution of FT and TF

b) t(FT)

Figure 5.5: Analysis of spreaders in F ∩ T.

5.5.4 Models and metrics

We compare our proposed attention based model with 10 baseline models. Among the baselines, 3 models use node features only (SVM_{Tr}, SVM_{Cr}, $SVM_{Tr,Cr}$), 1 model uses network structure only ($LINE$) and 6 models integrate both node features and the network structure ($GraphSAGE_{Tr}$, $GraphSAGE_{Cr}$, $GraphSAGE_{Tr,Cr}$, GCN_{Tr}, GCN_{Cr}, $GCN_{Tr,Cr}$).

1. Node feature-based models:

A. SVM_{Tr}: This model applies Support Vector Machines (SVM) [135] on nodes' trust based features Tr to find an optimal classification threshold.

B. SVM_{Cr}: This model applies SVM on nodes' credibility based features Cr.

C. $SVM_{Tr,Cr}$: This model applies SVM by combining node's trust based and credibility based features.

2. Network structure-based models:

D. $LINE$: Applies the Large-scale Information Network Embedding [81] as a transductive representation learning baseline, where node embeddings are generated after optimization is performed on the entire graph structure.

3. Network structure + node feature-based models:

E. $GraphSAGE_{Tr}$: GraphSAGE [100] serves as the inductive representation learning baseline where node embeddings are generated by aggregating Tr features from a node's neighborhood.

F. $GraphSAGE_{Cr}$: This inductive representation learning baseline generates node embeddings by aggregating Cr features from a node's neighborhood.

G. $GraphSAGE_{Tr,Cr}$: This inductive representation learning baseline generates node embeddings by aggregating both Tr and Cr features from a node's neighborhood.

H. GCN_{Tr}: This model applies Graph Convolution Networks [122] — an effective variant of CNNs for graph data — and learns node embeddings by aggregating Tr features from a node's neighborhood.

I. GCN_{Cr}: This model applies Graph Convolution Networks by aggregating Cr features from a node's neighborhood.

J. $GCN_{Tr,Cr}$: This model applies Graph Convolution Networks by aggregating both Tr and Cr features from a node's neighborhood.

$GAT_{Tr}GCN_{Cr}$ is the proposed model in this chapter, which aggregates a node neighborhood's Cr features based on attention based importance scores assigned using Tr.

For evaluation, we did a 70-15-15 train-validation-test split of the dataset. We used 5-fold cross validation and four common metrics: Accuracy, Precision, Recall, and F1 score. We report precision, recall and F1 scores of false information spreaders class (-) for F and F ∪ T networks and refutation spreaders class (+) for T network.

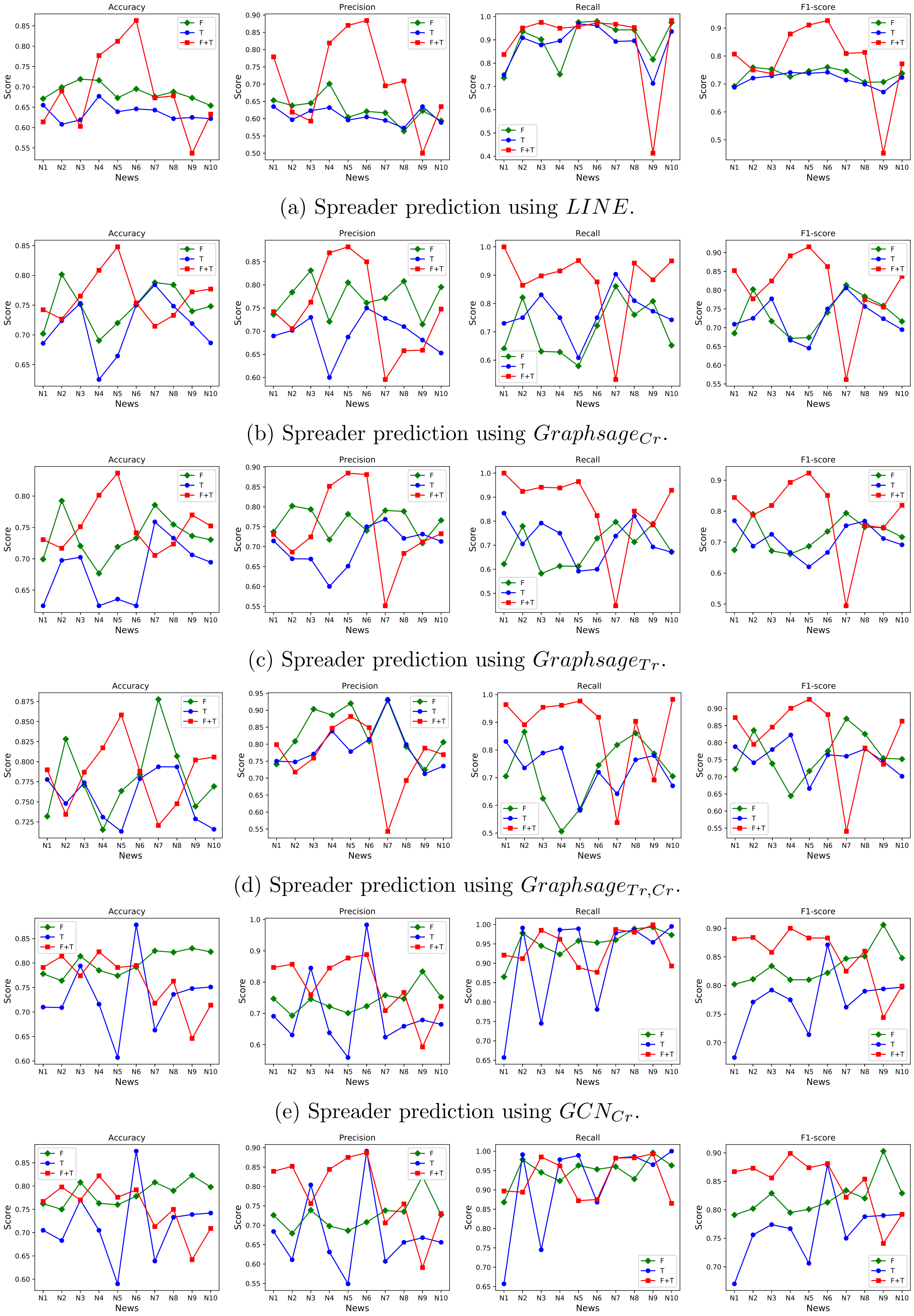
5.5.5 Implementation details

We obtained Tr^G features by running the TSM algorithm on the follower-following network of the spreaders. We used the generic settings for TSM parameters (number of iterations = 100, involvement score = 0.391) based on [50]. Follower-following networks around spreaders and Tr^L, Cr^U, Cr^C features from timeline data of all nodes was collected using the Twitter API. The features were then aggregated using the 100 most recent tweets from each user's timeline. The size of sampled neighborhood was set to 50 and depth was set to 1. We considered neighbors with higher degrees in order to generate denser adjacency matrices. We used a GAT-GCN model with two hidden layers. The number of epochs was set to 200 and the batch size was set to 64. The learning rate was set to 0.001 and the dropout rate was set to 0.2. Parameters were trained using an AdaGrad optimizer with a weight decay of $5e^{-4}$. We used a single attention head. The model was implemented using PyTorch. Code implementation is made available[3].

