ABSTRACT

EARLY DETECTION OF FAKE NEWS ON SOCIAL MEDIA

by
Yang Liu

The ever-increasing popularity and convenience of social media enable the rapid widespread of fake news, which can cause a series of negative impacts both on individuals and society. Early detection of fake news is essential to minimize its social harm. Existing machine learning approaches are incapable of detecting a fake news story soon after it starts to spread, because they require certain amounts of data to reach decent effectiveness which take time to accumulate. To solve this problem, this research first analyzes and finds that, on social media, the user characteristics of fake news spreaders distribute significantly differently from those of the general user population. Based on this finding and also the fact that news spreaders' user profiles are usually readily available at the start of news propagation, this research proposes three machine learning models to achieve the goal of fake news early detection based on the user characteristics of its spreaders. The first model named *Propagation Path Classification (PPC)* detects fake news by combining recurrent neural networks with convolution neural networks to classify its propagation path which is represented as a sequence of user feature vectors. The second model named *Social Media Content Classification (SMCC)* improves the first model by adding 1) an embedding layer and an integration layer to model news spreaders, and 2) a fake news spreader likelihood score to model source users independently, which is particularly useful when the propagation path is extremely short, i.e., only very few retweets. The third model named *Fake News Early Detection (FNED)* further improves the first two models by combining users' text responses with their user characteristics as status-sensitive crowd responses, which contain more information than text responses or user characteristics alone. Two novel deep learning mechanisms are also proposed as key components in the third model: 1) Position-aware attention mechanism to determine which status-sensitive crowd responses are more discriminative; and 2) Multi-region mean-pooling to aggregate intermediate features in multiple timeframes, which improves the performance when very few retweets are available and thus needing zero-padding. The third model also incorporates a PU-Learning (Learning from Positive and Unlabeled Examples) framework to handle unlabeled and imbalanced data.

Comprehensive experiments were conducted to evaluate the proposed models on two datasets collected from Twitter and Sina Weibo, respectively. The experimental results demonstrate that the proposed models can detect fake news with over 90% accuracy within five minutes after it starts to spread and before it is retweeted 50 times, which is significantly faster than state-of-the-art baselines. Also, the third proposed model requires only 10% labeled fake news samples to achieve this effectiveness under PU-Learning settings. These advantages indicate a promising potential for the proposed models to be implemented in real-world social media platforms for fake news detection.

EARLY DETECTION OF FAKE NEWS ON SOCIAL MEDIA

by
Yang Liu

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology – Newark
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Information Systems

Department of Informatics

December 2019

APPROVAL PAGE

EARLY DETECTION OF FAKE NEWS ON SOCIAL MEDIA

Yang Liu

Dr. Yi-Fang Brook Wu, Dissertation Advisor Date
Associate Professor of Informatics, New Jersey Institute of Technology

Dr. Vincent Oria, Committee Member Date
Professor of Computer Science, New Jersey Institute of Technology

Dr. Hai Nhat Phan, Committee Member Date
Assistant Professor of Informatics, New Jersey Institute of Technology

Dr. Shaohua David Wang, Committee Member Date
Assistant Professor of Informatics, New Jersey Institute of Technology

Dr. Zhi Wei, Committee Member Date
Professor of Computer Science, New Jersey Institute of Technology

BIOGRAPHICAL SKETCH

Author:	Yang Liu
Degree:	Doctor of Philosophy
Date:	December 2019

Undergraduate and Graduate Education:

- Doctor of Philosophy in Information Systems,
  New Jersey Institute of Technology, Newark, NJ, 2019

- Bachelor of Engineering,
  Tongji University, Shanghai, P. R. China, 2013

Major: Information Systems

Presentations and Publications:

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 Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI' 18)*, 2018.

Yang Liu and Songhua Xu, "A local context-aware LDA model for topic modeling in a document network," *Journal of the Association for Information Science and Technology (JASIST' 16)*, pp 1429-1448, 2016.

Yang Liu and Songhua Xu, "Detecting rumors through modeling information propagation networks in a social media environment," *IEEE Transactions on Computational Social Systems (TCSS' 16)*, vol. 3, pp 46-62, 2016.

Yang Liu and Songhua Xu, "Detecting rumors through modeling information propagation networks in a social media environment," *In International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction (SBP' 15)*, pp 121-131, 2015.

Yang Liu, Songhua Xu, Hong-Jun Yoon, and Georgia Tourassi, "Extracting patient demographics and personal medical information from online health forums," *In AMIA Annual Symposium Proceedings (AMIA' 15)*, vol. 2014, p. 1825, 2014.

Yang Liu, Songhua Xu, and Lian Duan, "Relationship Emergence Prediction in Heterogeneous Networks through Dynamic Frequent Subgraph Mining," *In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM' 14)*, pp 1649-1658, 2014.

This dissertation is dedicated to my beloved family.
To my parents, grandparents,
With whom I have shared
Many precious moments of my life.

ACKNOWLEDGMENT

First, I would like to express my deepest appreciation to my research advisor, Dr. Yi-Fang Brook Wu, for her insightful guidance, support, encouragement, kindness and enthusiasm during the course of this work and my entire graduate study at NJIT.

I would also like to thank the rest of my dissertation committee members, Dr. Vincent Oria, Dr. Hai Nhat Phan, Dr. Shaohua David Wang, and Dr. Zhi Wei, for their valuable comments, insightful questions and actively participating in my committee. It was my great honor to have the committee's guidance and help to complete this dissertation.

In addition, I would like to thank my fellow graduate students such as Ye, Eric, Mussa, Han, and Yi at our research lab for their interesting discussions.

Finally, I am grateful to all the colleagues such as Dr. Biocca Frank, Dr. Michael Lee, Dr. Richard Egan, Dr. Lin Lin, Dr. Quentin Jones, and Dr. Donghee Yvette Wohn in the Department of Informatics at NJIT for their support and encouragement.

TABLE OF CONTENTS

Chapter Page

TABLE OF CONTENTS
(Continued)

Chapter Page

TABLE OF CONTENTS
(Continued)

Chapter Page

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF TABLES
(Continued)

LIST OF FIGURES

Figure Page

LIST OF FIGURES
(Continued)

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Background

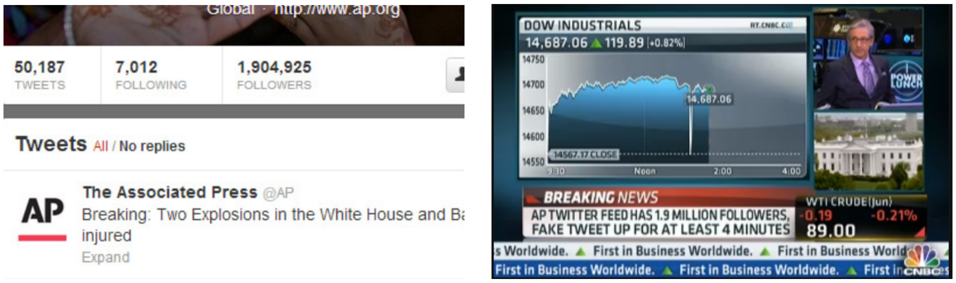
Nowadays, as social media becomes indispensable, people consume news more often from social media than from traditional news media. It was reported that in 2017, 67% of U.S. adults consumed news mainly from social media[1]. Social media enables news to reach a broad audience rapidly due to its inherent advantages over traditional news media: (i) It is less expensive in terms of both time and money to consume news from social media; (ii) It is easier to disseminate news via social media; (iii) News consumers become news spreaders after sharing a news article to their online friends; (iv) It requires less content censorship for a news article to be posted on social media. However, these advantages in the meanwhile enable "fake news," i.e., news carrying intentionally and verifiably false information to spread widely and rapidly among social media users. Researchers found that fake news spread significantly farther, faster, deeper, and more broadly than true news (Vosoughi, Roy, & Aral, 2018). Two different studies conducted in 2016 found that 23% of Americans say they have shared fake news stories [2].

The fast and massive spreading of fake news can cause inestimable social harm. For example, fake news can manipulate the outcome of political events such as the election. During the 2016 U.S. presidential election, the top 20 election-related fake news stories, most of which had information favoring Donald Trump, received more Facebook engagements than the top 20 legitimate mainstream news stories, most of which were pro-Hillary Clinton[3]. Thus, some commentators had suggested that Donald Trump would not have been elected president, were it not for the influence of fake news (Allcott &

[1] http://www.journalism.org/2017/09/07/news-use-across-social-media-platforms-2017/

[2] https://www.consumer-action.org/english/articles/fake_news

[3] https://www.vox.com/new-money/2016/11/16/13659840/facebook-fake-news-chart



The Associated Press @AP
Breaking: Two Explosions in the White House and Ba
injured

Global · http://www.ap.org
50,187 TWEETS
7,012 FOLLOWING
1,904,925 FOLLOWERS
Tweets All / No replies
AP
Expand
DOW INDUSTRIALS
14,687.06 ▲ 119.89 (+0.82%)
14750
14700
14650
14600
14550
14,687.06
14567.17 CLOSE
9:30
Noon
2:00
4:00
POWER LUNCH
BREAKING NEWS
AP TWITTER FEED HAS 1.9 MILLION FOLLOWERS,
FAKE TWEET UP FOR AT LEAST 4 MINUTES
WTI CRUDE(Jun)
-0.19 -0.21%
89.00
s Worldwide. ▲ First in Business Worldwide. ▲ First in Business World
First in Business Worldwide. ▲ First in Business Worldwide. ▲ First in
CNBC

(a) A fake news on Twitter

(b) Its impact on the Dow

Figure 1.1 A fake news story on Twitter and its impact on the Dow.

Gentzkow, 2017). Other than political repercussions, fake news can also cause severe damage to the economy by creating panic over the market rapidly. In 2013, a hacker's false Associated Press (AP) tweet claiming that an "explosion" had injured President Obama (shown in Figure 1.1-(a)) caused stocks to briefly plunge shortly after the tweet was released. Within 6 minutes, the Dow plunged over 140 points (shown in Figure 1.1-(b)), and the estimated temporary loss of market cap in the S&P 500 alone totaled $136.5 billion[4].

The prevalence of fake news on social media and its serious negative impacts have become a primary concern of the general public. A 2017 survey found that that almost three out of five Americans believe that fake news is a serious threat to their financial decision-making[5]. The phrase "fake news" has been declared the official Collins Dictionary Word of the Year for 2017[6]. To mitigate the negative effects caused by fake news, it is crucial to stop fake news before it reaches a broad audience. One of the key steps to achieve this goal is *early detection of fake news*, i.e., detecting fake news shortly after it starts to spread.

[4] https://www.cnbc.com/id/100646197

[5] https://www.aicpa.org/press/pressreleases/2017/fake-financial-news-is-a-real-threat-to-majority-of-americans-new-aicpa-survey.html

[6] http://www.newsweek.com/fake-news-word-year-collins-dictionary-699740

Human efforts have been involved in detecting and combatting fake news. Fact-checking sites, e.g., Snopes[7], Politifact[8], Factcheck.org[9], etc., rely on human experts to investigate and judge potential fake news articles reported by online readers. The judging results are then released to the public as a reference for fact-checking (shown in Figure 1.2). After the 2016 election, Google and Facebook also took steps to combat fake news. Facebook enables users to mark news stories as fake[10]. The marked news stories will then be subjected to a fact-checking process and will be attached with a warning label below its link to discourage users from sharing it if the news story is confirmed as fake news. Google enhanced its search function by displaying the fact-checking result conducted by news publishers and fact-checking organizations under the snippet of news stories[11]. Although manual fact-checking can indeed help readers identify fake news, they are far from meeting the goal of fake news early detection because of the following reasons. First, manual fact-checking often delivers a late response to fake news because it is time-consuming. By the time a news article is announced as fake by manual fact-checking sites or tools, it often has already reached a broad audience and caused social harm; Second, manual fact-checking is not scalable to deal with the huge amount of potential fake news articles published on the Internet every day. Under such a background, automatic detection approaches are urgently necessitated to provide real-time detection of fake news from a huge volume of news articles published every day.

1.2 Motivation

With the fast development of machine learning and deep learning (LeCun, Bengio, & Hinton, 2015) techniques during recent years, machine learning (ML)-based automatic

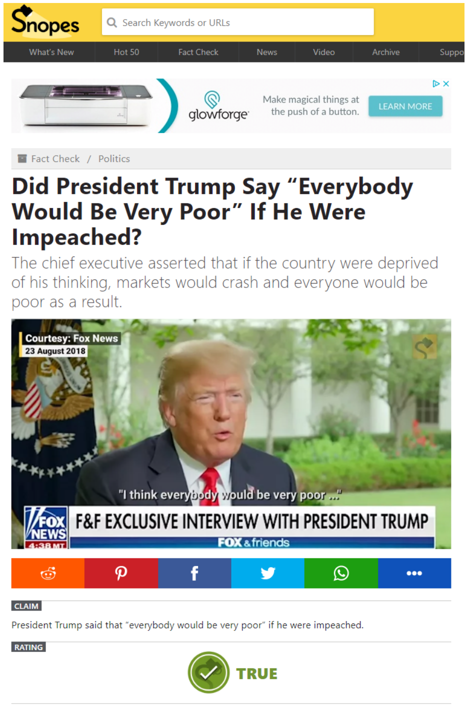
[7] https://www.snopes.com/

[8] http://www.politifact.com/

[9] https://www.factcheck.org/

[10] https://www.facebook.com/help/572838089565953?helpref=faq_content

[11] https://blog.google/products/search/fact-check-now-available-google-search-and-news-around-world/



Did President Trump Say "Everybody Would Be Very Poor" If He Were Impeached?

The chief executive asserted that if the country were deprived of his thinking, markets would crash and everyone would be poor as a result.

CLAIM

President Trump said that "everybody would be very poor" if he were impeached.

RATING

TRUE

Figure 1.2 Manual fact-checking on Snope.com.

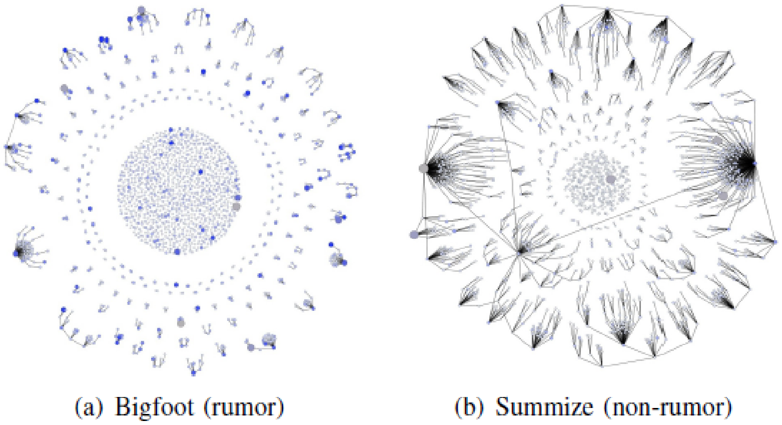
detection approaches have become a major alternative to manual fact-checking and have attracted significant attention both from the research communities and the industry. There are plenty of existing studies focusing on automatic detection of fake news (Ma et al., 2016; L. Wu, Li, Hu, & Liu, 2017; Kwon, Cha, & Jung, 2017; Ma, Gao, & Wong, 2017; Ruchansky, Seo, & Liu, 2017; Shu, Sliva, Wang, Tang, & Liu, 2017), as well as closely-related topics, such as rumor detection (K. Wu, Yang, & Zhu, 2015; Sampson, Morstatter, Wu, & Liu, 2016; L. Wu et al., 2017), misinformation detection (Qazvinian, Rosengren, Radev, & Mei, 2011; H. Zhang, Alim, Li, Thai, & Nguyen, 2016; Jain, Sharma, & Kaushal, 2016), and social spam detection (D. Wang, Irani, & Pu, 2011; Hu, Tang, Zhang, & Liu, 2013; Markines, Cattuto, & Menczer, 2009; Li & Liu, 2017), etc. Most ML-based detection approaches are based on the underlying premise that there exist some latent patterns that can differentiate fake news from true news, and those patterns can be recognized from a series of news-related features. From a data mining perspective, most of the state-of-the-art machine learning-based detection approaches work in the following routine: (i) Given a news article, relevant data required for detecting fake news is collected, which can be broadly categorized into two groups, i.e., news content data and social context data. News content includes the textual, visual, audio, and video content of a news article (Note that many online news articles contain embedded photos or videos.) The social context of a news story refers to the information related to how it spreads via social media, e.g., the author's and spreaders' information, social interactions around the news story such as comments, shares, and likes created by social media users, etc.; (ii) A set of features are extracted from the relevant data to represent the news article. Different types of features can be extracted from different kinds of relevant data. For instance, textual features such as N-grams (Brown, Desouza, Mercer, Pietra, & Lai, 1992) can be extracted from the news content. Graph theory-based features such as average in-degree and out-degree (Broder et al., 2000) can be extracted from a propagation network constructed from user sharing records; (iii) A machine learning model is then applied to predict the truthfulness of the news article based on the extracted features. The type of machine learning model is usually chosen or designed based on the feature representation of news articles.

During our literature review, we found one significant limitation of most existing machine learning-based detection approaches. That is, they only focus on improving the optimal detection effectiveness given sufficient data required to detect fake news. Recent studies have made great strides in that regard. However, we found no research focuses on early detection effectiveness when the required data is usually insufficient at this stage. The main reason is that, in order to improve the optimal detection effectiveness, many approaches extract features from an extensive amount of social context data from social interactions observed over a long period of time after a news article has been posted. Then, they apply complex machine learning models to recognize patterns from the extracted features. However, the data required by those approaches is often **unavailable or insufficient at the early stage of news propagation**. As a result, their effectiveness in early detection tend to be low. With the lack of relevant data, a machine learning model is prone to overfitting. On the other hand, by the time those approaches can effectively detect a fake news story, it usually has already spread among a large number of audiences and has resulted in some form of social harm. Early detection effectiveness is critically important because fake news usually causes social damages fast. If a detection approach cannot effectively detect fake news shortly after it starts to spread, it will have marginal usage in the real world, although they might perform well in experimental conditions.

Below is an example showing why an existing detection approach that is effective given enough amount of relevant data, is ineffective in early detection when relevant data is insufficient. A recent work (Ma et al., 2016) adopts recurrent neural networks (RNN) to detect fake news by classifying the sequence of social media posts related to the news event. According to their experimental results, the performance of their approach peaks after 24 hours after a news article starts to spread. However, the performance of their approach is much lower when the detection deadline is less than 24 hours. The reason is as follows. After we investigated their datasets, we found that the average number of posts per event at 24 hours after a news article starts to spread is around 500 in the Twitter dataset and 400 in the Weibo dataset. That is to say, their approach requires around 400-500 relevant posts to accurately detect fake news. Through our analysis of their experimental dataset, we found that the average number of posts per event is less than 200 within the first hour after a news article starts to spread and less than 50 in the first 15 minutes. When the number of relevant posts observed is much less than required, their approach's performance drops significantly. Recall the fake tweet example we discussed in Section 1.1, fake news caused significant damage to the stock market within five minutes. In such a scenario, an approach that can only detect fake news after 24 hours after it starts to spread has marginal usefulness.

Another example of a similar case is as follows. Kwon et al. (2013) extract a series of structural features from the propagation networks, e.g., median in-degree and median out-degree, to detect fake news. Figure 1.3 shows the propagation network of a fake news event named "Bigfoot" and a true news event named "Summize", respectively.

These two propagation networks are constructed from a large amount of propagation data. According to the statistics of their datasets reported in their paper, the number of spreaders and audience of the "Bigfoot" event is 462 and 1,731,926, respectively; The number of spreaders and audience of the "Summize" event is 2054 and 4,367,672, respectively. In such a condition, the two networks have significantly different structural features. Their structural difference can be easily recognized by human eyes. Thus, it is easy for a machine learning model to differentiate these two networks. However, when the two concerned news articles just start to spread, only a small propagation network can be observed, which is the center circle of the two large networks. Since it is unlikely to observe millions of audiences within the first hour of the news' propagation, in the very early stage of the news propagation, the structural difference between the two small propagation networks is no longer significant, and their respective structure looks identical via human eyes. Thus, it might be difficult for a machine learning model to differentiate these two small networks. Moreover, their paper only reported an overall detection effectiveness, not the corresponding detection deadline. Thus, their approach's performance on early detection remains unknown.


(a) Bigfoot (rumor)

(b) Summize (non-rumor)

Figure 1.3 Propagation network of a fake news article and a true news article, respectively.

With a lack of early detection capability, an ML-based detection approach will have marginal usefulness because delayed responses to fake news cannot effectively reduce its social harm. Early detection of fake news remains a challenging problem, but the research community has not reported any significant success in this regard. Besides detection efficiency, data quality is another issue. Existing studies only showed their results on fully labeled and balanced distributed experimental datasets. However, real-world data is expected to be mostly unlabeled and extremely imbalanced because verified fake news consists of only a very small portion of the entire news stream. Unfortunately, despite some laboratory results, no existing real-world fake news detection application can really solve those issues. During Mark Zuckerberg's congressional hearing in April 2018, the CEO of Facebook stated that artificial intelligence would solve Facebook's most vexing problems, including fake news, but the outcome is expected to be seen in five to ten years[12].

1.3 Overview of the Proposed Research

In this study, we define fake news as "news carrying intentionally and verifiably false information". Our research objective is to propose a machine learning model to detect fake news on social media shortly after it starts to spread, and before it reaches a broad audience. Our proposed research framework is summarized in Figure 1.4. To solve this research problem, we first analyzed the existing datasets and found that on social media,

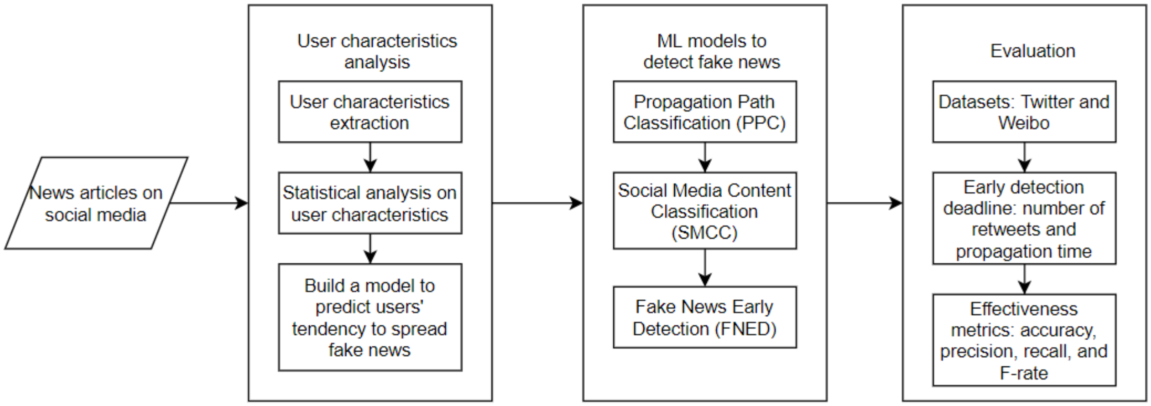

News articles on social media
User characteristics analysis
User characteristics extraction
Statistical analysis on user characteristics
Build a model to predict users' tendency to spread fake news
ML models to detect fake news
Propagation Path Classification (PPC)
Social Media Content Classification (SMCC)
Fake News Early Detection (FNED)
Evaluation
Datasets: Twitter and Weibo
Early detection deadline: number of retweets and propagation time
Effectiveness metrics: accuracy, precision, recall, and F-rate

Figure 1.4 Overview of the proposed research framework.

[12] https://www.washingtonpost.com/news/the-switch/wp/2018/04/11/ai-will-solve-facebooks-most-vexing-problems-mark-zuckerberg-says-just-dont-ask-when-or-how

the user characteristics of fake news spreaders distribute significantly differently from those of the general user population. For instance, fake news spreaders tend to have a shorter registration age than normal users. Also, as a recent study (Shu, Wang, & Liu, 2018) pointed out, there are specific users who are more likely to share fake news, and these users possess different features from those who are not as likely to share fake news. These findings laid the foundation of using user profiles for fake news detection.

Based on these findings, we built a machine learning model to predict whether a user is likely to spread fake news through the utilization of his/her user characteristics. It achieved a good prediction accuracy based on our experimental results. Since user characteristics can reflect a user's tendency to spread fake news, thus, it provides us with a possibility to identify fake news based on its spreaders' user characteristics. If a news article is spread by many users who are very likely to be fake news spreaders, then it is very likely to be fake news. Also, since at the early stage of news propagation, news spreaders' user profiles are usually readily available compared to other types of data required by existing approaches, detecting fake news based on spreaders' user characteristics can be potentially much more efficient than existing approaches that require complex features.

Based on these assumptions, in this study, we proposed three machine learning models to detect fake news early. The first model named *Propagation Path Classification (PPC)* combines recurrent neural networks with convolution neural networks to classify news propagation paths. The second model named *Social Media Content Classification (SMCC)* improves the first model by adding 1) an embedding layer and an integration layer to model news spreaders, and 2) a fake news spreader likelihood score to model source users independently, which is particularly useful when the propagation path is short, i.e., only very few retweets. The third model named *Fake News Early Detection (FNED)* further improves the first two models. It combines users' text responses with their user characteristics as status-sensitive crowd responses, which contain more information than text responses or user characteristics alone.

We also proposed two novel deep learning mechanisms as key components in the third model, i.e., position-aware attention mechanism and multi-region mean-pooling. Position-aware attention mechanism determines which status-sensitive crowd responses are more discriminative, which need to be paid with more attention by the model. Multi-region mean-pooling aggregates intermediate features in multiple timeframes, which improves the performance of early detection when very few retweets are available and needing zero-padding. The third model also incorporates a PU-Learning framework to handle unlabeled and imbalanced data. We conducted comprehensive experiments to evaluate the proposed models on two datasets collected from Twitter and Sina Weibo, respectively. Experimental results demonstrate that our proposed models can detect fake news with over 90% accuracy within 5 minutes after it starts to spread and before it is retweeted 50 times, which is significantly faster than state-of-the-art baselines. Also, our third model requires only 10% labeled fake news samples to achieve this effectiveness under PU-Learning settings. Those advantages indicate promising potential for our models to be implemented in real-world social media platforms for fake news detection.

It is equally important to mention here that, since our approach does not analyze the content of a news story itself, it is both content- and domain-independent. Thus, it also implies that the formats of news (text, video, audio) are unimportant in our approach. We should also make clear that our proposed approach is used to detect whether a news article is potentially fake as a whole. It is not designed to pinpoint which part of the news article is fake and why it is fake. In the real-world scenario, our proposed approach can be applied on social media sites as a filter to label potential fake news articles automatically. This is the first step in combating fake news, i.e., “fake news early detection”. Then, the labeled articles can be sent to social media administrators who will perform content verification and then decide how to handle them. This is the second step in combating fake news, i.e., “fake news verification”.

1.4 Organization of the Dissertation

The remainder of this dissertation is organized as follows: Chapter 2 presents a theoretical background of the fake news detection problem and an overview of existing detection approaches in the literature. Chapter 3 presents a study on user characteristics and a machine learning model to predict a user's tendency to spread fake news. Chapter 4 introduces the Propagation Path Classification (PPC) model. Chapter 5 introduces the Social Media Content Classification (SMCC) model that improves the first model. Chapter 6 introduces the Fake News Early Detection (FNED) model which further improves the first two models. Chapter 7 provides limitations, discussions, contributions, future directions, and a summary of this research.

CHAPTER 2

LITERATURE REVIEW

This chapter presents a theoretical background of fake news detection and an overview of existing detection approaches in the literature. Section 2.1 presents some theoretical background of fake news and its relationship with social media, as well as why people tend to believe fake news. Section 2.2 presents an overview of existing machine learning-based automatic detection approaches. Section 2.3 summarizes our literature review.

2.1 Theoretical Background

2.1.1 What is “Fake News”

The problem of fake news has existed since news began to circulate widely after the printing press was invented in 1439 (Biyani, Tsioutsiouliklis, & Blackmer, 2016). In recent years, fake news has reached a broader audience with the help of social media and has caused more serious social harm. Fake news detection has been widely studied by both academic communities and the industry. However, there is still no agreement on the definition of fake news among many existing studies. Therefore, we first discuss and compare several definitions of fake news that are adopted in existing studies. Then, we give our definition of fake news that will be adopted in the rest of this research.

Fake news was exclusively used in the satire context (Brewer, Young, & Morreale, 2013; Balmas, 2014; V. Rubin, Conroy, Chen, & Cornwell, 2016). Balmas et al. (2014) found that fake news is meant to perceived as unrealistic, while traditional news content is meant to be perceived as realistic. Cohen et al. (2017) provided a broad definition of fake news, i.e., fake news is everything from malicious stories to political propaganda. They pointed out that many articles are written by journalists who write articles using web searches but with no actual verification. Willnat et al. (2014) found that 53.8% of journalists use microblogs (ex. Twitter) to gather information and report from news stories.

In a recent study of fake news in the 2016 election (Allcott & Gentzkow, 2017), it is defined as a news article that is intentionally and verifiably false and could mislead readers. This definition has been widely adopted in several existing studies (Conroy, Rubin, & Chen, 2015; Klein & Wueller, 2017; Mustafaraj & Metaxas, 2017; Potthast, Kiesel, Reinartz, Bevendorff, & Stein, 2017). Based on the two key features of fake news under this definition, i.e., *authenticity* and *intent*, a recent survey paper on the topic of fake news detection (Shu et al., 2017) provides a more concise definition of fake news, i.e., fake news is a news article that is intentionally and verifiably false. Under this definition, fake news must include information that can be verified as false and must be intentionally created to mislead readers. Nowadays, the fast development of social media and Web 2.0 enables fake news to be shared over millions of times and generates a huge amount of advertising revenue. Considering this impact, Klein et al. (2017) define fake news as the online publication of intentionally or knowingly false statements of fact. Several previous studies regard fake news as a particular news article being intentionally deceptive (fake, fabricated, staged news, or a hoax) (V. L. Rubin, Chen, & Conroy, 2015; V. L. Rubin, 2017). Since the scope of this study is detecting fake news on social media, based on the definitions discussed above, we formally define fake news as follows,

Definition 2.1.1. (FAKE NEWS) Fake news is a news article that carries intentionally and verifiably false information.

2.1.2 Related Terms

There has been a variety of existing studies that focus on topics related to fake news detection, e.g., rumor detection (K. Wu et al., 2015; Sampson et al., 2016; L. Wu et al., 2017), misinformation detection (Qazvinian et al., 2011; H. Zhang et al., 2016; Jain et al., 2016), and spam detection (Hu et al., 2013; Markines et al., 2009; Li & Liu, 2017), etc. In this section, we distinguish the concept of fake news from a variety of related concepts such as rumor, misinformation, spam, etc., because of the following reasons. First, it is necessary to clarify the scope of this study, i.e., detecting fake news instead of rumors, spam, etc. Second, many existing papers either do not give a clear definition of those terms or have their own definition that conflict or overlap with either the definition of fake news adopted in this study or the definition of other terms adopted in other papers. Third, many existing studies focusing on those related concepts are closely relevant to our study since their method can be directly or indirectly adopted for detecting fake news.

We adopted the definition of a series of key terms related to fake news from previous research (L. Wu, Morstatter, Hu, & Liu, 2016), which introduces the concept of misinformation and the 5 Key Terms. Figure 2.1 shows a concept map with the root concept "misinformation" and a list of subconcepts. This article provides the following definitions. *Misinformation* is fake or inaccurate information that is unintentionally spread. *Disinformation* is fake or inaccurate information that is intentionally spread. A *Rumor* is a story circulating from one person to another, of which the truth is unverified or doubtful. An *Urban Legend* is a fictional story that contains themes related to local popular culture. *Spam* is unsolicited messages sent to a large number of recipients, containing irrelevant or inappropriate information, which is unwanted. A *Troll* is a user who posts messages that are deliberately offensive or provocative, with the aim of upsetting other people.

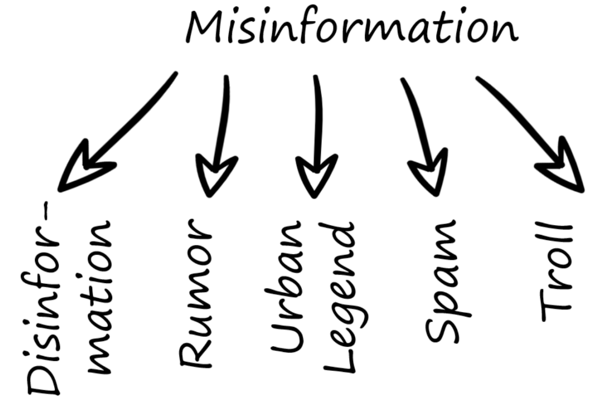


Figure 2.1 Concepts related to fake news (L. Wu, Morstatter, Hu, and Liu, 2016)

Fake news is also different from alt-facts and journalism. Alt-facts are different from fake news in that they have no basis in reality (Berghel, 2017); journalism “attempts at exercising reliability, selecting the important over the trivial while avoiding sensationalism,” instead of intentionally creating false content. (Borden & Tew, 2007)

2.1.3 Fake News on Traditional News Media

Before online news and social media became popular, fake news has been spread via traditional news media, i.e., newspaper and television, over time. We investigated several psychological and social science theories that describe why people tend to believe and spread fake news and the impact of fake news on both individuals and society.

Psychological Theories There are two major psychological and cognitive factors that make people naturally vulnerable to fake news: (1) *Naive Realism:* people tend to believe that their perceptions of reality are the only accurate views, while others who disagree are regarded as uninformed, irrational, or biased (Reed, Turiel, & Brown, 2013); and (2) *Confirmation Bias:* consumers prefer to receive information that confirms their existing views (Nickerson, 1998). Due to these two cognitive biases, fake news is often perceived as true news by some people. Moreover, people’s misperception of fake news is hard to change once it is formed. Psychology studies show that factual information is not helpful to correct false information (e.g., fake news), but sometimes can increase the misperception (Nyhan & Reifler, 2010).

Social Science Theories Many social science theories explain why people tend to spread fake news within their social circle. Prospect theory (Kahneman & Tversky, 2013; Tversky & Kahneman, 1992) describes decision making as a process by which people make choices to maximize the relative gains or minimize relative losses as compared to their current state. According to social identity theory (Tajfel & Turner, 1979, 1986) and normative influence theory (Asch & Guetzkow, 1951), social acceptance and affirmation are essential to a person’s identity and self-esteem. Due to the above theories, when a fake news article is spreading among a social group, people in the group tend to spread it, because it is a "socially safe" option and they think it can maximize their social gain.

2.1.4 Fake News on Social Media

In this subsection, we will discuss some unique characteristics of fake news on social media, which make them spread more wildly and rapidly than in traditional news media.

Malicious Accounts on Social Media On social media, there are plenty of malicious accounts that actively spread fake news. Some of them are controlled by robots instead of real humans. A social bot refers to a social media account that is controlled by a computer algorithm to automatically produce content and interact with humans (or other bot users) on social media (Ferrara, Varol, Davis, Menczer, & Flammini, 2016). Due to the low cost of creating social media accounts, a massive amount of social bots can be easily created with the specific purpose of spreading fake news on social media. One study showed that the 2016 U.S. presidential election was distorted by a massive amount of online social bots (Bessi & Ferrara, 2016). About 19 million social bot accounts on Twitter posted tweets in support of either Trump or Clinton in the single week before election day[1]. Besides social bots, trolls, i.e., real human users who actively post biased or false information on social media or online discussion forums in order to emotionally manipulate the online public, are another group of users who tend to spread fake news. They are often paid so that they have a strong incentive to spread fake news or other misinformation as widely as they can. For instance, there was evidence that showed 1,000 paid Russian trolls spread fake news on Hillary Clinton[2]. The effect of trolling is to trigger people's inner negative emotions, such as anger and fear, resulting in doubt, distrust, and irrational behavior (Shu et al., 2017). Another type of malicious account is cyborg account. Cyborg accounts have mixed functions of real human accounts and social bots. A cyborg account is usually registered

[1] http://comprop.oii.ox.ac.uk/2016/11/18/resource-for-understanding-political-bots/

[2] http://www.huffingtonpost.com/entry/russian-trolls-fake-news_us 58dde6bae4b08194e3b8d5c4

by a human but set an automated computer program that responds to human input quickly. This type of accounts is also widely used in spreading fake news (Chu, Gianvecchio, Wang, & Jajodia, 2012). To deal with malicious accounts, Twitter deleted tens of millions of suspicious accounts in the cull, which up to 6 percent of all its registered accounts[3].

Change of Roles Social media has changed people's roles in consuming and disseminating news. From a traditional communication theory's point of view (Shannon & Weaver, 1963), news is released by a source and goes through a media to reach its consumers. However, the interactive property of social media brings a fundamental shift in communication, i.e., receivers become the new "sources" (Sundar & Nass, 2001) (shown in Figure 2.2[4]). On social media, information consumers themselves become information creators and distributors once they share the information with their friends or followers (shown in Figure 2.3[5]).

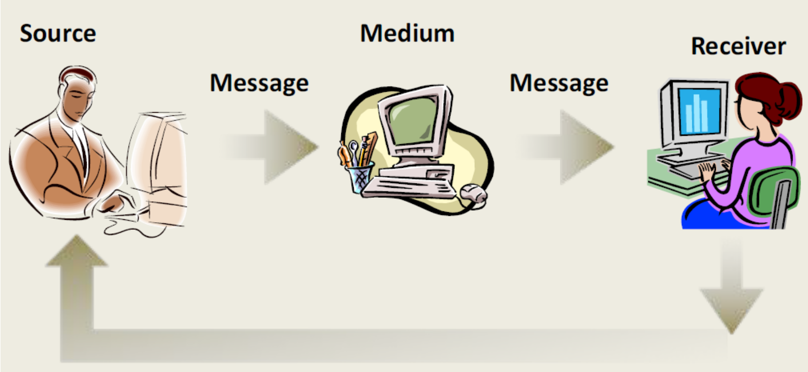


Figure 2.2 A fundamental shift in communication brought on by social media.