Table 5.4: Model performance evaluation of false information spreaders (-) and refutation information spreaders (+).

	F				**T**				**F ∪ T**			
	Accu.	**Prec(-)**	**Rec(-)**	**F1(-)**	**Accu.**	**Prec(+)**	**Rec(+)**	**F1(+)**	**Accu.**	**Prec(-)**	**Rec(-)**	**F1(-)**
SVM_{Tr}	0.497	0.512	0.468	0.478	0.473	0.472	0.452	0.445	0.398	0.19	0.465	0.229
SVM_{Cr}	0.508	0.517	0.517	0.509	0.501	0.477	0.565	0.509	0.408	0.196	0.542	0.272
$SVM_{Tr,Cr}$	0.516	0.514	0.579	0.53	0.52	0.513	0.598	0.545	0.444	0.193	0.489	0.267
$LINE$	0.686	0.626	0.896	0.733	0.635	0.608	0.881	0.717	0.688	0.71	0.896	0.786
$GraphSAGE_{Tr}$	0.734	0.762	0.691	0.722	0.680	0.698	0.719	0.705	0.752	0.743	0.859	0.793
$GraphSAGE_{Cr}$	0.747	0.772	0.710	0.736	0.714	0.692	0.764	0.725	0.764	0.747	0.881	0.805
$GraphSAGE_{Tr,Cr}$	0.779	0.831	0.720	0.763	**0.755**	**0.787**	0.732	0.755	0.785	0.764	0.878	0.814
GCN_{Tr}	0.784	0.726	0.947	0.821	0.718	0.675	0.916	0.767	0.753	0.783	0.930	0.845
GCN_{Cr}	0.800	0.742	0.953	0.834	0.731	0.697	0.906	0.773	0.762	0.786	0.940	0.851
$GCN_{Tr,Cr}$	0.824	0.774	0.942	0.848	0.743	0.702	0.916	0.783	0.776	**0.788**	0.954	0.861
$GAT_{Tr}GCN_{Cr}$	**0.876**	**0.834**	**0.966**	**0.893**	0.734	0.674	**0.981**	**0.794**	**0.789**	0.785	**0.972**	**0.866**

[3] https://github.com/BhavtoshRath/GAT-GCN-SpreaderPrediction



(a) Spreader prediction using *LINE*.

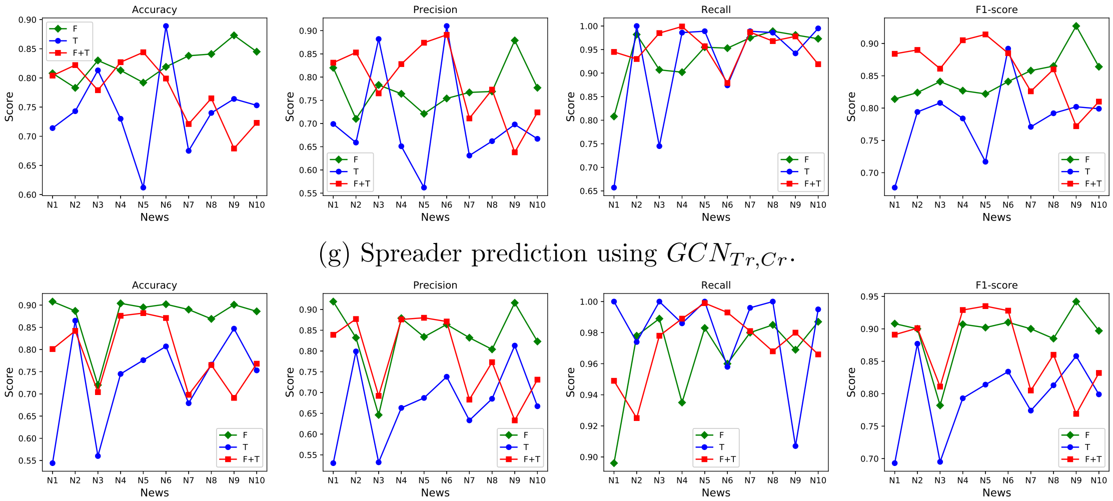
(b) Spreader prediction using $Graphsage_{Cr}$.

(c) Spreader prediction using $Graphsage_{Tr}$.

(d) Spreader prediction using $Graphsage_{Tr,Cr}$.

(e) Spreader prediction using GCN_{Cr}.

(f) Spreader prediction using GCN_{Tr}.



(g) Spreader prediction using $GCN_{Tr,Cr}$.

(h) Spreader prediction using $GAT_{Tr}GCN_{Cr}$.

Figure 5.6: Metric performance for spreader prediction for news events (N1-N10).