Echo Chamber Effect Recent findings showed that users on Facebook tend to select the information that adheres to their system of beliefs and to form polarized groups, i.e., echo chambers (Del Vicario, Vivaldo, et al., 2016). Such a tendency dominates information

[3] https://www.independent.co.uk/life-style/gadgets-and-tech/news/twitter-fake-followers-lost-delete-accounts-cull-a8444236.html
[4] https://john.cs.olemiss.edu/ñhassan/file/aaai2018tutorial.html
[5] https://john.cs.olemiss.edu/ñhassan/file/aaai2018tutorial.html

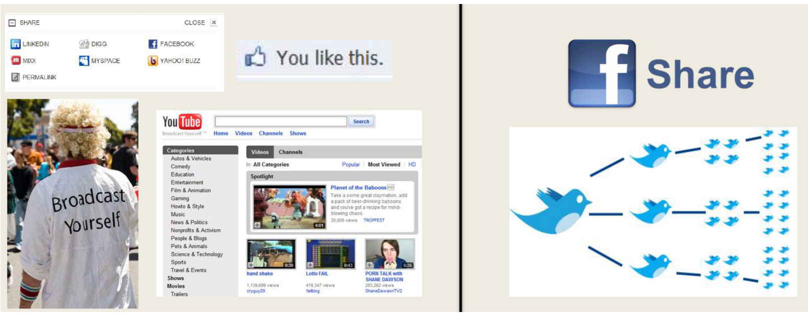


Figure 2.3 Information receivers become creators and distributors.

cascades and can affect public debates on socially relevant issues. The echo chamber effect facilitates the spreading of fake news due to the two following psychological factors (Paul & Matthews, 2016): (1) *Social Credibility*, people are more likely to perceive a piece of information as credible if others perceive it as credible, especially when the credibility of the concerned information is hard to assess due to lack of evidence; and (2) *Frequency Heuristic*, people are more likely to perceive fake news as true if it is heard frequently. Studies have shown that increased exposure to an idea is enough to generate a positive opinion of it (Zajonc, 1968; Del Vicario, Bessi, et al., 2016), and in echo chambers, users continue to share and consume the same information. As a consequence, this echo chamber effect creates segmented, homogeneous communities with a very limited information ecosystem, which becomes the primary driver of information diffusion that further strengthens polarization (Del Vicario, Bessi, et al., 2016).

2.2 Existing Detection Approaches

As we described in Chapter 1, manual fact-checking cannot meet the requirement of fake news early detection. Thus, in this section, we will present an overview of existing machine learning-based automatic detection approaches.

2.2.1 Categorization of ML-based Detection Approaches

With the fast development of machine learning and deep learning in recent years, there has been plenty of automatic detection approaches proposed in the literature. Given an online news article, a typical machine learning-based detection approach first extracts features from either its text content or its social context data or both, then applies a machine learning model/algorithm that predicts the truthfulness of the news based on the extracted features. Therefore, in this section, we categorize existing ML-based detection approaches by the following two dimensions: (1) features, and (2) machine learning model. Figure 2.4[6] shows an example of categorization of existing detection approaches, where the x-axis represents the feature type, and the y-axis represents the type of machine learning models.

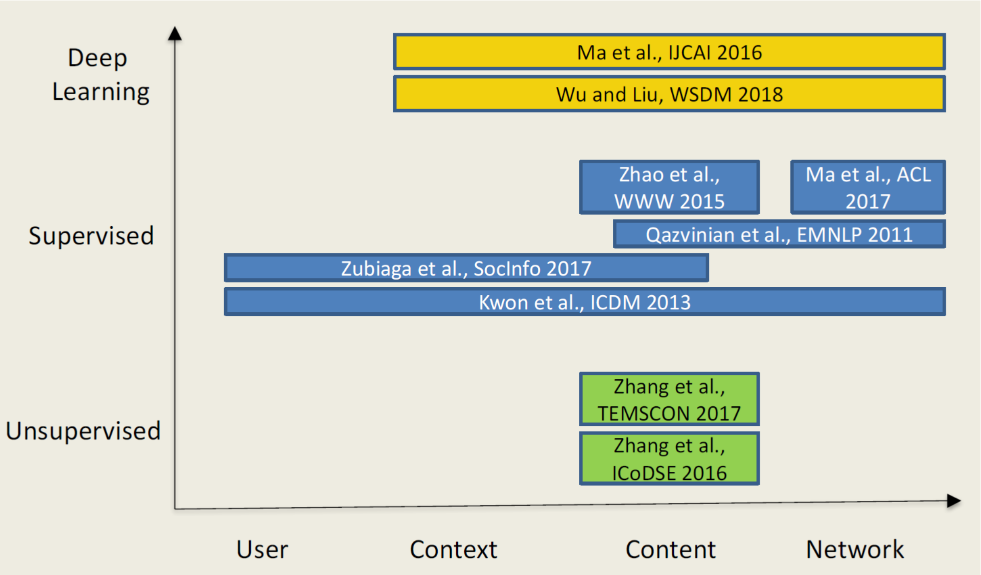


Figure 2.4 Example of a categorization of existing detection approaches.

Since machine learning models are more diverse than features, we will first group existing approached based on the features they adopt and then discuss their corresponding machine learning models. Table 2.1 shows the categorization of the features adopted by existing detection approaches.

[6] https://john.cs.olemiss.edu/ñhassan/file/aaai2018tutorial.html

Table 2.1 Categorization of Features

Feature category	Subcategory	Data source
news content-based	textual-based	headline, body text
	visual based	video, image
social context-based	user-based	user profile, user post history
	post-based	user comments, retweets
	network-based	diffusion network, social network

2.2.2 Detecting Fake News via News Content-Based Features

Content-based features broadly include textual-based features and visual-based features.

Textual-Based Textual-based features can be extracted from the news headline and its text content. An intuitive and straightforward approach adopted by many existing studies is to detect fake news based on its text content. Castillo et al. (2011) adopt a list of rudimentary content-based features, e.g., question marks, emoticon symbols, sentiment positive/negative words, pronouns, etc., to gauge the information credibility on Twitter. Popat et al. (Popat, 2017) found that the language style of an article plays a crucial role in understanding its credibility. Thus, they adopt language stylistic features, e.g., assertive verbs, factive verbs, implicatives, etc., to assess the credibility of web claims. Opinionated and inflammatory language has been adopted as indicators of fake news (Y. Chen, Conroy, & Rubin, 2015). Natural language processing (NLP) techniques (Chowdhury, 2003) have also been adopted by existing studies to discover syntaxical or semantical patterns from news content to detect fake news. Syntactic features such as n-grams and part-of-speech (POS) tags have been explored in (Fürnkranz, 1998; Qazvinian et al., 2011). Zubiaga et al. (Zubiaga, Liakata, & Procter, 2017) adopt Word2Vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) to create vector representations of words in tweets to detect rumors.

There are several limitations of text-based detection approaches. First, these approaches need enough text content to make a prediction. Thus, they cannot detect fake news with very short or no text content. Second, the textual content of fake news is diverse in terms of topic, style, and platform. Thus, content-based features that work well on one particular fake news dataset may not work well on another (Shu et al., 2017).

Visual-Based Visual-based features extracted from visual elements (e.g., images and videos) have been explored to detect fake news. Gupta et al. (2013) explored the influence of fake images on Twitter during disasters and ways to detect them. Jin et al. (2017) proposed novel visual and statistical image features for microblogs news verification. Visual features include clarity score, coherence score, similarity distribution histogram, diversity score, and clustering score. Statistical features include count, image ratio, multi-image ratio, hot image ratio, long image ratio, etc.

One limitation of adopting visual-based features is the lack of training data. Constructing a human-labeled fake news dataset is time-consuming and requires a lot of manpower. Public fake news datasets usually do not contain more than 10,000 news articles. Most of them do not include any image or video. Therefore, it is even harder to construct a fake news dataset that contains enough images or videos to train a machine learning model.

2.2.3 Detecting Fake News via Social Context-Based Features

The interactive attribute of social media enables a variety of social engagements surrounding a news story. After a news article is released on social media, users can share, comment, and discuss it with their neighborhood users within an online community. Those social engagements form the social context of the news article. The abundant amount and diversity of social context data can provide us with clues about the truthfulness of a news story. Recently, with the fast development of machine learning and deep learning techniques, advanced detection models have been developed to predict the truthfulness of online news stories based on a variety of social context-based features. We categorize social context-based features into three broad categories: user-based, post-based, and network-based.

Adopting Post-Based Features Post-based features can be extracted from a series of posts, comments, or discussions around the concerned news article. Since user engagements usually follow a sequence and their timestamps are recorded by social media platforms, a variety of temporal-based features extracted from time series of social engagements have been proposed to detect fake news. Ma et al. (2015) proposed an SVM-based model called SVM-TS that detects fake news based on time-series of aggregated news characteristics, e.g., percentage of microblogs with URL, percentage of verified users, etc. However, this type of approach has the same limitation as aggregated features and is often be unreliable for early detection. User comments are another type of sequential data. Recent works adopt deep learning techniques such as recurrent neural network (RNN) to extract temporal-linguistic patterns from sequences of user comments (Ma et al., 2016; W. Chen, Zhang, Yeo, Lau, & Lee, 2017) to identify rumors. Ma et al. (2016) proposed an RNN-based model called GRU that detects fake news based on temporal-linguistic patterns recognized from sequences of user comments. However, user comments can be very few at the early stage of a news story's propagation process, which can significantly degrade the performance of RNN models and easily cause them to overfit.

Adopting Network-Based Features Network-based features can be extracted from the propagation network of a news article, whose nodes are users who spread the news, edges are links between those users. Social media users are connected through either directed or undirected links, such as following and friendship. Thus, when a news story spread through these links, a propagation network can be observed. Existing studies have investigated structural features extracted from propagation networks as another type of feature to detect fake news. Jin et al. (2013) utilized epidemiological models to characterize information cascades in Twitter, resulting from both true news and fake news. Wu et al. (2015) proposed a graph kernel-based SVM-based classifier that learns high-order propagation patterns to detect fake news. Sampson et al. (2016) utilized implicit linkages between conversation fragments about a news story to predict its truthfulness. Ma et al. (2017) proposed a graph kernel-based SVM classifier named PTK that captures high-order patterns differentiating different types of fake news by evaluating the similarities between their propagation tree structures. Later, they proposed another deep network named RvNN ((Ma, Gao, & Wong, 2018)) based on a top-down/bottom-up tree-structured neural networks for rumor representation learning and classification. Wu et al. (2018) proposed a detection approach named TraceMiner to represent and classify propagation pathways using LSTM-RNN. However, detecting fake news based on propagation networks is inefficient because it usually takes a long time to observe a propagation network large enough to extract useful structural features.

Adopting User-Based Features User-based features include user characteristics that can be extracted from user profiles. As a recent study (Shu, Wang, & Liu, 2018) pointed out, there are some users who are more likely to share fake news, and these users possess different features from those who are not as likely to share fake news. These findings laid the foundation of using user profiles for fake news detection. Early studies adopt user-based features extracted from the user profile of news spreaders to detect fake news. Castillo et al. (2011) utilized a list of basic user-based features supported by most social media platforms, e.g., followers count, friends count, registration age, etc., to gauge the credibility of the information posted by its source user. Besides common user features, Yang et al. (2012) added some platform-specific user features, e.g., gender, registration place, etc., to detect rumor on Sina Weibo[7], the largest social media site in China. User-based features can also be categorized across the group level. Group level user-based features

[7] https://weibo.com

depict the overall characteristics of a group of news spreaders. Group level features can be constructed by aggregating individual-level features, e.g., ‘the average number of followers’ and ‘percentage of varied users’ (Kwon et al., 2013; Ma et al., 2015). Castillo et al. (2011) proposed a decision tree-based model called DTC to detect fake news based on aggregated user characteristics, i.e., average registration age and average followers count, of both source users and news spreaders.

Highly relying on user features of the source user to judge whether a news story is fake has a significant limitation. That is, fake news producers can mix a few fake news stories with a bunch of true news stories in order to increase the chance of their fake news being trusted. When a detection model is trained based on user features of source users alone, if in a particular training dataset, the news articles released by a particular user are all true news, then the next time if this user releases a fake news story, the model will label it as true. Thus, user-based features of source users alone cannot be reliably used to determine whether a news story is fake. Group level features can discard the diversity of individual-level features and lose information on individuals who engaged in spreading fake news. Also, aggregated features become statistically significant only after a number of news spreaders are observed. Thus, they are often unreliable for early detection.

Adopting Hybrid Features Recently, hybrid models that combine multiple types of features have been proposed to enhance the performance of fake news detection. A typical detection model that combines hybrid features is CSI (Ruchansky et al., 2017) that detects fake news based on a combination of temporal-linguistic features extracted from user comments and user-based features extracted from social network structure. CSI consists of three modules, i.e., Capture, Score, and Integrate. The Capture module adopts long short term memory networks (LSTM) (Gers, Schmidhuber, & Cummins, 1999) to produce a vector representation of a sequence of user comments under a particular news story. The Score module produces a credibility score for each user based on its user vector representation derived from singular value decomposition (SVD) (De Lathauwer, De Moor, Vandewalle, & by Higher-Order, 1994) of the entire social network. Then, the credibility scores of all users who engaged in spreading a news story will be reduced to one single score that represents the overall credibility of all its spreaders. The Integrate module integrates the vector representation of user comments and that of news spreaders and then produce a class label via a neural network based on the integrated vector. Guo et al. (Guo, Cao, Zhang, Guo, & Li, 2018) proposed a Hierarchical Social Attention Network that injects social context features into the LSTM model for the retweet text via attention mechanism. More hybrid detection approaches have been proposed in (Sun, Liu, He, & Du, 2013; Q. Zhang, Zhang, Dong, Xiong, & Cheng, 2015; Zhou et al., 2015; Zubiaga, Aker, Bontcheva, Liakata, & Procter, 2018; Liang, Yang, & Xu, 2016; Z. Jin, Cao, Guo, Zhang, & Luo, 2017; Nguyen, Li, & Niederée, 2017). Although these hybrid models achieved higher detection effectiveness when sufficient data is observed, they are inefficient for early detection, i.e., some key components are too complex and require long training time. For instance, as of the second quarter of 2018, Facebook had 2.23 billion monthly active users. In the CSI model, a real-time SVD of a social network consists of billions of users is extremely time-consuming thus is not suitable for real-time early detection of fake news. However, with proper adjustments, these hybrid models will likely produce better results for fake news early detection.

2.3 Summary

In this chapter, we first present a theoretical background of fake news by discussing several definitions of fake news proposed in previous studies and then give our own. Then we differentiate fake news from a series of related terms. Next, we discuss some social science theories that explain why some people tend to believe fake news on traditional news media and how social media further enhances fake news spreading. In the second part of this chapter, we present an overview of existing machine learning-based detection approaches.

We group existing approaches based on the type of feature they adopt and discuss their corresponding limitations, respectively. Based on the results of prior research and the scope of our study, it seems that utilizing user characteristics that are readily available in the user profiles might be further utilized to create new avenues for early detection of fake news on social media. In the next chapter, we will report on our comparisons of user characteristics between fake news spreaders and the rest of the user population.

CHAPTER 3

A STUDY ON USER CHARACTERISTICS AND A MACHINE LEARNING MODEL TO PREDICT A USER'S TENDENCY TO SPREAD FAKE NEWS

User characteristics of news spreaders can be potentially useful in detecting fake news because they are readily available at the early stage of news propagation. From our literature review, some studies have adopted user features to detect fake news and yielded different results. One study (Shu, Wang, & Liu, 2018) further investigates that there are some specific users who are more likely to trust fake news than real news, and these users possess different features from those who are more likely to trust real news. In their study, a subset of user profile features are examined, and a statistical t-test shows that these features distribute significantly differently between users who share the most real news pieces and users who share the most fake news pieces in their experimental dataset. Although the above-mentioned study yields convincing results, it has the following limitations: (1) It only examines a subset of user profile features instead of a complete set; (2) It does not differentiate source users, i.e., users who initially post a news piece, from news retweeters. Due to these limitations in their research, we decided to conduct a more comprehensive study, including all user characteristics that were available in our datasets and how they might contribute to predicting users' fake news spreading behavior.

To comprehensively examine whether user characteristics distribute differently between fake news spreaders and normal users, in this chapter, we first present our user study that shows how the user characteristics of fake news spreaders significantly differ from those of the general user population on social media. Then, we propose a machine learning model to predict a user's tendency to spread fake news.

3.1 Terminologies

In this section, we briefly introduce some basic terminologies used in the context of social media shown in Table 3.1.

Table 3.1 Terminologies Used in the Context of Social Media

Terminology	Explanation
User	A person or a computer program who registers on a social media platform.
Follower	Another user who follows the concerned user and will automatically receive his/her posts
Friend	Another user who is followed by the concerned user.
Post	A social media object posted by a user, e.g., a text block, a photo, an video, etc.
Retweet	The action of reposting or forwarding a message posted by another user.
User Characteristics	A series of features/attributes that describe a user, e.g., the number of followers, the number of friends, etc.
Status	A social media post plus the user characteristics of its source user.
Source user	A user who initially post a news article on social media.
Spreader	Users who retweet a news article.

3.2 Datasets

In this section, we introduce the two experimental datasets we used in our study. To evaluate the effectiveness of the proposed fake news detection framework, we conducted comprehensive experiments on two real-world datasets constructed from Twitter and Sina Weibo, respectively. We directly adopt a public Weibo dataset (Ma et al., 2016), which consists of 2,351 true news and 2,313 fake news collected during 2015-2016, since it provides all the information we need, especially user characteristics. We name this dataset as *Weibo16* in this study. We also found a public Twitter dataset (Ma et al., 2017). It consists of two parts, i.e., *Twitter15* and *Twitter16*, which are constructed based on two reference datasets collected in 2015 (X. Liu, Nourbakhsh, Li, Fang, & Shah, 2015) and 2016 (Ma et al., 2016), respectively. We slightly modified this Twitter dataset by the following steps and finally regenerated our own: (1) We removed the tweets labeled as “unverified” or “true rumor” since they are beyond our research interest; (2) We removed the tweets that are no longer accessible now, since we needed to collect their corresponding features which were not available in the original dataset, for model training; (3) We eventually discarded the original *Twitter16* dataset, because the number of remaining tweets was too small (309), and we think it is inappropriate to mix tweets collected in 2015 and those collected in 2016 together since they were collected by different approaches according to the original papers; (4) We developed a crawler to acquire corresponding user profiles for each of the remaining retweets; (5) We augmented the resultant dataset which consists of 353 true news and 327 fake news with user features extracted from the corresponding crawled user profiles and made it publicly-accessible.[1] We use the name *Twitter15* to refer to the augmented dataset in this study. Table 3.2 shows some basic statistics of the two datasets. In addition, we will use these two datasets to evaluate our proposed fake news detection models in later chapters.