5.6 Analysis of Results

5.6.1 Performance evaluation

Classification results of the baselines and proposed model are summarized in Table 5.4. The results shown are averaged over the 10 news events. We implement the model as a two-class (spreader/ non-spreader) classification problem on F and T news networks and as three-class (false information spreader/ refutation spreader/ non-spreader) classification problem on F ∪ T network. We observe that structure only baseline performs better than feature only baselines, and baselines that combine both node features and network structure perform the best. Additionally, we observe that Cr features perform better than Tr features (because there are more number of Cr features than Tr features) and the model performance increases when we use Tr and Cr features together. Node features-based baselines have notably poor performance on the false information spreader class in F ∪ T network. This is because these models do not capture the structural knowledge from the node's neighborhood. $LINE$ — which generates node embeddings only based on the network structure — performs better than node feature-based baselines by a substantial margin, suggesting that network structure is relatively more important than node features in identifying false information spreaders. In terms of accuracy, the $LINE$ model shows an increase of 32.9%, 22.1% and 54.9% for F, T and F ∪ T networks, respectively, over $SVM_{Tr,Cr}$. Graph neural network-based baselines that combine both network structure and node features show a significant increase in all metrics for the three network types. Among $GraphSAGE$ models, $GraphSAGE_{Tr,Cr}$ has the best performance, with an increase in accuracy of 13.6%, 18.9% and 14.1% for F, T and F ∪ T networks, respectively, over $LINE$. Similarly, $GCN_{Tr,Cr}$ has the best performance among all GCN models, with an increase in accuracy of 20.1%, 17% and 12.8% for F, T and F ∪ T networks, respectively, over $LINE$. GCN models perform better than $GraphSAGE$ models on all metrics for F networks, while that is not the case for T and F ∪ T networks. This is because Tr and Cr features for refutation information spreaders and non-spreaders do not differ from each other, which reinforces our hypothesis that trust and credibility features are important to identify people likely to spread false information. $GAT_{Tr}GCN_{Cr}$ — the proposed model — shows an increase in performance for all three networks. However, $GraphSAGE_{Tr,Cr}$ shows better accuracy and precision on T networks because the specific news events on which it performed better involved religious tones, and so decisions to refute them were more sensitive to neighborhood's Cr than Tr. Precision on F ∪ T networks is highest for $GCN_{Tr,Cr}$, though it is still comparable to the proposed model's performance. More importantly, in F ∪ T network — where false and its refutation information co-exist — we observe highest accuracy and F1 scores of 78.9% and 86.6% in the proposed model, thus supporting our hypothesis that false information spreading is more sensitive to trust and credibility thus, making their inclusion in false information spreader prediction models very important. Plots of the evaluation metric of 10 news events for all models, except SVM is shown in Figure 5.6.

5.6.2 Parameter sensitivity analysis

Figure 5.7 shows sensitivity of F1 scores of the proposed $GAT_{Tr}GCN_{Cr}$ model on two important parameters: the size of neighborhoods whose features are aggregated (Neighbors), and the number of recent timeline tweets from which features are aggregated (Tweets).

Neighbors: We evaluated our model on n-neighbors, where n = 10, 20, 30, 40, 50. Plots a), c), and e) show results on F, T and F ∪ T networks, respectively. We observe that model performance is not very sensitive to varying neighborhood size, which could be attributed to the fact that since we have only the immediate follower-following network (sampling depth=1) we are not able to entirely capture meaningful dynamics (i.e. the decision to retweet probably does not depend only on the immediate neighbors).

Tweets: We also evaluated our model on the n-most recent timeline tweets, where n = 20, 40, 60, 80, 100. Plots b), d), and f) shows results on F, T and F $\cup$ T networks, respectively. We observe that for all three networks, prediction performance tends to linearly increase as the number of timeline tweets used to aggregate features increases. Thus, the model performance is more sensitive to the number of tweets than to neighborhood size. This is probably because using more behavioral data helps us estimate trust and credibility features better.

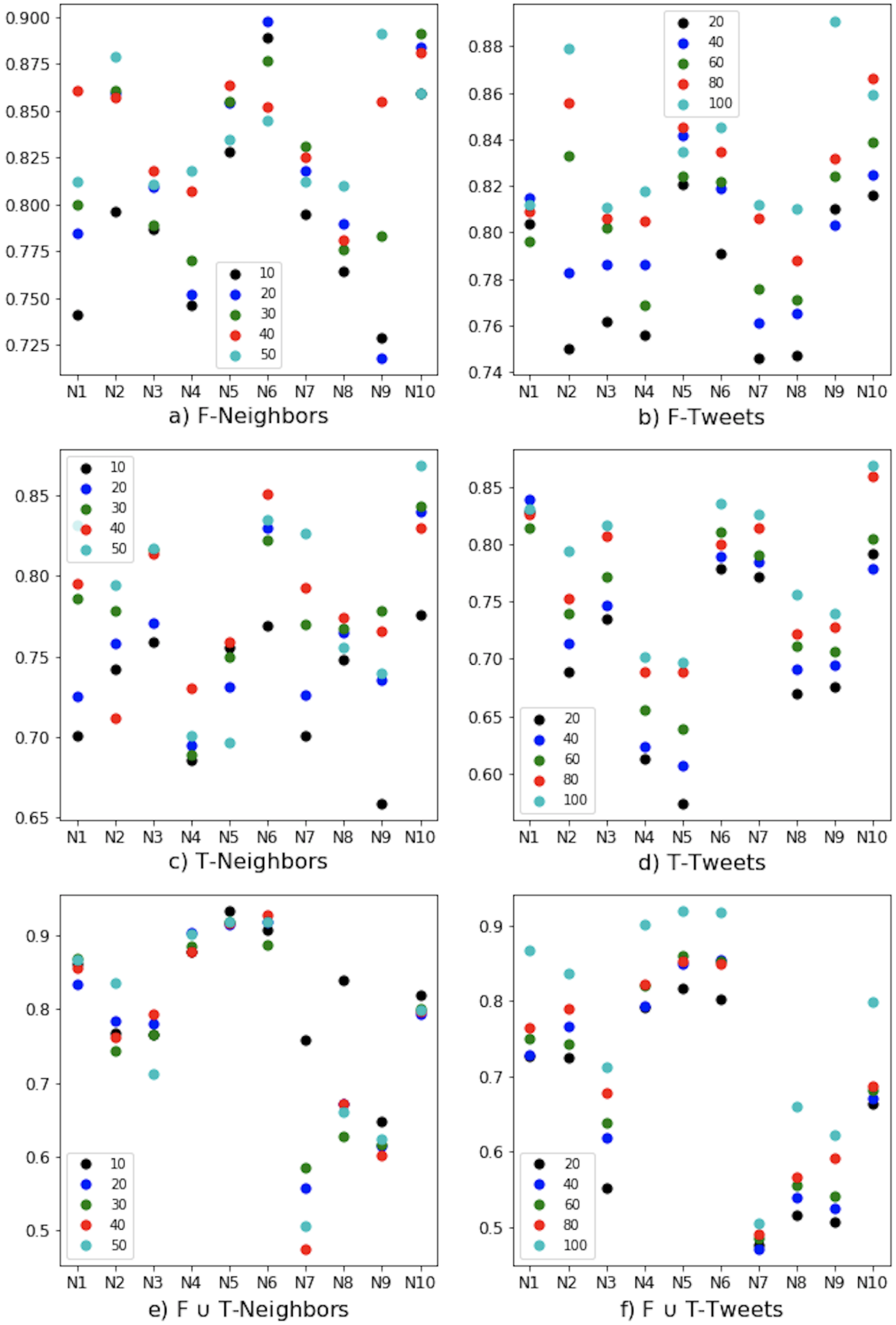
5.6.3 Explainability analysis of trust and credibility

Figure 5.8 shows importance scores that false and refutation spreader's neighbors (size=10) assign each other based on trust dynamics and the neighbor's credibility score (euclidean norm). We show examples for neighbors with both high and low modularity. We observe that refutation spreader's neighbors have higher credibility than false spreader's neighbors because of network homophily. The magnitude of importance scores suggest that false spreader's neighbors trust each other less compared to refutation spreader's neighbors. Node 0 is the neighbor that the spreader endorses. We observe than false spreader endorses the neighbor having strong trust dynamics with its neighbors while that need not be the case for refutation spreaders neighbor. This is because the decision to endorse depends on information source viz. a fact checker.