Table 3.2 Statistics of the Experimental Datasets

Statistic	*Twitter15*	*Weibo16*
# news articles	680	4664
# true news	353	2351
# fake news	327	2313
# source users	277	2309
# retweeters	215,691	2,818,002

3.3 User Characteristics

In this section, we briefly introduce the set of user characteristics that are included in Twitter and Weibo platforms. Since the *Weibo16* dataset already includes user characteristics, we directly adopt a full list of them to construct user features. Those features include username length, screenname length, personal description length, followers count,

[1] https://github.com/yl558/Twitter15

friends count, listed count, attitudes count, favorites count, statuses count, registration age, “is account verified”, “is GEO enabled”, and gender. We also extract a full list of user characteristics from Twitter user profiles, most of them also appear in Weibo user profiles. Thus, here we only list those that are not included in Weibo user profiles, including favorites count, “has location info.”, “has personal URL”, “are tweets protected” (protected tweets are privately accessible), “is language English”, “has profile background tile”, “has profile background image”, and “has default profile”. The detailed explanation of each user characteristic can be found in the corresponding social media API documents. We apply log scale (log of 10) on several numerical features entitled with “X counts,” since those features have a near log-normal distribution. Registration age is measured in hours and is calculated using the time when a tweet/retweet was posted minus the time when the corresponding user was registered. Features entitled with “X length” are measured in character’s level. Boolean features such as “is account verified” are directly transformed to 0 or 1. Tables 3.3 and 3.4 show a list of user characteristics in *Twitter15* and *Weibo16* dataset, respectively.

3.4 User Categorization

As a recent study (Shu, Wang, & Liu, 2018) pointed out, there are some users who are more likely to share fake news, and these users possess different features from those who are not as likely to share fake news. To examine whether this assumption also holds in our experimental datasets, we first categorize all the users included in the datasets into the following categories: (1) **Source users** are users who initially posted news articles on social media; (2) **Fraudulent source users** are source users who have initially posted one or more fake news articles; (3) **Legitimate source users** are source users who have never posted any fake news articles; (4) **Retweeters** are users who retweeted news articles on social media; (5) **Fraudulent retweeters** are retweeters who have retweeted one or more fake news articles; (6) **Legitimate retweeters** are retweeters who have never retweeted

Table 3.3 List of User Characteristics Extracted from *Twitter15* User Profiles

No.	Feature	Type
1	Username length	Integer
2	Screenname length	Integer
3	Personal description length	Integer
4	Followers count	Float
5	Friends count	Float
6	Listed count	Float
7	Favorites count	Float
8	Statuses count	Float
9	Has location info.	Binary
10	Has personal URL	Binary
11	Are tweets protected	Binary
12	Is account verified	Binary
13	Is GEO enabled	Binary
14	Is language English	Binary
15	Is Contributors Enabled	Binary
16	Has profile background tile	Binary
17	Has profile background image	Binary
18	Has default profile	Binary
19	Has default profile image	Binary
20	Registration age	Integer

Table 3.4 List of User Characteristics Extracted from *Weibo16* User Profiles

No.	Feature	Type
1	Username length	Integer
2	Screenname length	Integer
3	Personal description length	Integer
4	Followers count	Float
5	Friends count	Float
6	Attitudes count	Float
7	Favorites count	Float
8	Statuses count	Float
9	Registration age	Integer
10	Is account verified	Binary
11	Is GEO enabled	Binary
12	Gender	Binary
13	Has location info.	Binary

any fake news articles. Table 3.5 shows the distribution of these two groups of users in the two experimental datasets. We also divide all users included in our datasets into two broad groups: *fake news spreaders* who had tweeted or retweeted at least one fake news articles, including fraudulent source users and fraudulent retweeters; *fake news ignorants* who had never tweeted or retweeted any fake news article, including legitimate source users and legitimate retweeters.

Table 3.5 Distribution of User Groups

	Twitter15	Weibo16
# Source user	277	2309
# Fraudulent source user	232	1809
# Legitimate source user	45	470
# Retweeter	215,463	2,818,002
# Fraudulent retweeter	81,302	1,622,424
# Legitimate retweeter	134,164	1,195,578

3.5 Hypothesis Testing on the Distribution of User Characteristics

In this section, we conducted hypothesis tests to investigate whether there is a significant difference between the distribution of each user feature in one specific user group, e.g., fake news spreaders or ignorants and that in the entire user population. We categorize all the social media users into six groups:

Based on the above categorization, fake news spreaders include fraudulent source users and fraudulent retweeters, fake news ignorants include legitimate source users and legitimate retweeters. Fake news spreaders and fake news ignorants can be regarded as two sets of samples taken from the entire user population.

For user features carrying continuous values, we conducted Z-tests. For one particular user feature, the **null hypothesis** is that there is no significant difference between the mean of this user feature for fake news spreaders (fraudulent source users and retweeters) / or fake news ignorants (legitimate source users and retweeters) and that for the entire user population. Z-score is calculated by the following formula:

$$z = \frac{M - \mu}{\sigma/\sqrt{n}}, \tag{3.1}$$

where M is the sample mean, i.e., the mean of one feature among fake news spreaders (or fake news ignorants), μ is the population mean, i.e., the mean of one feature for the entire user population, σ is the population variance, n is the sample size. A z-score larger than 1.5 (critical threshold based on a p-value of 0.05) will reject the null hypothesis, i.e., indicating that there is a significant difference between the mean of the concerned user feature for fake news spreaders (or ignorants) and those for the entire user population.

Tables 3.6-3.9 shows the results of Z tests. From these tables, we can find that there is a significant difference between the mean of most user features for fake news spreaders (or ignorants) and those for the entire user population in both two datasets.

Table 3.6 Results of Z-Test (*Twitter15*, Source Users)

Feature	Source users		Fraudulent source users		Legitimate source users	
	Mean	Std.	Mean	z-score	Mean	z-score
Username length	12.13	4.93	12.21	0.25	11.71	-0.57
Screenname length	9.94	3.21	10.12	0.82	9.04	-1.88
Personal description length	96.22	45.81	96.74	0.17	93.57	-0.38
Followers count	5.41	1.18	5.23	-2.33	6.35	5.03
Friends count	2.99	0.90	3.01	0.03	2.90	-0.69
Listed count	3.41	0.99	3.23	-2.80	4.36	6.37
Favorites count	3.10	1.06	3.16	0.91	2.77	-2.08
Statuses count	4.65	0.68	4.61	-0.84	4.84	1.91
Registration age	1846.82	869.99	1646.61	-3.50	2879.01	7.95

For user features carrying binary values, we conducted Chi-Square Goodness of Fit Tests. For one particular user feature, the **null hypothesis** is that there is no significant

Table 3.7 Results of Z-Test (*Twitter15*, Retweeters)

Feature	Retweeters		Fraudulent retweeters		Legitimate retweeters	
	Mean	Std.	Mean	z-score	Mean	z-score
Username length	10.91	5.29	10.64	-14.58	11.08	11.37
Screenname length	10.81	2.59	10.72	-9.99	10.86	7.79
Personal description length	63.27	53.96	62.18	-5.55	63.93	4.48
Followers count	2.63	0.66	2.69	25.33	2.59	-19.75
Friends count	2.69	0.53	2.70	4.53	2.60	-3.53
Listed count	1.39	0.43	1.38	-6.89	1.40	5.37
Favorites count	3.42	0.93	3.36	-15.90	3.45	12.40
Statuses count	3.92	0.82	4.02	36.16	3.85	-28.20
Registration age	1287.63	775.10	1111.68	-64.62	1394.26	50.38

Table 3.8 Results of Z-Test (*Weibo16*, Source Users)

Feature	Source users		Fraudulent source users		Legitimate source users	
	Mean	Std.	Mean	z-score	Mean	z-score
Username length	5.52	2.36	5.58	0.60	5.39	-1.2
Screenname length	5.52	2.36	5.58	0.60	5.39	-1.2
Personal description length	37.09	29.38	35.66	-2.06	42.66	4.11
Followers count	4.83	1.30	4.55	-8.95	5.90	17.86
Friends count	2.69	0.56	2.73	3.31	2.52	-6.62
Attitudes count	1.45	0.64	1.32	-8.21	1.93	16.37
Favorites count	1.79	0.74	1.77	-1.18	1.87	2.36
Statuses count	3.72	0.78	3.65	-3.77	3.99	7.52
Registration age	724.49	440.36	652.61	-6.94	1005.78	13.84

Table 3.9 Results of Z-Test (*Weibo16*, Retweeters)

Feature	Retweeters		Fraudulent retweeters		Legitimate retweeters	
	Mean	Std.	Mean	z-score	Mean	z-score
Username length	6.98	3.07	6.86	-63.19	7.14	73.62
Screenname length	6.98	3.07	6.86	-63.19	7.14	73.62
Personal description length	12.74	18.15	12.73	-0.58	12.76	0.68
Followers count	2.26	0.61	2.40	142.86	2.06	-166.43
Friends count	2.33	0.40	2.38	104.27	2.27	-121.47
Attitudes count	1.00	0.02	1.002	-3.89	1.005	4.53
Favorites count	1.62	0.66	1.66	70.47	1.56	-82.09
Statuses count	2.99	0.67	3.12	222.11	2.80	-258.47
Registration age	776.22	538.75	616.45	-462.13	993.03	534.04

difference between the distribution of this user feature among fake news spreaders (or ignorants) and that for the entire user population. χ^2 score is calculated by the following formula:

$$\chi^2 = \Sigma_i \frac{(E_i - O_i)^2}{E_i}, \tag{3.2}$$

where E_i, O_i are the expected counts and observed counts of users in one particular category. For each binary feature, users can be divided into two categories based on their feature values. A χ^2 score larger than 3.84 (critical threshold based on a p-value of 0.05) will reject the null hypothesis, i.e., indicating that there is a significant difference between the distribution of the concerned user feature for fake news spreaders (or ignorants) and that for the entire user population.

Tables 3.10-3.13 show the results of the Chi-Square Goodness of Fit Tests. From these tables, we can find that several binary user features distribute significantly differently between fraudulent source users and the entire source user population, legitimate source users and the entire source user population. However, most of the binary user features distribute significantly differently between fraudulent retweeters and the entire retweeter population, legitimate retweeters and the entire retweeter population.

Table 3.10 Results of Chi-Square Goodness of Fit Test (*Twitter15*, Source Users)

Feature	Source users		Fraudulent source users			Legitimate source users		
	O_1	O_2	O_1	O_2	χ^2	O_1	O_2	χ^2
Has location info.	214	63	176	56	0.25	38	7	1.32
Has personal URL	224	53	179	53	2.06	45	0	10.64
Are tweets protected	2	275	2	230	0.06	0	45	0.32
Is account verified	186	91	143	89	3.19	43	2	16.46
Is GEO enabled	141	136	121	111	0.14	20	25	0.75
Is language English	272	5	227	5	0.16	45	0	0.82
Is Contributors Enabled	0	277	0	232	NA	0	45	NA
Has profile background tile	97	180	80	152	0.03	17	28	0.15
Has profile background image	214	63	181	51	0.07	33	12	0.19
Has default profile	30	247	29	203	0.66	1	44	3.45
Has default profile image	0	277	0	232	NA	0	45	NA

From the results of our user feature study, we found that most user features distribute significantly different across fake news spreaders and the entire user population, as well as across fake news ignorants and the entire user population. These results indicate that whether a social media user is a fake news spreader or fake news ignorants can be reflected from his/her user characteristics. In the next section, we built a machine learning model to predict whether a user is a fake news spreader based on his/her user characteristics.

3.6 A Machine Learning Model to Predict a User's Tendency to Spread Fake News

Since we found that most user characteristics distribute significantly differently across fake news spreaders and fake news ignorants, we then built a machine learning model (a simple neural network with one hidden layer) to predict whether a user is a fake news spreader based on his/her user characteristics. Figure 3.1 shows its architecture.

Table 3.11 Results of Chi-Square Goodness of Fit Test (*Twitter15*, Retweeters)

Feature	Source users		Fraudulent source users			Legitimate source users		
	O_1	O_2	O_1	O_2	χ^2	O_1	O_2	χ^2
Has location info.	151366	64097	58216	23086	71.22	93150	41014	43.32
Has personal URL	62303	153160	24783	56519	97.09	37520	96644	58.91
Are tweets protected	12244	203219	4882	76420	15.74	7362	126802	9.55
Is account verified	2907	212556	1049	80253	2.12	1858	132306	1.28
Is GEO enabled	121878	93585	48062	33240	215.13	73816	60348	130.58
Is language English	189132	26331	73024	8278	315.05	116108	18056	191.52
Is Contributors Enabled	0	215463	0	81302	NA	0	134164	NA
Has profile background tile	61692	163771	29169	52133	2088.46	32523	101641	1265.95
Has profile background image	184857	30606	71462	9840	294.68	113395	20769	179.11
Has default profile	81423	134040	24026	57276	2347.11	57397	76767	1421.84
Has default profile image	3509	211954	820	80482	195.07	2689	131475	118.19

Table 3.12 Results of Chi-Square Goodness of Fit Test (*Weibo16*, Source Users)

Feature	Retweeters		Fraudulent retweeters			Legitimate retweeters		
	O_1	O_2	O_1	O_2	χ^2	O_1	O_2	χ^2
Is account verified	999	1310	687	1152	26.15	312	158	102.32
Is GEO enabled	1601	708	1304	535	0.45	297	173	8.35
Gender	828	1481	635	1204	0.32	193	277	5.53
Has location info.	2309	0	1839	0	0	470	0	NA

Table 3.13 Results of Chi-Square Goodness of Fit Test (*Weibo16*, Retweeters)

Feature	Retweeters		Fraudulent retweeters			Legitimate retweeters		
	O_1	O_2	O_1	O_2	χ^2	O_1	O_2	χ^2
Is account verified	101680	2716322	60559	1561865	72.18	41121	1154457	97.95
Is GEO enabled	2554957	263045	1465097	157327	252.02	1089860	105718	342.007
Gender	1535478	1282524	786947	835477	23425.52	748531	447047	31788.92
Has location info.	2818002	0	1622424	0	NA	1195578	0	NA

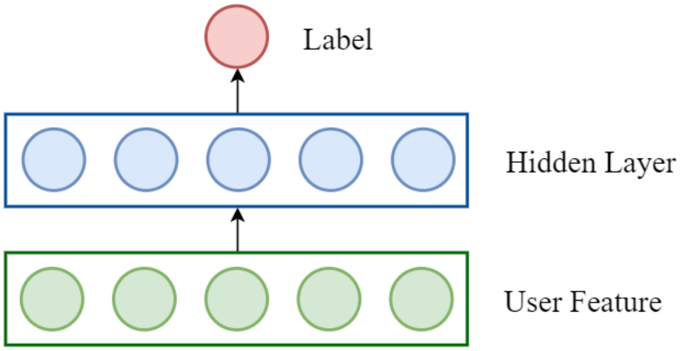


Figure 3.1 A two-layer neural network to predict a user's tendency to spread fake news.

The proposed model can be formulated as follows:

$$\hat{y}_i = \sigma\big(\mathbf{W}_2 \mathrm{Relu}(\mathbf{W}_1 \cdot \mathbf{x}_i + \mathbf{b}_1) + \mathbf{b}_2\big),$$

where $\sigma(\cdot)$ is the Sigmoid activation function, $\mathrm{Relu}(\cdot)$ is the ReLU activation function, $\mathbf{W}_1, \mathbf{b}_1$ are the weights and bias of the feature input layer, $\mathbf{W}_2, \mathbf{b}_2$ are the weights and bias of the hidden layer, $\mathbf{x}_i$ is the input user feature vector. In this model, we adopt all the user features in the user profiles that are included in our datasets because of the following two reasons: (1) Most user features distribute significantly differently across fake news spreaders and fake news ignorants. Therefore, they are highly discriminative; (2) Our neural network model can learn to select important features; (3) We have performed manual feature selection based on the results of hypothesis testing, but it could not improve the model's effectiveness compared with using all features. The parameters of our model are optimized to minimize the training loss.

Table 3.14 shows the performance of the proposed user classification model. From this table, we can find that our user classification model can predict whether a user is likely to be a fake news spreader with high accuracy in the three user groups except for retweeters in the Twitter15 dataset. We think that the main reason for the low classification effectiveness on retweeters in the Twitter15 dataset is because of the size of this dataset. Twitter15 dataset includes much fewer retweeters (roughly 10%) than Weibo16 does.

Table 3.14 Performance of the Proposed User Classification Model

User Group	Accuracy	Precision	Recall	F1 score
Source user (Twitter15)	0.91	0.91	0.99	0.95
Retweeter (Twitter15)	0.72	0.68	0.51	0.58
Source user (Weibo16)	0.87	0.88	0.97	0.92
Retweeter (Weibo16)	0.82	0.83	0.87	0.85

3.7 Discussion

The performances of our machine learning model to predict whether a user is likely to spread fake news are acceptable except for the performance for detecting fake news retweeters in the Twitter15 dataset. The low performance of the user classification on the retweeters in the Twitter15 dataset is likely due to the small size of this dataset.

Overall, these results give us several implications on how to build a fake news detection model that utilizes user characteristics of news spreaders to detect fake news early:

(1) User characteristics of news spreaders can be potentially useful for fake news early detection since they are readily available at the early stage of news propagation. On the contrary, other social context data, such as user comments can be very few at the early stage of news propagation because users can retweet a news article without posting any comments. In this case, those detection approaches' effectiveness will be affected. However, for those users who retweet a news article without any comments, their user profiles and user characteristics are already available, which provide an important source of data for early detection.

(2) The combination of user characteristics with other social context data, e.g., user comments, might give us more insight into whether a news article is fake than a single source of social context data alone. For example, a user comment "I believe this is true" posted by a user who has never spread fake news and the same comment posted by another

user who has spread some fake news pieces might give us an entirely different clue about whether the concerned news is fake. Although using user characteristics to predict an individual user's tendency to spread fake news did not achieve high effectiveness in all four user groups in our experiments reported in Section 3.6., we believe that our user classification model's performance can be further improved given a larger experimental dataset. Since fake news is often intentionally spread by a group of malicious users, their user profiles need to be paid with additional attention in the process of fake news detection by some approach similar to our user classification model.

(3) User characteristics are harder to be manipulated, thus more reliable for detecting fake news compared with other social context features. For instance, it is more expensive for fake news producers to buy a lot of social media accounts with user profiles similar to normal users' to spread fake news than simply creating a lot of new accounts to post comments to their fake news.

Based on these implications, we build several detection models which will be discussed in the next few chapters.

3.8 Summary

In this chapter, we first explored what user characteristics are included in Twitter and Weibo platforms. We then investigated their distribution among different user groups and found that many user characteristics distribute significantly differently across fake news spreaders and fake news ignorants. Thus, these results gave us an implication that the user characteristics of the source user and the retweeters of a news article might be used to predict whether it is fake news, which led to our proposed fake news detection models.