5.7 Conclusions and Future work

In this chapter we proposed a graph neural network model to predict whether a node in a social network is likely to spread false information or not. The model learns node embeddings by first assigning interpersonal trust-based importance scores using attention mechanism and then aggregating its neighborhood's credibility features proportionally. What makes this model different from most existing research is that a) it proposes a more spreader-centric modelling approach for comabiting false information instead of a content-centric approach, and b) it does not rely on features extracted from the information itself. Our model uses two important components that do not rely on the presence of false information: underlying network structure and people's historical behavioral data. Thus, it can be used to predict people susceptible to false information spreading, even before its spreading has actually begun.

As part of future work we would like to analyze our model on more news events comprising larger networks. We would also want to consider incorporating bot detection techniques to exclude any bots from our network analysis. Additionally, since our model has been evaluated only on the immediate follower-following network of the information spreaders, we would want to extend the network further in order to sample neighborhoods from greater sampling depths.



a) F-Neighbors

b) F-Tweets

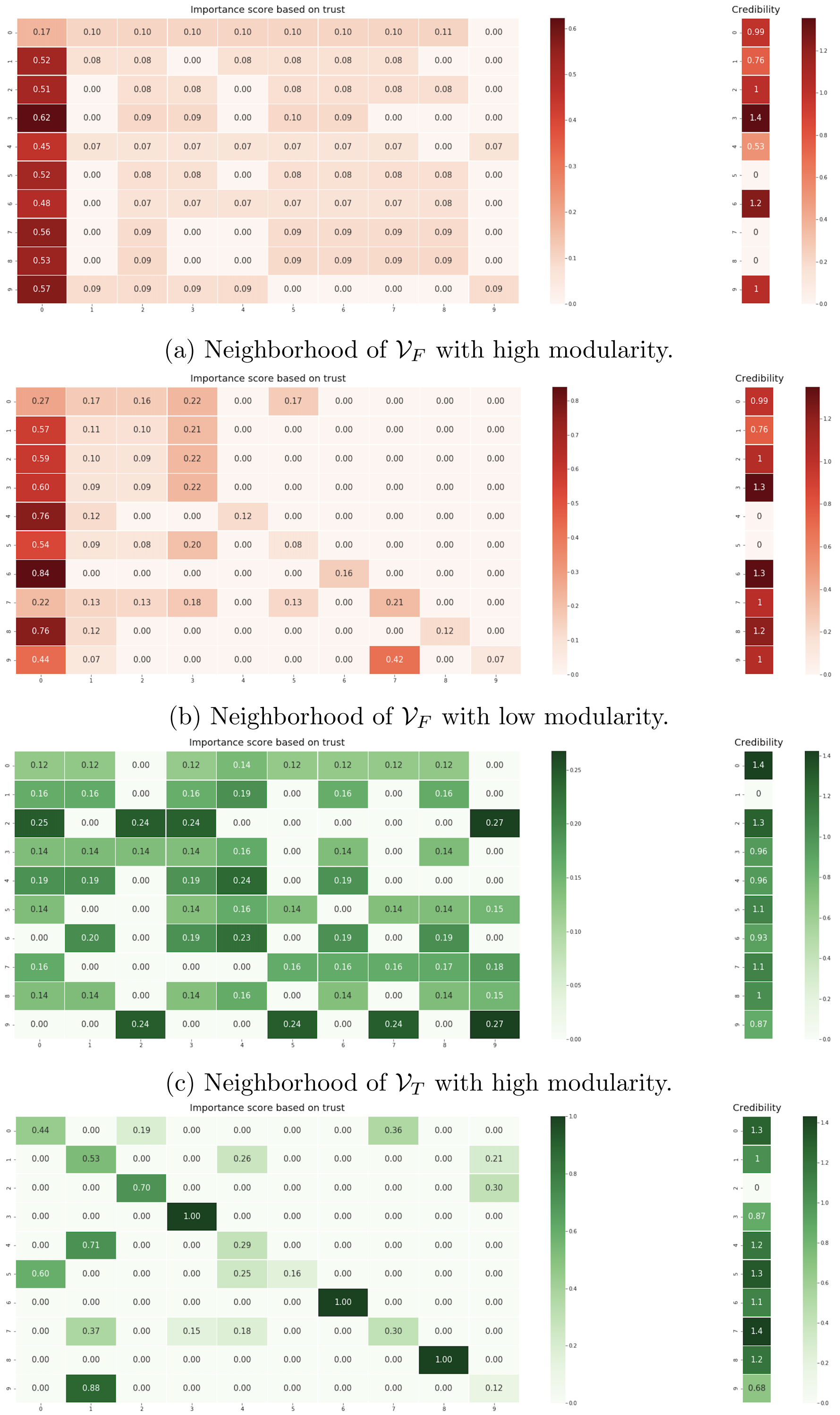
c) T-Neighbors

d) T-Tweets

e) F ∪ T-Neighbors

f) F ∪ T-Tweets

Figure 5.7: Parameter sensitivity analysis.



	0	1	2	3	4	5	6	7	8	9	Credibility
0	0.17	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.11	0.00	0.99
1	0.52	0.08	0.08	0.00	0.08	0.08	0.08	0.08	0.00	0.00	0.76
2	0.51	0.00	0.08	0.08	0.00	0.08	0.08	0.08	0.08	0.00	1
3	0.62	0.00	0.09	0.09	0.00	0.10	0.09	0.00	0.00	0.00	1.4
4	0.45	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.00	0.07	0.53
5	0.52	0.00	0.08	0.08	0.00	0.08	0.08	0.08	0.08	0.00	0
6	0.48	0.00	0.07	0.07	0.07	0.07	0.07	0.07	0.08	0.00	1.2
7	0.56	0.00	0.09	0.00	0.00	0.09	0.09	0.09	0.09	0.00	0
8	0.53	0.00	0.09	0.00	0.00	0.09	0.09	0.09	0.09	0.00	0
9	0.57	0.09	0.09	0.09	0.09	0.00	0.00	0.00	0.00	0.09	1

(a) Neighborhood of $\mathcal{V}_F$ with high modularity.

	0	1	2	3	4	5	6	7	8	9	Credibility
0	0.27	0.17	0.16	0.22	0.00	0.17	0.00	0.00	0.00	0.00	0.99
1	0.57	0.11	0.10	0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.76
2	0.59	0.10	0.09	0.22	0.00	0.00	0.00	0.00	0.00	0.00	1
3	0.60	0.09	0.09	0.22	0.00	0.00	0.00	0.00	0.00	0.00	1.3
4	0.76	0.12	0.00	0.00	0.12	0.00	0.00	0.00	0.00	0.00	0
5	0.54	0.09	0.08	0.20	0.00	0.08	0.00	0.00	0.00	0.00	0
6	0.84	0.00	0.00	0.00	0.00	0.00	0.16	0.00	0.00	0.00	1.3
7	0.22	0.13	0.13	0.18	0.00	0.13	0.00	0.21	0.00	0.00	1
8	0.76	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.00	1.2
9	0.44	0.07	0.00	0.00	0.00	0.00	0.00	0.42	0.00	0.07	1

(b) Neighborhood of $\mathcal{V}_F$ with low modularity.

	0	1	2	3	4	5	6	7	8	9	Credibility
0	0.12	0.12	0.00	0.12	0.14	0.12	0.12	0.12	0.12	0.00	1.4
1	0.16	0.16	0.00	0.16	0.19	0.00	0.16	0.00	0.16	0.00	0
2	0.25	0.00	0.24	0.24	0.00	0.00	0.00	0.00	0.00	0.27	1.3
3	0.14	0.14	0.14	0.14	0.16	0.00	0.14	0.00	0.14	0.00	0.96
4	0.19	0.19	0.00	0.19	0.24	0.00	0.19	0.00	0.00	0.00	0.96
5	0.14	0.00	0.00	0.14	0.16	0.14	0.00	0.14	0.14	0.15	1.1
6	0.00	0.20	0.00	0.19	0.23	0.00	0.19	0.00	0.19	0.00	0.93
7	0.16	0.00	0.00	0.00	0.00	0.16	0.16	0.16	0.17	0.18	1.1
8	0.14	0.14	0.00	0.14	0.16	0.00	0.14	0.00	0.14	0.15	1
9	0.00	0.00	0.24	0.00	0.00	0.24	0.00	0.24	0.00	0.27	0.87

(c) Neighborhood of $\mathcal{V}_T$ with high modularity.

	0	1	2	3	4	5	6	7	8	9	Credibility
0	0.44	0.00	0.19	0.00	0.00	0.00	0.00	0.36	0.00	0.00	1.3
1	0.00	0.53	0.00	0.00	0.26	0.00	0.00	0.00	0.00	0.21	1
2	0.00	0.00	0.70	0.00	0.00	0.00	0.00	0.00	0.00	0.30	0
3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.87
4	0.00	0.71	0.00	0.00	0.29	0.00	0.00	0.00	0.00	0.00	1.2
5	0.60	0.00	0.00	0.00	0.25	0.16	0.00	0.00	0.00	0.00	1.3
6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.1
7	0.00	0.37	0.00	0.15	0.18	0.00	0.00	0.30	0.00	0.00	1.4
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	1.2
9	0.00	0.88	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.68

(d) Neighborhood of $\mathcal{V}_T$ with low modularity.

Figure 5.8: Explainability analysis of trust and credibility for spreader's neighbors.

Chapter 6

Conclusions and Discussions

6.1 Conclusions

We are spending an increasing amount of time on social media interacting with others and in the process consuming and sharing information. As of early 2020, almost half of the world population is active on various social media platforms[1], and with technology becoming easily accessible this trend is only on the rise. With major social media platforms like Twitter becoming the 'go-to places' for consuming news, where practically all the information is user-generated content and whose veracity is not guaranteed, society is facing the problem of false information. Countries across the world have identified the danger it poses to national security. Thus there has been significant interest in developing computational models to handle this problem, with majority of focus on determining the veracity of the information and content analysis. Our proposed approach is complementary to it. We propose a novel spreader-centric false information detection and control model analyzing social network structure, inspired by the domain of Epidemiology [136]. The thesis proposes four sequential spreader-centric phases for false information mitigation inspired from the domain of epidemiology:

I. **Vulnerability Assessment** [51]: In this phase we proposed the *Community Health Assessment* model novel metrics to quantify the vulnerability of nodes and communities to false information spreading for the scenario when false information

[1] https://www.oberlo.com/blog/social-media-marketing-statistics

spreading has not begun. In the context of epidemiology, this would help assess the vulnerability of people before infection spreading begins.

II. **Identification of infected population** [137]: In this phase we proposed a node embedding based recurrent neural network model to identify false information spreaders and true information for the scenario when information spreading has finished. In the context of epidemiology, this would help identify infected population after the infection spreading is complete.

III. **Risk assessment of population** [53]: In this phase we proposed a model that applies a graph neural network model that samples and aggregates features to identify false information spreaders and non-spreaders for the scenario when information is in the course of spreading. In the context of epidemiology, this would be equivalent to identifying the exposed population that needs to be quarantined to prevent infected spreading.

IV. **Infection control and prevention**: In this phase we proposed a model that applies an attention based graph neural network model to identify information spreaders for the scenario when false and its refutation information co-exist. In the context of epidemiology, this would help in targeting people with refutation information to a) change the role of a false information spreader into a true information spreader (i.e. as an antidote) and b) prevent further spread of false information (i.e. as a vaccine).

What makes this research different from most existing research is that a) it proposes a more spreader-centric modelling approach instead of content-centric approach, and b) it does not rely on features extracted from false information thus serving as motivation to build false information mitigation strategies, even for the scenario when false information has not yet originated. The research has shown encouraging results, and thus serves as motivation to pursue the idea further. It is also worth noting that while the epidemiology framework is proposed for false information prevention, it can also be generalized for mining insights for applications like cyberbullying, abuse detection, viral marketing, etc.

In addition to false information mitigation research, we have applied the concept of Computational Trust working in collaboration with researchers from Journalism and

Mass Communication department. Two problems where we have made contributions to include proposing a seeding strategy for viral advertising on social networks [138] and studying the impact of news organizations' trustworthiness and social media activity on audience engagement [137].