CHAPTER 4

EARLY DETECTION OF FAKE NEWS ON SOCIAL MEDIA THROUGH PROPAGATION PATH CLASSIFICATION

From Chapter 3, we know that user characteristics distribute significantly differently among fake news spreaders and normal users, which provides us a theoretical foundation for utilizing user characteristics to detect fake news early. Starting in this chapter, we build fake news detection models that rely on user characteristics of news spreaders. This chapter introduces the first proposed deep learning-based model named *Propagation Path Classification (PPC)*. The PPC model combines a recurrent and a convolutional network to classify propagation paths formed by sequences of news spreaders. The details of the PPC model will be presented in the remainder of this chapter.

4.1 Problem Statement

Let $\mathcal{A} = \{a_1, a_2, \dots, a_{|\mathcal{A}|}\}$ be a set of news stories, $\mathcal{U} = \{u_1, u_2, \dots, u_{|\mathcal{U}|}\}$ be a set of social media users. Each user $u_j \in \mathcal{U}$ is associated with a *user vector* $\mathbf{x}_j \in \mathbb{R}^d$, which represents the characteristics of the user. We define the *propagation path* of a given news object a_i as a *variable-length multivariate time series* $\mathcal{P}(a_i) = \langle \dots, (\mathbf{x}_j, t), \dots \rangle$, in which each tuple $(\mathbf{x}_j, t)$ denotes that user u_j tweets/retweets the news object a_i at time t. In this chapter, we set the time of a source tweet being posted to 0. Thus, $t > 0$ refers to the time of a retweet being posted. Each news object a_i is associated with a label $L(a_i)$ that reflects its truthfulness. Each label $L(a_i) \in \{0,1\}^r$. When $r = 1$, $L(a_i) = 0$ denotes the news object a_i is true, and $L(a_i) = 1$ denotes a_i is fake. When $r > 1$, the label $L(a_i)$ is a categorical variable that reflects multiple levels of the truthfulness of the news object a_i, e.g., true, fake, or unverified, etc. Our goal is to design a model f that can predict the label of a given news object a_i based on its propagation path $\mathcal{P}(a_i)$, i.e., $\hat{L}(a_i) = f\big(\mathcal{P}(a_i)\big)$.

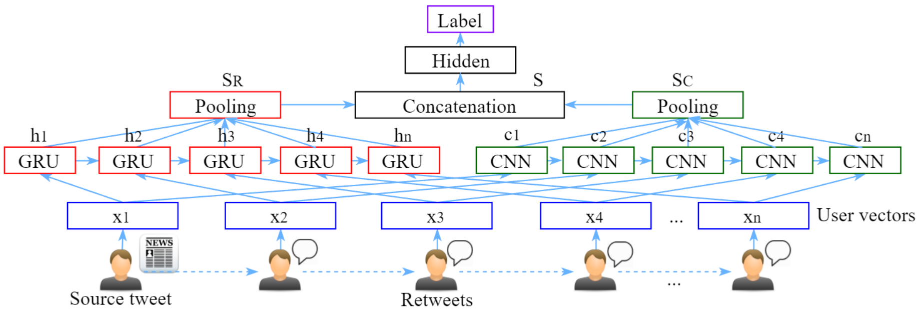


Figure 4.1 Architecture of the proposed fake news detection model.

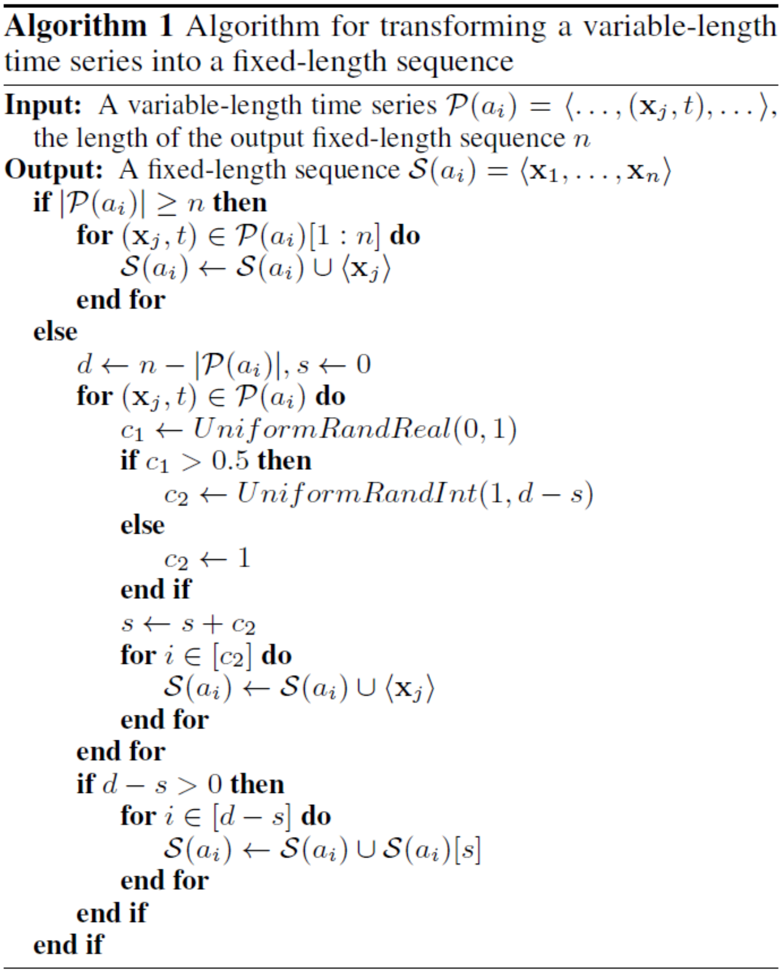
Since we aim to detect fake news as early as possible after it starts to spread, our model should be able to make predictions based on only a partial propagation path observed in the early stage of news propagation. We define the *partial propagation path* of a given news object a_i as $\mathcal{P}(a_i, T) = \langle(\mathbf{x}_j, t < T)\rangle$, where T is a *detection deadline* after which all the observed data cannot be used in detecting fake news. We call the task of predicting the truthfulness of news stories given partial propagation paths as *early detection of fake news*. In this case, we aim to design a model f_T that predicts the label of a given news object a_i based on its partial propagation path, i.e., $\hat{L}(a_i) = f_T\big(\mathcal{P}(a_i, T)\big)$.

4.2 The Proposed Model

The proposed fake news detection model consists of four major components, i.e., propagation path construction and transformation, RNN-based propagation path representation, CNN-based propagation path representation, and propagation path classification, which are integrated together to detect fake news at the early stage of its propagation. Figure 4.1 shows the architecture of the proposed model. Next, we will introduce each of the major components in detail.

4.2.1 Propagation Path Construction and Transformation

Given a news object propagating on social media, we first construct its propagation path by first identifying the users who engaged in propagating the news. Then, its propagation path denoted as a *variable-length multivariate time series* $\mathcal{P}(a_i) = \langle \dots, (\mathbf{x}_j, t), \dots \rangle$ is constructed by extracting user characteristics from relevant user profiles. After $\mathcal{P}(a_i)$ is obtained, we transform it into a *fixed-length multivariate sequence*, denoted as $\mathcal{S}(a_i) = \langle \mathbf{x}_1, \dots, \mathbf{x}_n \rangle$, where n is the length of the sequence. If there are more than n tuples in $\mathcal{P}(a_i)$, then $\mathcal{P}(a_i)$ will be truncated so that only the first n tuples will appear in $\mathcal{S}(a_i)$; If $\mathcal{P}(a_i)$ contains less than n tuples, then we randomly oversample tuples in $\mathcal{P}(a_i)$ to ensure the final length of $\mathcal{S}(a_i)$ equals n. Figure 4.2 shows the algorithm of transforming a variable-length multivariate time series into a fixed-length multivariate sequence.



```
**Algorithm 1** Algorithm for transforming a variable-length time series into a fixed-length sequence
**Input:** A variable-length time series $\mathcal{P}(a_i) = \langle \dots, (\mathbf{x}_j, t), \dots \rangle$,
  the length of the output fixed-length sequence $n$
**Output:** A fixed-length sequence $\mathcal{S}(a_i) = \langle \mathbf{x}_1, \dots, \mathbf{x}_n \rangle$
  if $|\mathcal{P}(a_i)| \geq n$ then
      for $(\mathbf{x}_j, t) \in \mathcal{P}(a_i)[1:n]$ do
          $\mathcal{S}(a_i) \leftarrow \mathcal{S}(a_i) \cup \langle \mathbf{x}_j \rangle$
      end for
  else
      $d \leftarrow n - |\mathcal{P}(a_i)|, s \leftarrow 0$
      for $(\mathbf{x}_j, t) \in \mathcal{P}(a_i)$ do
          $c_1 \leftarrow UniformRandReal(0, 1)$
          if $c_1 > 0.5$ then
              $c_2 \leftarrow UniformRandInt(1, d - s)$
          else
              $c_2 \leftarrow 1$
          end if
          $s \leftarrow s + c_2$
          for $i \in [c_2]$ do
              $\mathcal{S}(a_i) \leftarrow \mathcal{S}(a_i) \cup \langle \mathbf{x}_j \rangle$
          end for
      end for
      if $d - s > 0$ then
          for $i \in [d - s]$ do
              $\mathcal{S}(a_i) \leftarrow \mathcal{S}(a_i) \cup \mathcal{S}(a_i)[s]$
          end for
      end if
  end if
```

Figure 4.2 Algorithm for transforming a variable-length time series into a fixed-length sequence

4.2.2 RNN-Based Propagation Path Representation

We utilize a variant of RNN called *Gated Recurrent Unit (GRU)* (Chung, Gulcehre, Cho, & Bengio, 2014) to learn a vector representation for each transformed propagation path, i.e., $\mathcal{S}(a_i)$. For the t^{th} user vector in $\mathcal{S}(a_i)$, i.e., $\mathbf{x}_t$, a GRU unit takes as input $\mathbf{x}_t, \mathbf{h}_{t-1}$ and produces $\mathbf{h}_t$ as output according to the following formulas:

$$\begin{aligned}
\mathbf{z}_t &= \sigma(U_z\mathbf{x}_t + W_z\mathbf{h}_{t-1}) \\
\mathbf{r}_t &= \sigma(U_r\mathbf{x}_t + W_r\mathbf{h}_{t-1}) \\
\tilde{\mathbf{h}}_t &= \tanh(U_h\mathbf{x}_t + \mathbf{h}_{t-1} \odot W_h\mathbf{r}_t) \\
\mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t
\end{aligned} \tag{4.1}$$

where $U_z, U_r, U_h \in \mathbb{R}^{m\times d}, W_z, W_r, W_h \in \mathbb{R}^{m\times m}$ are weight matrices, d is the dimension of the user vector $\mathbf{x}_t$, and m is the output dimension of the GRU units. The symbols $\sigma(\cdot)$ and $\tanh(\cdot)$ denote the element-wise sigmoid and hyperbolic tangent functions, respectively, $\odot$ denotes the element-wise vector multiplication operation. $\mathbf{h}_0 = \mathbf{0}$. We then apply mean pooling to reduce the sequence of output vectors $\langle \mathbf{h}_1, \ldots, \mathbf{h}_n \rangle$ produced by GRU units into a single vector $\mathbf{s}_R = \frac{1}{n}\sum_{t=1}^{n} \mathbf{h}_t$, which is the final vector representation of $\mathcal{S}(a_i)$ that encodes the global variation of user characteristics.

4.2.3 CNN-Based Propagation Path Representation

We also use convolutional networks (CNN) to learn another vector representation for each $\mathcal{S}(a_i)$. We first apply a 1-D convolution on h consecutive user vectors, i.e., $\langle \mathbf{x}_t, \ldots, \mathbf{x}_{t+h-1} \rangle$ with a *filter* $W_f \in \mathbb{R}^{h\times m}$ of height h, to produce a *scalar feature* $c_t \in \mathbb{R}$ according to the following formula:

$$c_t = \mathrm{ReLU}(W_f \cdot X_{t:t+h-1} + b_f) \tag{4.2}$$

where $X_{t:t+h-1} \in \mathbb{R}^{h\times m}$ is the matrix whose i^{th} row is $\mathbf{x}_i$ and $b_f \in \mathbb{R}$ is a bias. The symbol $\mathrm{ReLU}(\cdot)$ refer to the element-wise rectified linear unit function. We perform the same convolution operation with k filters to produce a multivariate feature vector $\mathbf{c}_t \in \mathbb{R}^k$. By repeating the same convolution operations for each window of h consecutive user vectors, we obtain a sequence of multivariate feature vectors, i.e., $\langle \mathbf{c}_1, \ldots, \mathbf{c}_{n-h+1} \rangle$. Then, we apply mean pooling to produce a final vector representation of $\mathcal{S}(a_i)$, i.e., $\mathbf{s}_C = \frac{1}{n} \sum_{t=1}^{n-h+1} \mathbf{c}_t$ that encodes the local variation of user characteristics.

4.2.4 Propagation Path Classification

After $\mathbf{s}_R \in \mathbb{R}^m, \mathbf{s}_C \in \mathbb{R}^k$ are obtained through RNNs and CNNs, they are concatenated into a single vector that represents the transformed propagation path, i.e. $\mathbf{s} \in \mathbb{R}^{m+k}$ by the following formula:

$$\mathbf{s} = \text{Concatenate}(\mathbf{s}_R, \mathbf{s}_C) \tag{4.3}$$

which is then fed into a multi-layer feedforward neural network that finally predicts the class label for the corresponding propagation path by the following formulas:

$$\begin{aligned} \mathbf{l}_j &= \text{ReLU}(W_j \mathbf{l}_{j-1} + \mathbf{b}_j),\ \forall j \in [q] \\ \mathbf{z} &= \text{Softmax}(\mathbf{l}_q) \end{aligned} \tag{4.4}$$

where q is the number of hidden layers, $\mathbf{l}_j \in \mathbb{R}^{v_j}$ is the output of the j^{th} hidden layer ($\mathbf{l}_0 = \mathbf{s}$), v_j is the output dimension for the j^{th} hidden layer, $W_j \in \mathbb{R}^{v_j \times v_{j-1}}, \mathbf{b}_j \in \mathbb{R}^{v_j}$ are the weight matrix and bias for the j^{th} hidden layer, and $\mathbf{z} \in \mathbb{R}^r$ is the final output that represents the probability distribution over the set of r classes for the corresponding propagation path. We adopt both RNN and CNN to extract different aspects of latent features from a propagation path and then combine those features by concatenating the two intermediate feature vectors produced by them, respectively. This idea is also implemented in a previous study (Lee & Dernoncourt, 2016) that combines RNN and CNN to classify short sentences.

4.3 Experiments

4.3.1 Datasets

We used the same datasets described in Chapter 3 to evaluate our model and the baselines. However, in this chapter, we also show the results conducted on the *Twitter16* dataset mentioned in Chapter 3.

4.3.2 Baseline Models

We compare our model with a series of baseline fake news detection models as follows:

- DTC (Castillo et al., 2011) A decision-tree-based model that utilizes a combination of news characteristics.
- SVM-RBF (Yang et al., 2012) An SVM model with RBF kernel that utilize a combination of news characteristics.
- SVM-TS (Ma et al., 2015) An SVM model that utilizes time-series to model the variation of news characteristics.
- DTR (Zhao, Resnick, & Mei, 2015) A decision-tree-based ranking method for detecting fake news through enquiry phrases.
- GRU (Ma et al., 2016) An RNN-based model that learns temporal-linguistic patterns from user comments.
- RFC (Kwon et al., 2017) A random forest classifier that utilizes user, linguistic and structure characteristics.
- PTK (Ma et al., 2017) An SVM classifier with a propagation tree kernel that detects fake news by learning temporal-structure patterns from propagation trees.

We denote our proposed model as "PPC" (Propagation Path Classification), also as "PPC_RNN+CNN". We also implement two reduced version of the proposed model which only utilizes RNNs or CNNs alone, denoted as "PPC_RNN" and "PPC_CNN", respectively.

Table 4.1 Model Configuration

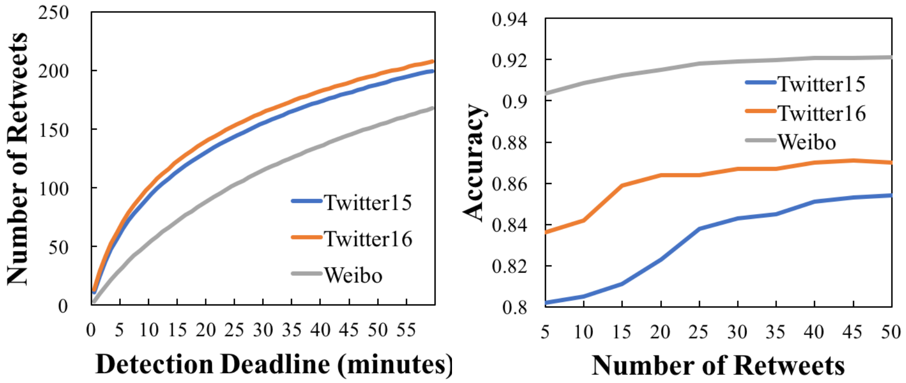
Hyperparameter	**Choice**	**Experimental Range**
GRU output dim	32	8 - 64
CNN # filters	32	8 - 64
CNN filter height	3	1 - 10
Dropout rate	0.5	0 - 1

4.3.3 Model Configuration

We implemented our proposed model by using Keras[1]. The model is trained to minimize the binary/categorical loss function of predicting the class label of news stories in the training set. The weights and biases are updated using stochastic gradient descent with the Adadelta update rule (Zeiler, 2012). Dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) is applied to hidden layers above the concatenation layer to avoid overfitting. We set the number of training epochs to be 200. Early stop is applied when the validation loss peaks for ten epochs. The network structure and hyperparameters are set based on the performance of our model on the validation set, which is shown in Table 4.1.

Note that the sequence length n used in Algorithm 1, which is also the number of source tweets plus the number of retweets we need to observe in a news propagation path to detect fake news, is explored to investigate both a) the overall optimal effectiveness and b) effectiveness of ***early*** detection of fake news. A longer sequence length might improve the overall optimal effectiveness of fake news detection since more data will be observed. However, early detection efficiency will be affected since it requires a longer time to observe a longer propagation path than a shorter one. On the other hand, a framework that requires only shorter sequence length improves the efficiency of early detection of fake news, since it means less amount of data, and thus time too, required to make a prediction. However, the effectiveness might be affected in this case. Therefore, we

[1] https://keras.io/



(a) propagation speed

(b) detection speed

Figure 4.3 News propagation speed and fake news detection speed

need to balance the trade-off between optimal detection effectiveness and early detection effectiveness by choosing the most appropriate sequence length. Figure 4.3 shows the speed of news propagation on social media and the speed of fake news detection conducted by our proposed model with both recurrent and convolutional networks. Figure 4.3-(b) shows that the accuracy of our proposed model in detecting fake news peaks when the required number of retweets, i.e., the sequence length, is above 40 in the *Twitter15* and *Twitter16* datasets, and above 30 in the *Weibo* dataset, respectively. Figure 4.3-(a) shows that it requires about 5 minutes to observe 40 retweets in the *Twitter15* and *Twitter16* datasets and 30 retweets in the *Weibo* dataset. Therefore, when we observe more than 40 retweets on Twitter and more than 30 retweets on Weibo, our proposed model can detect fake news with accuracy around 85% and 92% on Twitter and Weibo, respectively, within five minutes after it starts to spread. This detection speed is significantly faster than manual fact-checking.

4.3.4 Results

Tables 4.2, 4.3, and 4.4 show the performance of the proposed model and that of the baseline models in the task of fake news detection on Twitter15, Twitter16, and Weibo dataset, respectively. For most of the baseline models, their performance peaks when the detection deadline is above 24 hours. Therefore, to make a fair comparison, we set the detection deadline to 24 hours here. We can find that the proposed models, i.e., PPC_RNN, PPC_CNN, and PPC_RNN+CNN outperform the baseline models. Among them, PPC_RNN+CNN performs the best. It achieves 84.2%, 86.3%, 92.1% accuracy on *Twitter15*, *Twitter16*, and *Weibo* dataset, respectively. Based on these results, we can find that when observing relatively complete propagation paths, the proposed model outperforms the baseline models slightly in terms of optimal effectiveness.

In the previous studies that introduce the peer models, a detection deadline of 24 hours is considered to be early. However, we aim to detect fake news as early as possible so that its harmful effects can be minimized. Therefore, we carefully investigate the performance of all the models in detecting fake news in less than 24 hours after it starts to spread. Figure 4.4 shows the results of early detection of fake news. Among all the baseline models, we select three recent ones that have reported results on early detection of fake news, namely, DTR, GRU, and PTK. DTR and GRU rely on linguistic features extracted from user comments, while PTK relies on both linguistic and structural features extracted from propagation trees. We can find that when the detection deadline is less than 24 hours, the performance of the baseline models decreases significantly, while the performance of the proposed model is not affected since it only requires the first five minutes' data to make accurate predictions. Among the three baseline models, DTR yields the worst performance, because the number of inquiry posts is usually very small in the early stage of news propagation. PTK yields better performance than GRU because it utilizes temporal-structural features besides of temporal-linguistic features.

Table 4.2 Detection Performances on *Twitter15* Dataset When the Detection Deadline is 24 hours (“T”: True News; “F”: Fake News; “U”: Unverified News; “D”: Debunking of Fake News)

Method	**Acc.**	**T**	**F**	**U**	**D**
		$\mathbf{F_1}$	$\mathbf{F_1}$	$\mathbf{F_1}$	$\mathbf{F_1}$
DTC	0.454	0.733	0.355	0.317	0.415
SVM-RBF	0.318	0.455	0.037	0.218	0.225
SVM-TS	0.544	0.796	0.472	0.404	0.483
DTR	0.409	0.501	0.311	0.364	0.473
GRU	0.646	0.792	0.574	0.608	0.592
RFC	0.565	0.810	0.422	0.401	0.543
PTK	0.750	0.804	0.698	0.765	0.733
PPC_RNN	0.811	0.759	0.842	0.765	0.787
PPC_CNN	0.803	0.737	0.835	0.751	0.775
PPC_ RNN+CNN	**0.842**	**0.811**	**0.875**	**0.790**	**0.818**

Table 4.3 Detection Performances on *Twitter16* Dataset When the Detection Deadline is 24 hours (“T”: True News; “F”: Fake News; “U”: Unverified News; “D”: Debunking of Fake News)

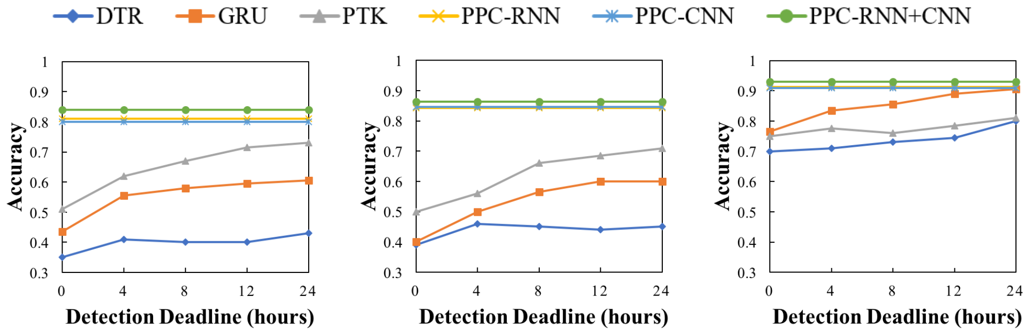
Method	**Acc.**	**T**	**F**	**U**	**D**
		$\mathbf{F_1}$	$\mathbf{F_1}$	$\mathbf{F_1}$	$\mathbf{F_1}$
DTC	0.465	0.643	0.393	0.419	0.403
SVM-RBF	0.321	0.423	0.085	0.419	0.037
SVM-TS	0.574	0.755	0.420	0.571	0.526
DTR	0.414	0.394	0.273	0.630	0.344
GRU	0.633	0.772	0.489	0.686	0.593
RFC	0.585	0.752	0.415	0.547	0.563
PTK	0.732	0.740	0.709	0.836	0.686
PPC_RNN	0.842	0.809	0.865	0.836	0.839
PPC_CNN	0.847	0.812	0.871	0.833	0.841
PPC_RNN+CNN	**0.863**	**0.820**	**0.898**	**0.837**	**0.843**

Table 4.4 Detection Performances on *Weibo* Dataset when the Detection Deadline is 24 hours ("F": Fake News; "T": True News)

Method	**Class**	**Acc.**	**Precision**	**Recall**	**F_1**
DTC	F	0.831	0.847	0.815	0.831
	T		0.815	0.847	0.830
SVM-RBF	F	0.818	0.822	0.812	0.817
	T		0.815	0.824	0.819
SVM-TS	F	0.857	0.839	0.885	0.861
	T		0.878	0.830	0.857
DTR	F	0.732	0.738	0.715	0.726
	T		0.726	0.749	0.737
GRU	F	0.910	0.876	0.956	0.914
	T		0.952	0.864	0.906
RFC	F	0.849	0.786	0.959	0.864
	T		0.947	0.739	0.830
PPC_RNN	F	0.912	0.878	0.958	0.916
	T		0.944	0.866	0.908
PPC_CNN	F	0.919	0.889	0.958	0.922
	T		0.946	0.880	0.916
PPC_ RNN+CNN	F	**0.921**	**0.896**	**0.962**	**0.923**
	T		**0.949**	**0.889**	**0.918**

4.3.5 Discussion

As pointed out by a recent study (Kwon et al., 2017), structural and temporal features are useful for detecting fake news after observing propagation data over a certain amount of time. However, they are less useful for early detection, since the propagation data is often



(a) Twitter15 (b) Twitter16 (c) Weibo

Figure 4.4 Results of early detection of fake news.

insufficient in the early stage of news propagation. From the results of our user feature studies in Chapter 3, we found that user features can, to some extent, indicate whether a user is a fake news spreader. By contrast, linguistic features are less available than user characteristics at the very beginning of news propagation, e.g., in the first five minutes. This is because there are often very few user comments that can be observed shortly after a news object is posted. Therefore, we assume that our model is more effective on early detection of fake news than baseline models since it only relies on user characteristics. Experimental results on three real-world datasets demonstrate that the proposed model can significantly improve early detection effectiveness while slightly improving optimal detection effectiveness given sufficient data. We also find that the two reduced models that only incorporate RNNs or CNNs yield similar accuracy results, which are still higher than those of baseline models but lower than the accuracy of the complete proposed model that combines RNNs and CNNs. This demonstrates that both recurrent networks and convolutional networks can capture the global and local variations of user characteristics, respectively. However, it is better to combine them to capture both the global and local variations of user characteristics to achieve the best performance of early detection.

Although the PPC model performs well on early detection, it can be potentially further improved from the following aspect. When the observed news propagation path is extremely short, i.e., very few retweeters are observed, the sequential information will not be very discriminative in differentiating fake news from true news. In this case, we can utilize the source users' characteristics more since, in Chapter 3, our user classification model performs particularly well on the group of source users. Therefore, we then proposed an improved deep learning-based model named *Social Media Content Classification (SMCC)*. The SMCC model incorporates a fake news spreader likelihood score that models the probability of a source user to spread fake news. This mechanism significantly improves the early detection performance when very few retweeters are observed. We will introduce our improved model in the next chapter.

4.4 Summary

In this study, we proposed a novel model for early detection of fake news on social media through classifying news propagation paths with both recurrent and convolutional networks. After modeling the new propagation paths as multivariate time series of user characteristics, we apply recurrent and convolutional networks to capture both global and local variations of user characteristics along propagation paths to detect fake news. Experimental results on three real-world datasets demonstrate that our proposed model outperforms state-of-the-art fake news detection approaches in terms of both effectiveness and efficiency. Since our model only relies on common user characteristics which are more available, reliable and robust than complex features such as linguistic or structural features that are widely used in state-of-the-art baseline approaches, it can detect fake news significantly faster than state-of-the-art baselines, e.g., in five minutes after the fake news starts to spread.

CHAPTER 5

A NOVEL DEEP LEARNING MODEL NAMED SOCIAL MEDIA CONTENT CLASSIFICATION (SMCC) AND ITS USAGE IN FAKE NEWS EARLY DETECTION

Compared with common retweeters, source users, i.e., users who initially post a news article on social media, often play a more critical role in fake news propagation. In Chapter 3, our user classification model performs particularly well on the group of source users. Therefore, source users' user characteristics contain more information about the truthfulness of a news article than common retweeters' user characteristics, thus need to be paid with additional attention in a fake news detection model. In this chapter, we present the details of the second and improved proposed deep learning model named *Social Media Content Classification (SMCC)*, which incorporates a fake news spreader likelihood score to highlight the source user in a news propagation path.

5.1 Preliminaries and Problem Statement

In this section, we first introduce some preliminaries used in this chapter and then revise the definition of the problem of fake news detection to make it more suitable in the context of this chapter. We adopt some terminologies on Twitter, such as "tweet" and "retweet", to discuss the context of our problem. We use italic lowercase characters (a) for scalar variables, italic uppercase characters (S) for sets and functions, bold lowercase characters ($\mathbf{x}$) for vectors, and bold uppercase characters ($\mathbf{X}$) for matrices.

Let $A = \{a_1, a_2, \ldots, a_{|A|}\}$ be a set of news objects, $U = \{u_1, u_2, \ldots, u_{|U|}\}$ be a set of social media users. Each news object a_i is first tweeted by a source user $S(a_i) \in U$, and then be retweeted by a set of retweeters $R(a_i) = \{u_j \in U\}$ by a certain time point which we call the "detection deadline". Each news object a_i is associated with a label $L(a_i) \in \{0, 1\}$, where $L(a_i) = 0$ when a_i is true news, and $L(a_i) = 1$ when it is fake news.

Each user u_i is associated with a feature vector $\mathbf{x}_i$, which represents the characteristics of that user. The feature vectors of all users form a feature matrix $\mathbf{X}$. The task of fake news detection is to predict a label $\hat{L}(a_i) \in \{0, 1\}$ for each news object $a_i \in A$. To achieve this goal, we propose a novel tweet classification model called SMCC to detect fake news based on user characteristics of its source user and retweeters, which is formulated as:

$$\hat{L}(a_i) = F\big(S(a_i), R(a_i), \mathbf{X}\big).$$

5.2 The Proposed Model

In this section, we present the details of the proposed SMCC model that is shown in Figure 5.1.

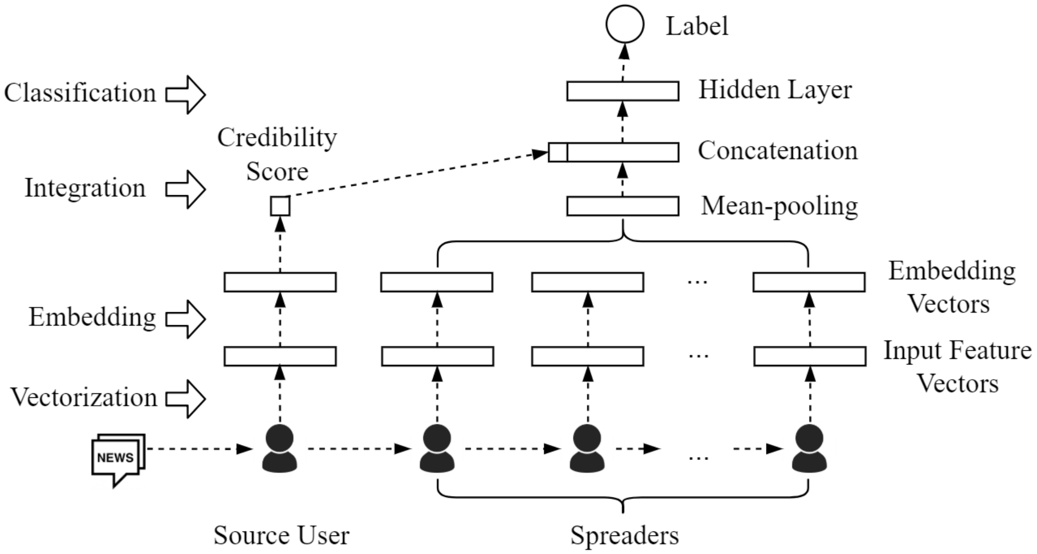


Figure 5.1 Architecture of the proposed SMCC model.

We design this model based on neural networks and deep learning (LeCun et al., 2015). Note that the SMCC model is not restricted within the domain of fake news detection. It also provides a general framework of classifying user-generated content on social media. SMCC is a neural network with four groups of layers, i.e., Input, Embedding, Integration, and Classification.

5.2.1 Input

The Input layer group has only one layer that is the input of SMCC, which consists of a certain number of the vector representations of news spreaders. Given a news object a_i, we first identify its source user $S(a_i) = u_s$, and its retweeters $R(a_i) = \{u_1, u_2, \dots, u_n\}$. Then, we represent each user $u_j \in S(a_i) \cup R(a_i)$ as a feature vector $\mathbf{x}_j$. The above two steps can be easily implemented via social media APIs, such as *Twitter API*[1] and *Weibo API*[2]. We construct user feature vectors by extracting a list of user features from user profiles supported by certain social media platforms. After this vectorization step, the input of the model will be a set of vectors $(\mathbf{x}_s, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$.

5.2.2 Embedding

The Embedding layer group has one or multiple layers. It transforms raw vector representations of news spreaders into latent (embedding) vector representations. In the context of machine learning, embedding generally refers to the process of transforming original vector representations of data instances into latent vector representations, typically for the purpose of dimension reduction. Embedding is widely used in machine learning models for natural language processing (Goldberg & Levy, 2014; Levy & Goldberg, 2014), graph modeling (Yan, Xu, Zhang, & Zhang, 2005), protein analytics (Shi, Liu, Perez, & Taylor, 2005), etc. We adopted embedding in our SMCC model because of the following two reasons: (i) Embedding can increase the neural network's learning ability and its robustness against noise in raw features; (ii) Embedding guarantees our model's flexibility. The types and dimensions of user features might vary across social media platforms. By transforming variable-length raw user feature vectors into fixed-length embedding vectors, other modules of our SMCC model do not need to change when being applied to different social media platforms.

[1] https://developer.twitter.com

[2] https://open.weibo.com

By embedding, each raw feature vector $\mathbf{x}_i$ is transformed into a latent feature vector $\mathbf{h}_i$ by the following formula:

$$\mathbf{h}_i = \mathrm{ReLu}(\mathbf{W}_e\mathbf{x}_i + \mathbf{b}_e),$$

where $\mathbf{W}_e, \mathbf{b}_e$ are the parameters of the embedding layer, and ReLu denotes the standard Relu function. After embedding, the input of the model will be transformed into a set of vectors $(\mathbf{h}_s, \mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n)$.

5.2.3 Integration

The Integration layer group has one or multiple layers. It integrates all embedding vectors into one single vector that represents the concerned news being spread. Before being fed to the classification layer, a set of embedding vectors will be aggregated into one single vector by the following steps. First, we produce a *fake news spreader likelihood score* $c_s \in [0, 1]$ for each source user by the following formula:

$$c_s = \mathrm{Sigm}(\mathbf{W}_c\mathbf{x}_s + \mathbf{b}_c),$$

where $\mathbf{W}_c, \mathbf{b}_c$ are the parameters of the fake news spreader likelihood scoring layer, Sigm denotes the standard Sigmoid function. The fake news spreader likelihood score measures the prior probability of a source user to produce fake news. It is calculated based on the characteristics of the source user. The reason to apply a fake news spreader likelihood score in our model will be discussed below in this section.

Next, we aggregate the embedding vectors of all the retweeters by mean-pooling, i.e.:

$$\bar{\mathbf{h}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{h}_i.$$

At last, we produce a final aggregated vector $\tilde{\mathbf{h}}$ by concatenating the source user's fake news spreader likelihood score c_s with the aggregated vector $\bar{\mathbf{h}}$ by the following formula:

$$\tilde{\mathbf{h}} = c_s \| \bar{\mathbf{h}},$$

where $||$ denotes the concatenation operation.

We design the above integration procedure due to the following reasons: (i) Source users who initially tweet news objects should be treated separately from retweeters. Compared with retweeters, source users often play a different role in spreading news on social media and have a different range of user features. For instance, based on the statistics of our experimental datasets in Chapter 3, we found that on average, a Twitter source user has 5,400 followers, while a Twitter news spreader has 2600 followers. Thus, in our SMCC model, we do not aggregate the feature vector of a source user and that of retweeters together. (ii) Source users can, to some extent, reflect the truthfulness of its news but should not dominate the model's output. The reason is that some malicious fake news producers mix a small amount of fake news with a large amount of true news to make their fake news harder to be distinguished. If the output of our model is mainly determined by the source users' characteristics, then it will be difficult for our model to detect fake news posted by the special type of fake news producers mentioned above. To address this issue, in our SMCC model, we first produce a fake news spreader likelihood score, which can be regarded as a prior probability of a news object tweeted by a certain user to be fake. It is intuitive to assume that if a user has a history of tweeting fake news objects, then the next news object he/she tweeted will have a higher prior probability of being fake. Recall in Chapter 3, we have shown that a user's tendency to spread fake news can be predicted based on his/her user characteristics. Thus, the fake news spreader likelihood score is an integration of that user model in Chapter 3 with the fake news classification model proposed in this chapter. However, the truthfulness of a news object is still mainly dependent on its retweeters since the source user's fake news spreader likelihood score only accounts for one element in the final aggregated vector. (iii) The output of the SMCC model should be robust enough against malicious manipulation. It is easy to edit a source user's profile or let a user who never tweeted fake news to tweet fake news once. However, it is much harder to control who will retweet a piece of fake news. When the number of news spreaders accumulates, it will be even harder to cheat our model by manipulating retweeters, since after a news object is posted, all users who read it can quickly retweet it.