6.2 Research Limitations

As any research work, the proposed research models also have certain limitations due to limited data availability (mainly due to time and Twitter API rate limits) that could have been addressed had Twitter data been more easily accessible (either through collaborations or working at Twitter). Also the models are evaluated on the immediate follower-following network of the information spreaders, i.e. depth $= 1$ as the network. Having network data with depth > 1 will help further improve the performance of the models. The work also does not distinguish active users from social bots or inactive users, as they constitute a very small percentage of the network.

The proposed models are based on the assumption that the action of retweeting (without commenting) is a proxy of believing. The research does not consider retweeting (with commenting), replying with affirmation or the action of liking as proxies of trust mainly due to API limitations. Also the quantification of *trustworthiness* in literature is usually based on a person's reliability based on behavioral indicators [139]. But quantifying the authenticity of past activity is beyond the scope of this research work.

Epidemiology in the infection spreading context is not just concerned with infection spreaders but also people who become ill or die. This distinction is not made in our false information spreading model. While epidemiology models are infection specific, our research considers all kinds of false information as one kind of infection i.e. we do not make distinction between types of false information. Also since we consider action of retweeting as the proxy of infection spreading (i.e. false information believers), our model cannot include spreaders who believe the false information but instead of retweeting and spreading on Twitter (platform specific), decide to either spread offline or on another social networking platform.

Computational models for non-vaccine based epidemiological solutions (washing hands, boiling water) for contact based spreading solutions have not been explored, while solutions such as wearing mask or social distancing can be considered analogous to believing whether a person in the vicinity is infected or not (in information spreading context this would be equivalent to believing a neighbor's information, that our framework models).

6.3 Future Work

Future work on the proposed models can incorporate network after eliminating bots and inactive users to better model psychological and sociological properties based on behavioral data. Volume of behavioral data can be further increased to learn better features. In addition to trust and credibility, other important psychological and sociological proxies can also be incorporated to distinguish false information spreaders from the rest. Since our models has been evaluated only on the immediate follower-following network of the information spreaders, the network size can be further increased in order to sample neighborhoods from greater sampling depths.

Since we are proposing a content-agnostic framework, our models do not factor *interestingness* of the information (whether it is carefully manipulated around a popular news topic) or the information endorser (whether the person shares information that is usually considered interesting and is likely to spark activity among other social media users). Integration of models from the domain of natural language processing and sequential data analysis on social networks to capture knowledge from content of people that comprise the social network can be a research extension.

Also the proposed Community Health Assessment model has only scratched the surface of false information mitigation research using a community structure perspective. Studying the dynamics of information spreading within communities has immense scope and needs further work. Also, since our framework considers all kind of false information as one kind of infection, in future there can be different frameworks based on the type of false information (i.e. political, satire, hoax etc).

References

[1] Mark Johnson. Pcfg models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632, 1998.

[2] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. Finding deceptive opinion spam by any stretch of the imagination. *arXiv preprint arXiv:1107.4557*, 2011.

[3] Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. Social spammer detection with sentiment information. In *2014 IEEE International Conference on Data Mining*, pages 180–189. IEEE, 2014.

[4] Benjamin D Horne and Sibel Adali. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[5] Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, and Anupam Joshi. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In *Proceedings of the 22nd international conference on World Wide Web*, pages 729–736, 2013.

[6] Kai Shu, Suhang Wang, and Huan Liu. Exploiting tri-relationship for fake news detection. *arXiv preprint arXiv:1712.07709*, 8, 2017.

[7] Svitlana Volkova and Jin Yea Jang. Misleading or falsification: Inferring deceptive strategies and types in online news and social media. In *Companion Proceedings of the The Web Conference 2018*, pages 575–583, 2018.

[8] Jun Ito, Jing Song, Hiroyuki Toda, Yoshimasa Koike, and Satoshi Oyama. Assessment of tweet credibility with lda features. In *Proceedings of the 24th International Conference on World Wide Web*, pages 953–958, 2015.

[9] Yunfei Long. Fake news detection through multi-perspective speaker profiles. Association for Computational Linguistics, 2017.

[10] Zhiwei Jin, Juan Cao, Han Guo, Yongdong Zhang, and Jiebo Luo. Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 795–816, 2017.

[11] Lizhao Li, Guoyong Cai, and Nannan Chen. A rumor events detection method based on deep bidirectional gru neural network. In *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, pages 755–759. IEEE, 2018.

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

[13] Yang Liu and Yi-Fang Brook Wu. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[14] Jiawei Zhang, Bowen Dong, and S Yu Philip. Fakedetector: Effective fake news detection with deep diffusive neural network. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1826–1829. IEEE, 2020.

[15] Jing Ma, Wei Gao, and Kam-Fai Wong. Detect rumors on twitter by promoting information campaigns with generative adversarial learning. In *The World Wide Web Conference*, pages 3049–3055, 2019.

[16] Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 395–405, 2019.

[17] Dhruv Khattar, Jaipal Singh Goud, Manish Gupta, and Vasudeva Varma. Mvae: Multimodal variational autoencoder for fake news detection. In *The World Wide Web Conference*, pages 2915–2921, 2019.

[18] Zhiwei Jin, Juan Cao, Yongdong Zhang, Jianshe Zhou, and Qi Tian. Novel visual and statistical image features for microblogs news verification. *IEEE transactions on multimedia*, 19(3):598–608, 2016.

[19] Thanh Thi Nguyen, Cuong M Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, and Saeid Nahavandi. Deep learning for deepfakes creation and detection. *arXiv preprint arXiv:1909.11573*, 1, 2019.

[20] Jing Ma, Wei Gao, and Kam-Fai Wong. Detect rumors in microblog posts using propagation structure via kernel learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 708–717, 2017.

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

[22] Jing Ma, Wei Gao, and Kam-Fai Wong. Rumor detection on twitter with tree-structured recursive neural networks. Association for Computational Linguistics, 2018.

[23] Fang Jin, Edward Dougherty, Parang Saraf, Yang Cao, and Naren Ramakrishnan. Epidemiological modeling of news and rumors on twitter. In *Proceedings of the 7th Workshop on Social Network Mining and Analysis*, page 8. ACM, 2013.

[24] Derek Ruths. The misinformation machine. *Science*, 363(6425):348–348, 2019.

[25] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1751–1754. ACM, 2015.

[26] Yi-Ju Lu and Cheng-Te Li. Gcan: Graph-aware co-attention networks for explainable fake news detection on social media. *arXiv preprint arXiv:2004.11648*, 2020.

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

[28] Liang Wu and Huan Liu. 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*, pages 637–645, 2018.

[29] Eugenio Tacchini, Gabriele Ballarin, Marco L Della Vedova, Stefano Moret, and Luca de Alfaro. Some like it hoax: Automated fake news detection in social networks. *arXiv preprint arXiv:1704.07506*, 2017.