5.2.4 Classification

The Classification layer group has one or multiple layers. It outputs a binary label that indicates whether the concerned news being spread is fake. Given the aggregated vector $\tilde{\mathbf{h}}$, our model first fed it to a hidden layer, then produces the output label by the following formulas:

$$\tilde{\mathbf{h}}' = \mathrm{ReLu}(\mathbf{W}_h\tilde{\mathbf{h}} + \mathbf{b}_h),$$

$$y(a_i) = \mathrm{Sigm}(\mathbf{W}_o\tilde{\mathbf{h}}' + \mathbf{b}_o),$$

where $\tilde{\mathbf{h}}'$ is the intermediate vector produced by the hidden layer, $\mathbf{W}_h, \mathbf{b}_h$ are the parameters of the hidden layer, $\mathbf{W}_o, \mathbf{b}_o$ is the parameters of the output layer, and y is the output label.

The loss function of training the proposed SMCC model is formulated as:

$$\mathcal{L} = -\frac{1}{|A|}\sum_{i=1}^{|A|}\big(L(a_i)\log y(a_i) + (1 - L(a_i))\log(1 - y(a_i))\big) + \lambda\|\Theta\|^2,$$

where $\Theta = \{\mathbf{W}_e, \mathbf{b}_e, \mathbf{W}_c, \mathbf{b}_c, \mathbf{W}_h, \mathbf{b}_h, \mathbf{W}_o, \mathbf{b}_o\}$ denotes all the parameters of the model and λ is a regularization factor.

5.3 Experiments and Results

5.3.1 Dataset

We still used the datasets introduced in Chapter 3 to evaluate our proposed model's and the baselines' detection effectiveness.

5.3.2 Baseline Approaches

We compared our proposed SMCC model with a series of baseline models discussed in Chapter 2, including:

- DTC ((Castillo et al., 2011)) A decision tree model that detects fake news based on aggregated news characteristics.
- SVM-TS ((Ma et al., 2015)) An SVM model that detects fake news based on time-series of aggregated news characteristics.
- GRU ((Ma et al., 2016)) An RNN model that detects fake news based on temporal-linguistic patterns recognized from sequences of user comments.
- PTK ((Ma et al., 2017)) An SVM model with a tree kernel that detects fake news based on structural patterns of news' propagation trees.
- CSI ((Ruchansky et al., 2017)) A hybrid deep learning model that detects fake news based on features extracted from news content, source user, and user comments.

We also evaluated a reduced version of SMCC, named SMCC-R, which does not take the source user into account.

5.3.3 Evaluation Metrics

To quantifiably evaluate the effectiveness of fake news detection approaches, we adopt several widely-used standard metrics for classification problems, i.e., *Accuracy*, *Precision*, *Recall*, and F_1 *Score*. Regarding fake news detection as a binary classification problem, those metrics can be calculated based on the following notations and formulas:

- True Positive (*TP*): a fake news object is predicted as fake;
- True Negative (*TN*): a true news object is predicted as true;
- False Negative (*FN*): a fake news object is predicted as true;

- False Positive (*FP*): a true news object is predicted as fake.

$$Precision = \frac{|TP|}{|TP| + |FP|}$$

$$Recall = \frac{|TP|}{|TP| + |FN|}$$

$$F_1 = \frac{2 * |TP|}{2 * |TP| + |FP| + |FN|}$$

$$Accuracy = \frac{|TP| + |TN|}{|TP| + |TN| + |FP| + |FN|}$$

5.3.4 Model Setup

We implemented the proposed SMCC model using *Keras*[3], which is a Python wrapper of TensorFlow[4]. The model is trained and tested using five-fold cross-validation. At each round of cross-validation, we randomly split the entire dataset into five folds of equal size. We keep three folds as the training set, one fold as the validation set, and the remaining one fold as the testing set. Then, the model is trained for 1000 epochs to minimize its training loss. Weights and bias are updated using stochastic gradient descent with the Adadelta update rule (Zeiler, 2012). Dropout (Srivastava et al., 2014) is applied to each hidden layer of the model to avoid overfitting. We also applied zero-padding[5] to handle news objects that have fewer retweeters than the number of spreaders required by an SMCC model. We adopt a file logger to keep track of the best model, which yields the highest classification accuracy on the validation set after each epoch during the entire training process. Finally, the best

[3] https://keras.io/

[4] https://www.tensorflow.org/

[5] http://www.bitweenie.com/listings/fft-zero-padding/

model acquired after the 1000 epochs is applied to the testing set for model testing. Before formal cross-validation, we perform 20 rounds of pre-training to empirically configure the model's hyperparameters based on the model's accuracy on the validation set. The model configurations are show in Table 5.1. After configuring the model, we formally performed 5-fold cross-validation for 50 rounds and recorded the average performance metrics yielded on the testing set as the evaluation results.

Table 5.1 Model Configurations

Hyperparameter	**Choice**	**Experimental Range**
Number of embedding layers	1	$1-5$
Number of hidden layers	1	$1-5$
Embedding layer size	2^7	2^3-2^{10}
Hidden layer size	2^5	2^3-2^{10}
Dropout rate	0.1	0 - 1
Regularization factor (λ)	0.01	0-0.1
Learning rate	0.001	0.0001 - 0.02

For the early detection of fake news, it is important to know how long it takes for a detection approach to identify a piece of fake news after it starts to spread. Thus, a *detection deadline* must be involved to evaluate a model's performance on early detection. A detection deadline can be measured either by absolute time or by the number of news spreaders. We first measured detection deadline by the number of news spreaders, since it can be directly implemented in our SMCC model. In Twitter, news spreaders refer to "retweeters", i.e., users who "retweet" a news object. Weibo has a similar news forwarding mechanism as Twitter. Therefore, in the two experimental datasets, we set the number of news spreaders to be the number of observed retweeters. We varied the number of retweeters from 10 to 150 with an interval of 10 to train multiple models. For example, if the number of retweeters is set to 10, then we only use the first ten retweeters' information to train and test the model.

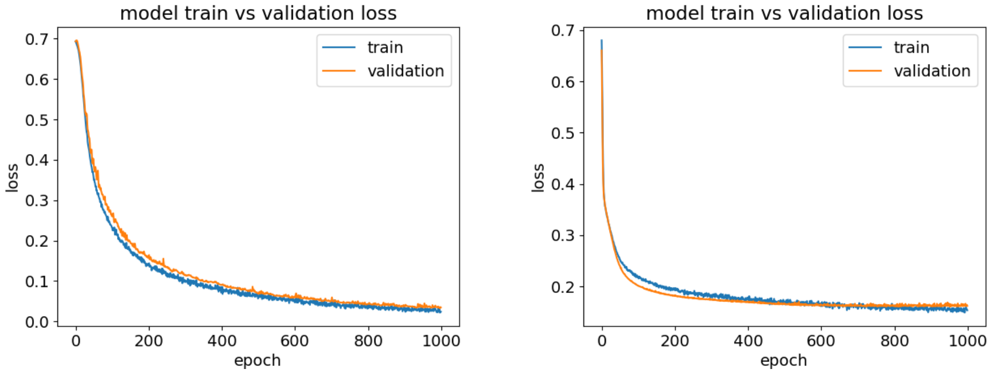
Next, we measured the detection deadline by absolute time. When a detection deadline reaches, the number of observed retweeters can be different for each news. Thus, in this scenario, we did not train multiple models again. Instead, we used the models trained when we measure the detection deadline by the number of retweeters and directly applied them to the testing set. Given a news object and a detection deadline, we applied the pre-trained model with a number of required spreaders that were less than but the closest to the observed number of retweeters to make a prediction. For example, if a news object had 25 retweeters observed by a detection deadline of 15 minutes, then we applied a model trained with 20 retweeters to make a prediction. For each detection deadline, we performed 50 rounds of 5-fold cross-validation and reported the average performance of the best model on the test set.

5.3.5 Results

Training Performance Figures 5.2-(a) and 5.2-(b) show the learning curves of the proposed model on the two experimental datasets at a random round of cross-validation, respectively. We find that the validation loss is very close to the training loss on both two datasets, which demonstrates that there exists no overfitting or underfitting in our model.

Comparison of Optimal Performance During our experimentation, we found that most detection approaches' performance peaks after observing more than 150 retweeters. Thus, we first compare their optimal performance by setting the detection deadline to be 150 retweeters. Table 5.2 shows the comparison of optimal detection effectiveness on both the *Twitter15* and the *Weibo16* datasets when the detection deadline is fixed at 150 retweeters.

From Table 5.2, we can find the following results: (i) Among the baseline models, CSI performs the best. GRU and PTK perform a little bit worse than CSI. DTC and SVM-TS perform the worst; (ii) Compared with the best baseline model CSI, the proposed SMCC



(a) Learning curve on *Twitter15*

(b) Learning curve on *Weibo16*

Figure 5.2 Learning curves on the two experimental datasets.

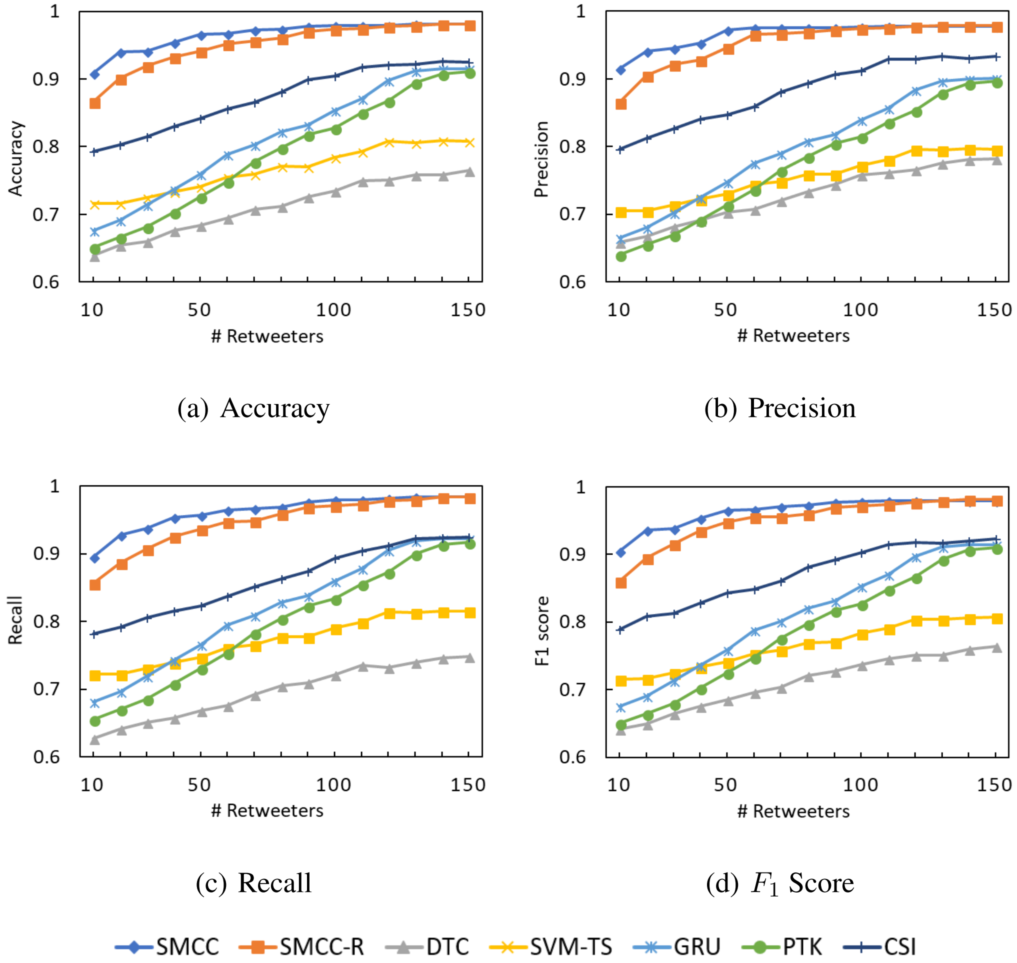
Table 5.2 Comparison of Optimal Performance

Approach	**Twitter15**				**Weibo16**			
	Accuracy	**Precision**	**Recall**	F_1 **Score**	**Accuracy**	**Precision**	**Recall**	F_1 **Score**
DTC	0.765	0.782	0.748	0.764	0.825	0.803	0.841	0.823
SVM-TS	0.808	0.796	0.815	0.807	0.867	0.842	0.877	0.867
GRU	0.915	0.901	0.923	0.915	0.921	0.906	0.945	0.921
PTK	0.911	0.896	0.917	0.910	0.914	0.909	0.938	0.915
CSI	0.925	0.934	0.910	0.923	0.934	0.906	0.947	0.932
SMCC-R	**0.980**	**0.978**	**0.983**	**0.979**	**0.934**	**0.917**	**0.959**	**0.936**
SMCC	**0.980**	**0.978**	**0.983**	**0.979**	**0.934**	**0.917**	**0.959**	**0.936**

model performs significantly better on the *Twitter15* dataset but only slightly better on the *Weibo16* dataset; (iii) The reduced model SMCC-R has the same optimal performance as SMCC, which will be discussed later; (iii) Among the four effectiveness metrics, the proposed SMCC model performs the best at recall, which we think is the most important effectiveness measurement for the task of fake news detection. In real-world social media platforms, fake news detected by our approach can be sent to social media administrators who can decide how to deal with them. Thus, it is more acceptable to receive more alerts of potential fake news that are actually true, than potentially letting through real fake news. That is the reason why recall is the most important metric. Besides recall, we also provide F-measure that also takes precision into consideration.

Comparison of Early Detection Performance Figures 5.3 and 5.4 show the comparison of early detection performance on the *Twitter15* and the *Weibo16* datasets, respectively when detection deadline is measured by the number of retweeters. Figures 5.5 and 5.6 show the comparison of early detection performance on the two datasets, respectively when detection deadline is measured by the absolute time. Figure 5.7 shows the average propagation speed of news objects on social media calculated based on our two experimental datasets.

From those figures we can find the following results: (i) For all the seven models, their performance peaks when the detection deadline is approaching 150 retweeters or 90 minutes; (ii) At the early beginning of news's propagation period, i.e., when the detection deadline is ten retweeters or five minutes, the baseline detection approaches all have a low performance while the proposed SMCC and SMCC-R have a high performance; (iii) The performance difference between the proposed models and the baseline models is larger when the detection deadline is shorter; (iv) The performance difference between SMCC and SMCC-R is also larger when the detection deadline is shorter. But this difference is much smaller than the performance difference between the proposed models and the baseline



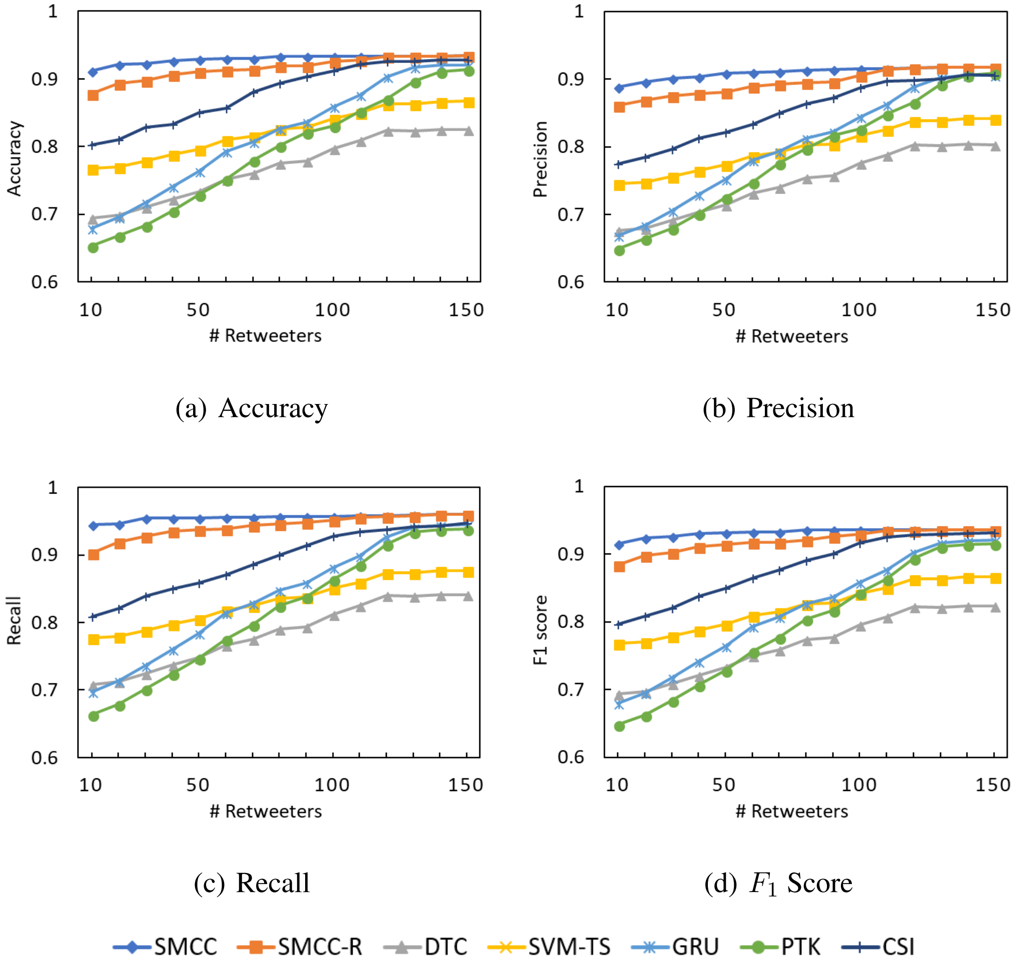
(a) Accuracy

(b) Precision

(c) Recall

(d) F_1 Score

Figure 5.3 Comparison of early detection performance on *Twitter15* when detection deadline is measured by the number of retweeters.