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

[31] Alessandro Bessi and Emilio Ferrara. Social bots distort the 2016 us presidential election online discussion. *First Monday*, 21(11-7), 2016.

[32] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Alessandro Flammini, and Filippo Menczer. The spread of fake news by social bots. *arXiv preprint arXiv:1707.07592*, 96:104, 2017.

[33] Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. Botornot: A system to evaluate social bots. In *Proceedings of the 25th international conference companion on world wide web*, pages 273–274, 2016.

[34] Michela Del Vicario, Alessandro Bessi, Fabiana Zollo, Fabio Petroni, Antonio Scala, Guido Caldarelli, H Eugene Stanley, and Walter Quattrociocchi. The spreading of misinformation online. *Proceedings of the National Academy of Sciences*, 113(3):554–559, 2016.

[35] Walter Quattrociocchi, Antonio Scala, and Cass R Sunstein. Echo chambers on facebook. *Available at SSRN 2795110*, 2016.

[36] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36, 2017.

[37] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*, 2017.

[38] Matthew L Newman, James W Pennebaker, Diane S Berry, and Jane M Richards. Lying words: Predicting deception from linguistic styles. *Personality and social psychology bulletin*, 29(5):665–675, 2003.

[39] Feng Qian, Chengyue Gong, Karishma Sharma, and Yan Liu. Neural user response generator: Fake news detection with collective user intelligence. In *IJCAI*, volume 18, pages 3834–3840, 2018.

[40] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[41] Xinyi Zhou and Reza Zafarani. Fake news: A survey of research, detection methods, and opportunities. *arXiv preprint arXiv:1812.00315*, 2, 2018.

[42] Karishma Sharma, Feng Qian, He Jiang, Natali Ruchansky, Ming Zhang, and Yan Liu. Combating fake news: A survey on identification and mitigation techniques. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(3):1–42, 2019.

[43] Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)*, 51(2):32, 2018.

[44] Srijan Kumar and Neil Shah. False information on web and social media: A survey. *arXiv preprint arXiv:1804.08559*, 2018.

[45] Atanu Roy. Computational trust at various granularities in social networks. 2015.

[46] Donovan Artz and Yolanda Gil. A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):58–71, 2007.

[47] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, pages 640–651. ACM, 2003.

[48] Abhinav Mishra and Arnab Bhattacharya. Finding the bias and prestige of nodes in networks based on trust scores. In *Proceedings of the 20th international conference on World wide web*, pages 567–576. ACM, 2011.

[49] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[50] Atanu Roy, Chandrima Sarkar, Jaideep Srivastava, and Jisu Huh. Trustingness & trustworthiness: A pair of complementary trust measures in a social network. In *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 549–554. IEEE Press, 2016.

[51] Bhavtosh Rath, Wei Gao, and Jaideep Srivastava. Evaluating vulnerability to fake news in social networks: A community health assessment model. 2019.

[52] Bhavtosh Rath, Wei Gao, Jing Ma, and Jaideep Srivastava. From retweet to believability: Utilizing trust to identify rumor spreaders on twitter. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 179–186. ACM, 2017.

[53] Bhavtosh Rath, Aadesh Salecha, and Jaideep Srivastava. Detecting fake news spreaders in social networks using inductive representation learning. *arXiv preprint arXiv:2011.10817*, 2020.

[54] Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. A stylometric inquiry into hyperpartisan and fake news. *arXiv preprint arXiv:1702.05638*, 2017.

[55] Tanushree Mitra, Graham P Wright, and Eric Gilbert. A parsimonious language model of social media credibility across disparate events. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 126–145. ACM, 2017.

[56] Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, page 13. ACM, 2012.

[57] Zhiwei Jin, Juan Cao, Yongdong Zhang, and Jiebo Luo. News verification by exploiting conflicting social viewpoints in microblogs. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[58] Adrien Friggeri, Lada Adamic, Dean Eckles, and Justin Cheng. Rumor cascades. In *Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[59] Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. Rosetta: Large scale system for text detection and recognition in images. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 71–79. ACM, 2018.

[60] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. Detecting rumors from microblogs with recurrent neural networks. In *Ijcai*, pages 3818–3824, 2016.

[61] Mark EJ Newman. Spread of epidemic disease on networks. *Physical review E*, 66(1):016128, 2002.

[62] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. Efficient estimation of influence functions for sis model on social networks. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[63] Laijun Zhao, Hongxin Cui, Xiaoyan Qiu, Xiaoli Wang, and Jiajia Wang. Sir rumor spreading model in the new media age. *Physica A: Statistical Mechanics and its Applications*, 392(4):995–1003, 2013.

[64] Devavrat Shah and Tauhid Zaman. Rumors in a network: Who's the culprit? *IEEE Transactions on information theory*, 57(8):5163–5181, 2011.

[65] Kai Zhu and Lei Ying. Information source detection in the sir model: A sample-path-based approach. *IEEE/ACM Transactions on Networking*, 24(1):408–421, 2014.

[66] Lidan Fan, Weili Wu, Xuming Zhai, Kai Xing, Wonjun Lee, and Ding-Zhu Du. Maximizing rumor containment in social networks with constrained time. *Social Network Analysis and Mining*, 4(1):214, 2014.

[67] Lidan Fan, Zaixin Lu, Weili Wu, Bhavani Thuraisingham, Huan Ma, and Yuanjun Bi. Least cost rumor blocking in social networks. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*, pages 540–549. IEEE, 2013.

[68] Nam P Nguyen, Guanhua Yan, My T Thai, and Stephan Eidenbenz. Containment of misinformation spread in online social networks. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 213–222. ACM, 2012.

[69] Wanita Sherchan, Surya Nepal, and Cecile Paris. A survey of trust in social networks. *ACM Computing Surveys (CSUR)*, 45(4):47, 2013.

[70] C-N Ziegler and Georg Lausen. Spreading activation models for trust propagation. In *IEEE International Conference on e-Technology, e-Commerce and e-Service, 2004. EEE'04. 2004*, pages 83–97. IEEE, 2004.

[71] Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. Processing social media messages in mass emergency: A survey. *ACM Computing Surveys (CSUR)*, 47(4):1–38, 2015.

[72] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[73] Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[74] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106, 2007.

[75] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. Introduction to information retrieval. In *Proceedings of the international communication of association for computing machinery conference*, volume 4, 2008.

[76] Andrew Guess, Jonathan Nagler, and Joshua Tucker. Less than you think: Prevalence and predictors of fake news dissemination on facebook. *Science advances*, 5(1):eaau4586, 2019.

[77] Gordon W Allport and Leo Postman. The psychology of rumor. 1947.

[78] Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. Real-time rumor debunking on twitter. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1867–1870. ACM, 2015.

[79] Jing Ma, Wei Gao, and Kam-Fai Wong. Detect rumor and stance jointly by neural multi-task learning. In *Companion Proceedings of the The Web Conference 2018*, pages 585–593, 2018.

[80] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1395–1405. International World Wide Web Conferences Steering Committee, 2015.

[81] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.

[82] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, Tieniu Tan, et al. A convolutional approach for misinformation identification. 2017.

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

[84] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[85] Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. Tweetcred: Real-time credibility assessment of content on twitter. In *International Conference on Social Informatics*, pages 228–243. Springer, 2014.

[86] Zhiwei Jin, Juan Cao, Yu-Gang Jiang, and Yongdong Zhang. News credibility evaluation on microblog with a hierarchical propagation model. In *2014 IEEE International Conference on Data Mining*, pages 230–239. IEEE, 2014.

[87] Miriam J Metzger and Andrew J Flanagin. Credibility and trust of information in online environments: The use of cognitive heuristics. *Journal of Pragmatics*, 59:210–220, 2013.

[88] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

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

[90] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[91] Yoshua Bengio, Patrice Simard, Paolo Frasconi, et al. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[92] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[93] Alex Graves. Generating sequences with recurrent neural networks. *arXiv:1308.0850*, 2013.

[94] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[95] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874, 2008.

[96] Bryan Conroy and Paul Sajda. Fast, exact model selection and permutation testing for l2-regularized logistic regression. In *Artificial Intelligence and Statistics*, pages 246–254, 2012.

[97] Fuhao Zou, Yunfei Wang, Yang Yang, Ke Zhou, Yunpeng Chen, and Jingkuan Song. Supervised feature learning via l2-norm regularized logistic regression for 3d object recognition. *Neurocomputing*, 151:603–611, 2015.

[98] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[99] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537, 2011.

[100] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

[101] Justin Sampson, Fred Morstatter, Liang Wu, and Huan Liu. Leveraging the implicit structure within social media for emergent rumor detection. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 2377–2382. ACM, 2016.

[102] Tu Ngoc Nguyen, Cheng Li, and Claudia Niederée. On early-stage debunking rumors on twitter: Leveraging the wisdom of weak learners. In *International Conference on Social Informatics*, pages 141–158. Springer, 2017.

[103] Liang Wu, Jundong Li, Xia Hu, and Huan Liu. Gleaning wisdom from the past: Early detection of emerging rumors in social media. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 99–107. SIAM, 2017.

[104] Tong Chen, Xue Li, Hongzhi Yin, and Jun Zhang. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 40–52. Springer, 2018.

[105] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[106] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.

[107] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377, 2019.

[108] Yuhao Zhang, Peng Qi, and Christopher D Manning. Graph convolution over pruned dependency trees improves relation extraction. *arXiv preprint arXiv:1809.10185*, 2018.

[109] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 974–983, 2018.

[110] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.

[111] Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 549–556, 2020.

[112] Lik Mui, Mojdeh Mohtashemi, and Ari Halberstadt. A computational model of trust and reputation. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, pages 2431–2439. IEEE, 2002.

[113] Bhavtosh Rath, Wei Gao, Jing Ma, and Jaideep Srivastava. Utilizing computational trust to identify rumor spreaders on twitter. *Social Network Analysis and Mining*, 8(1):64, 2018.

[114] Ortwin Renn and Debra Levine. Credibility and trust in risk communication. In *Communicating risks to the public*, pages 175–217. Springer, 1991.

[115] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

[116] Marianne E Jaeger, Susan Anthony, and Ralph L Rosnow. Who hears what from whom and with what effect: A study of rumor. *Personality and Social Psychology Bulletin*, 6(3):473–478, 1980.

[117] Richard E Petty and John T Cacioppo. *Communication and persuasion: Central and peripheral routes to attitude change*. Springer Science & Business Media, 2012.

[118] Ralph L Rosnow. Inside rumor: A personal journey. *American psychologist*, 46(5):484, 1991.

[119] Meredith Ringel Morris, Scott Counts, Asta Roseway, Aaron Hoff, and Julia Schwarz. Tweeting is believing? understanding microblog credibility perceptions. In *Proceedings of the ACM 2012 conference on computer supported cooperative work*, pages 441–450, 2012.

[120] Hyejin Kim. The role of trust in rumor suppression on social media: A multimethod approach applying the trust scores in social media (tsm) algorithm. 2019.

[121] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[122] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[123] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[124] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 950–958, 2019.

[125] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *The World Wide Web Conference*, pages 2091–2102, 2019.

[126] Qi Cao, Huawei Shen, Jinhua Gao, Bingzheng Wei, and Xueqi Cheng. Popularity prediction on social platforms with coupled graph neural networks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 70–78, 2020.

[127] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[128] Han Guo, Juan Cao, Yazi Zhang, Junbo Guo, and Jintao Li. Rumor detection with hierarchical social attention network. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 943–951, 2018.

[129] Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. Deepinf: Social influence prediction with deep learning. In *Proceedings of the*

24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2110–2119, 2018.

[130] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th international conference on World wide web*, pages 665–674, 2011.

[131] Amo Tong, Ding-Zhu Du, and Weili Wu. On misinformation containment in online social networks. In *Advances in neural information processing systems*, pages 341–351, 2018.

[132] Senzhang Wang, Xiaojian Zhao, Yan Chen, Zhoujun Li, Kai Zhang, and Jiali Xia. Negative influence minimizing by blocking nodes in social networks. In *Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[133] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. Minimizing the spread of contamination by blocking links in a network. In *Aaai*, volume 8, pages 1175–1180, 2008.

[134] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[135] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[136] Bhavtosh Rath and Jaideep Srivastava. Epidemiology inspired framework for fake news mitigation in social networks.

[137] Bhavtosh Rath, Jisu Kim, Jisu Huh, and Jaideep Srivastava. Impact of news organizations' trustworthiness and social media activity on audience engagement. *arXiv preprint arXiv:1808.09561*, 2018.

[138] Jisu Huh, Hyejin Kim, Bhavtosh Rath, Xinyu Lu, and Jaideep Srivastava. You reap where you sow and trust is the key to successful seeding: Computational research applying the trust scores in social media (tsm) algorithm. In *Proceedings of the 2018 American Academy of Advertising. Conference*, pages 48–48.

[139] Julian B Rotter. Interpersonal trust, trustworthiness, and gullibility. *American psychologist*, 35(1):1, 1980.