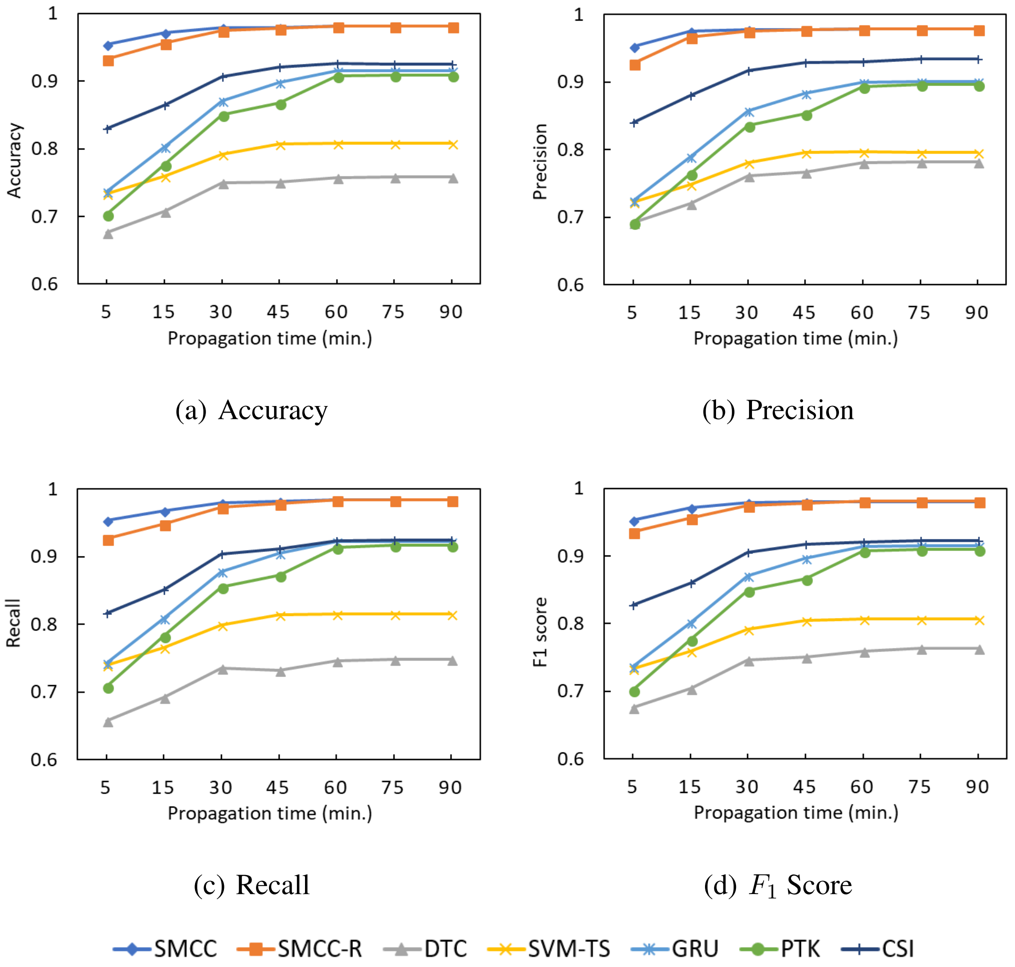
(a) Accuracy

(b) Precision

(c) Recall

(d) F_1 Score

Figure 5.4 Comparison of early detection performance on *Weibo16* when detection deadline is measured by the number of retweeters.



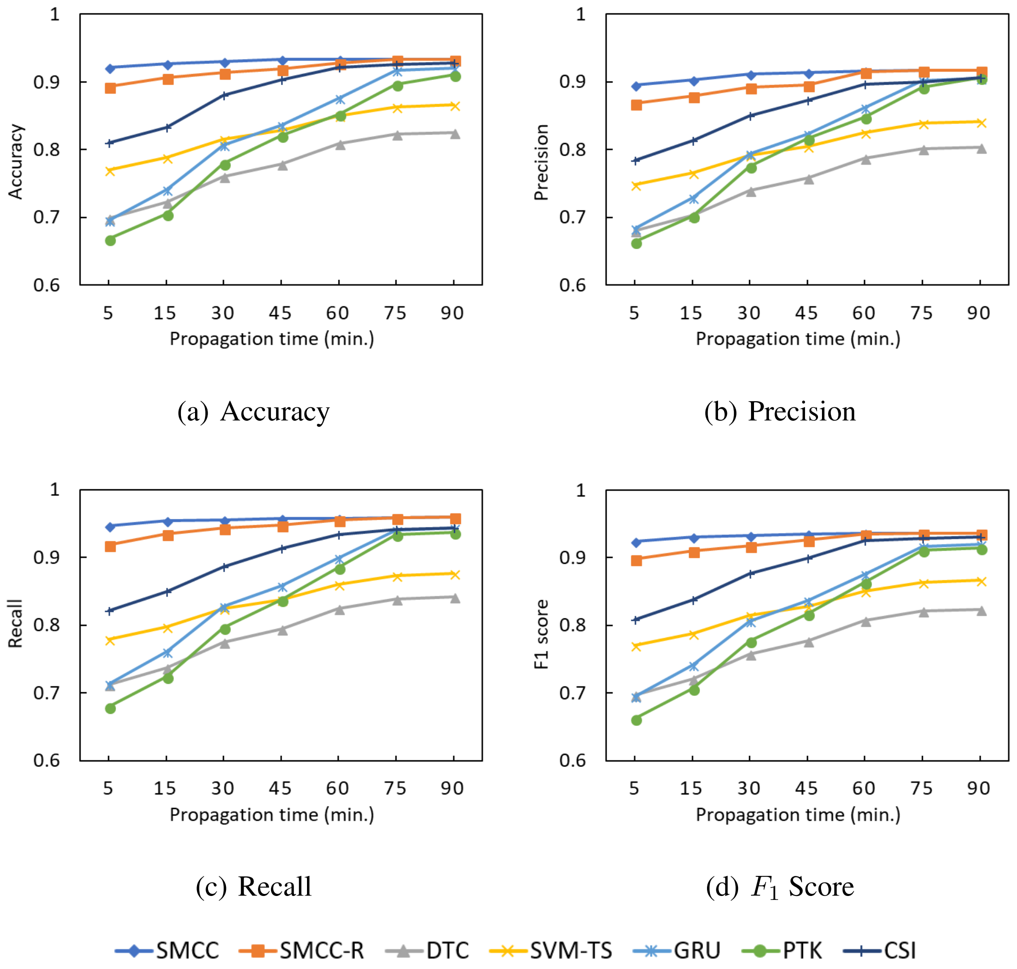
(a) Accuracy

(b) Precision

(c) Recall

(d) F_1 Score

Figure 5.5 Comparison of early detection performance on *Twitter15* when detection deadline is measured by the absolute time.



(a) Accuracy

(b) Precision

(c) Recall

(d) F_1 Score

Figure 5.6 Comparison of early detection performance on *Weibo16* when detection deadline is measured by the absolute time.

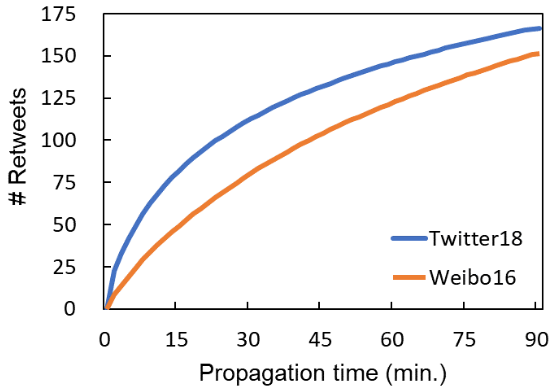


Figure 5.7 Average propagation speed of news objects on social media.

models. These results show that the output of the SMCC model is primarily determined by retweeters, and source users can increase its performance slightly when only very few retweeters are observed; (v) The early detection performances are consistent when the detection deadline is measured by both the number of retweeters or by the absolute time based on the average propagation speed data; (vi) These results demonstrate that SMCC outperforms baseline models significantly on fake news early detection.

Analysis of the Relationship between Fake News Spreader Likelihood Score and Fake News Tweeting Behavior Note that in Section 5.2.3, we produce a fake news spreader likelihood score for each source user. The proposed fake news spreader likelihood score is not applied on retweeters because, in Chapter 3, our user classification model did not perform very well on retweeters compared with on source users. To further justify its usefulness, we conducted an analysis of the relationship, fake news spreader likelihood scores, and source users' fake news tweeting behavior.

Table 5.3 shows the result of the analysis. We can find that the average fake news spreader likelihood score of source users who have tweeted fake news is significantly larger than that of source users who never tweeted fake news. Moreover, users who have tweeted fake news more than once have a larger average fake news spreader likelihood score than users who have tweeted fake news only once. These results show that there exists a strong relationship between a source user's fake news spreader likelihood score and the number of fake news he/she has tweeted. That is, the higher a source user's fake news spreader likelihood score is, the more likely and more often, the source user will spread fake news. Recall that in Section 5.2.3, we assume that the fake news spreader likelihood score is the prior probability that a news object tweeted by a particular source user is fake. The results of our correlation analysis confirm our assumption. Compared with the user model we proposed to predict whether a user is a fake news spreader in Chapter 3, we can find that the average fake news likelihood scores of source users who have posted only one fake

Table 5.3 Analysis of the Relationship Between Fake News Spreader Likelihood Score and Fake News Tweeting Behavior

Twitter15		
# Fake news	327	
# Source user	277	
# Fake news tweeted	# Source user	Avg. fake news spreader likelihood score
0	45	0.109
1	189	0.786
> 1	43	0.983
Weibo16		
# Fake news	2313	
# Source user	2309	
# Fake news tweeted	# Source user	Avg. fake news spreader likelihood score
0	470	0.102
1	1583	0.751
> 1	256	0.965

news (0.78 in *Twitter15* and 0.75 in *Weibo16*) are lower than the accuracy of the user model (0.91 in *Twitter15* and 0.87 in Weibo16), however, the average fake news likelihood scores of source users who have posted more than one fake news (0.98 in *Twitter15* and 0.96 in *Weibo16*) are higher than the accuracy of the user model. This result shows that source users who post fake news frequently are more informative for detecting fake news.

Discussion on the Effect of Missing User Profiles on the Experimental Results Recall that we slightly modified the original Twitter dataset by discarding users whose user profile is no longer available from original propagation paths. In this section, we discuss whether this operation will introduce any bias in our experimental results. Table 5.4 shows the statistics of news spreaders whose user profile is unavailable in the original *Twitter15* dataset, from which we constructed our *Twitter15* dataset. We can find that there is only a small percentage of spreaders ($< 4.5\%$) whose user profile is unavailable, both at the aggerated level and per news level. Furthermore, this percentage does not significantly differ between fake news and true news, and between the aggerated level and per news level. Thus, by discarding those users from the original propagation paths, we neither significantly reduced the size of the original dataset nor changed the relative percentages of fake news spreaders vs. non-fake news spreaders in our modified dataset. Thus, for the baselines, irrespective of whether the original or the new dataset was used, their performances should not result in significant differences. For our framework, using the modified dataset was a necessity since the original dataset did not contain user profiles. However, we do not believe our framework gained any advantage by using the modified dataset. Finally, we directly implemented all the baseline approaches using our modified dataset to make a fair comparison. Therefore, it is not likely to introduce any biased comparison results by using the modified dataset.

Table 5.4 Statistics of News Spreaders whose Profile is Unavailable in the Original Twitter Dataset

	Fake News	**True News**
Avg. number of spreaders per news	282.54	511.69
Avg. number of spreaders whose profile is unavailable per news	11.66	22.77
Avg. percentage of spreaders whose profile is unavailable per news	4.05%	4.37%
Total number of spreaders	92,391	180,627
Total number of spreaders whose profile is unavailable	3,813	7,861
Percentage of spreaders whose profile is unavailable	4.13%	4.35%

5.3.6 Discussion

In this section, we further interpret the experimental results from the perspective of methodology and discuss the advantages of SMCC over baseline models in terms of scalability, flexibility, and security. For a fake news detection approach, **scalability** measures how well it can handle massive and growing amounts of social media data; **flexibility** measures how easily it can be implemented on different social media platforms without major changes; **security** measures how strongly it can defend malicious attacks.

Among all the baseline models, DTC yielded the lowest performance because of the following reasons: (1) It adopts 15 hand-crafted features, most of which are in an aggregated level, e.g., average registration age and average followers count among retweeters. However, directly using aggregated features will lose information about individual retweeters; (2) Those 15 hand-crafted features cannot cover a wide range of user characteristics; (3) The aggregated features only become stable, given a relatively large number of retweeters. However, they often fluctuate significantly when very few retweeters are observed. For instance, the average registration age of the first ten and first 20 retweeters of the same news object may be significantly different. Thus, the training data may contain large noise for an early detection model. (4) DTC uses decision tree as its machine learning algorithm, of which the learning ability often cannot catch up with that of deep learning models, which are the most widely-used machine learning models nowadays. The basic methodology of SVM-TS is similar to that of DTC. It outperforms DTC because of the following reasons: (1) It extracts a time series of aggregated user features instead of static ones, which capture the dynamically aggregated user features within a time period; (2) SVM model has a better discriminative ability than decision tree, and it is less sensitive to dimensionality. SVM-TS still has the same four limitations of DTC. Especially, when the observed number of retweeters is small, time series of aggregated user features fluctuate more seriously. However, our SMCC model does not suffer from this problem since it aggregates a series of user features by mean-pooling.

Compared with DTC and SVM-TS, our SMCC model has the following advantages: (1) SMCC uses a comprehensive set of user features extracted from social media user profiles as input features, and only discards very few features that are extremely skewed; (2) Before aggregating user features, SMCC first transforms them into embedding vectors to reduce the amount of information loss; (3) SMCC produces a fake news spreader likelihood score of source users, which influences the model's output more significantly when the number of retweeters is very small. (4) SMCC model is less sensitive to the exact retweet sequence. The above two mechanisms reduce the impact of random noises when only a very small number of observed retweeters during an early stage of news's propagation period is observed, thus allowing the model to produce reliable outputs. (5) SMCC adopts a deep learning-based machine learning algorithm, which has a stronger discriminative ability than traditional algorithms.

GRU and PTK rely on latent and complex features extracted from the social engagements surrounding news objects to detect fake news. GRU extracts temporal patterns from sequences of user comments. PTK extracts structural patterns from retweet networks. Although these two approaches yield a good performance when the detection deadline is long, their performances drop the fastest when the detection deadline is short or very few tweets. The reason is that when a news object starts to spread, and only very few retweets are observed, there are often very few user comments, and the retweet network is quite simple. Therefore, temporal patterns and structural patterns have not formed yet. Our SMCC model does not rely on those complex features that are often inadequate for the early detection of fake news. In addition, GRU is prone to malicious attack, since it is easy for fake news producers to create several fake user comments shortly after they post fake news. Our SMCC model has a significant advantage in **security** compared with GRU. It relies on user characteristics that are more difficult to be manipulated than user comments. Moreover, when the number of retweeters grows large, the integrated embedding vector will be much more difficult to be manipulated.

CSI is a hybrid deep learning model that integrates linguistic patterns extracted from retweet text content and user features extracted from social networks. It yields the best effectiveness and efficiency among the baseline models. However, compared with our SMCC model, it has the following limitations: (1) The insufficient retweet text content degrades its performance for early detection; (2) The user features used in CSI are constructed based on SVD decomposition of social networks instead of being extracted from user profiles. This feature extraction mechanism does not fully utilize user characteristics already available from user profiles and is not scalable since it is very difficult to acquire and maintain a large social network and to perform SVD decomposition over very large networks. Compared with CSI, SMCC has major advantages on **scalability** and **flexibility**. It only requires low-level user characteristics that can be directly extracted from user profiles for model training. Moreover, its model structure is not so complex compared with very deep models like CSI. Thus, the SMCC model can be trained on a large volume of real-world data much more efficiently. In addition, SMCC can be easily implemented across multiple social media platforms without any major change. It only needs to extract different user characteristics that are supported by the underlying social media platforms.

5.3.7 Summary

In this chapter, we propose a novel machine learning model named Social Media Content Classification (SMCC) to classify social media content based on spreaders' characteristics, for the purpose of fake news early detection. Compared with the PPC model proposed in Chapter 3, SMCC is insensitive to retweet sequence; thus, it yields more robust performance when the retweet sequence is short at an early stage of news' propagation. Compared with PPC, SMCC is more suitable for Twitter-like platforms where a full propagation path is unavailable to observe.

CHAPTER 6

EARLY DETECTION OF FAKE NEWS ON SOCIAL MEDIA VIA STATUS-SENSITIVE CROWD RESPONSES

User characteristics combined with other social context data such as retweet text might encapsulate more useful patterns that can be used to differentiate fake news from true news compared with user characteristics or retweet text alone. Also, some specific retweet text posted by some specific users at some specific ranking positions of a retweet sequence might be more discriminative at identifying fake news. Thus it needs to be highlighted in some way. Thus, in this chapter, we propose our third proposed detection model named *FNED* that further improves our first two models by addressing the above issues.

6.1 Methodology

In this section, we first introduce some preliminaries used in the social media environment. Then, we formally define the problem of fake news early detection. Next, we present our proposed fake news detection model in detail.

6.1.1 Preliminaries and Problem Statement

In this section, we first introduce some preliminaries and then formally define the problem of fake news early detection. We adopt some terminologies on Twitter, such as “tweet” and “retweet”, to discuss the context of our problem. We use italic lowercase characters (a) for scalar variables, italic uppercase characters (A) for sets and functions, bold lowercase characters ($\mathbf{x}$) for vectors, and bold uppercase characters ($\mathbf{X}$) for matrices.

Let $A = \{a_1, a_2, \ldots, a_{|A|}\}$ be a set of news articles, each of which is associated with a label $y_i \in \{0, 1\}$, where $y_i = 0$ when a_i is true news, and $y_i = 1$ when it is fake news. When a news article a_i is posted on social media, usually it will be responded by a number of social media users. We define the *crowd response* of a news article a_i as a sequence of individual user responses, denoted as $R(a_i) = \big((u_0, r_0, t_0), (u_1, r_1, t_1), \ldots, (u_n, r_n, t_n)\big)$. Each tuple $(u_k, r_k, t_k) \in R(a_i)$ represents the k-th crowd response. That is, user u_k responds to the news with the response text r_k at time t_k. Without losing generalizability, let (u_0, r_0, t_0) be the first crowd response to a news article or a news event. In this case, r_0 might be the news content if u_0 originally composed the news article or a user comment plus the news content or the link of the original news article if the news article is migrated from other websites. We also call the user u_0 as the *source user* of the news article.

Next, we define the *status* of user u_k at time t_k as $S(u_k, t_k)$. The status of a social media user includes the user's characteristics and activity history observed at a certain time point. It is usually maintained in the form of *user profile* on a social media platform. Given the definition of user status, we extend $R(a_i)$ to let it be the *status-sensitive crowd responses* of a_i, denoted as:

$$R(a_i) = \big((u_0, S(u_0, t_0), r_0, t_0), (u_1, S(u_1, t_1), r_1, t_1), \ldots, (u_n, S(u_n, t_n), r_n, t_n)\big).$$

In the early stage of news propagation, the number of crowd responses is usually limited. Thus, we formulate the task of fake news early detection as detecting fake news based on the first k crowd responses, where k is a *detection deadline*. Here we measure detection deadline by the number of crowd responses instead of absolute time because of the following two reasons: (1) The number of crowd responses can be directly incorporated into the machine learning model as a parameter. (2) A detection deadline measured by absolute time can be easily transformed to that measured by the number of crowd responses via proper padding schema. We define $R(a_i, k)$ as the first k status sensitive crowd responses of a_i, denoted as: $R(a_i, k) = \big((u_0, S(u_0, t_0), r_0, t_0), \ldots, (u_k, S(u_k, t_k), r_k, t_k)\big)$. Then, the task of fake news early detection is to find a model H that predicts a label $\hat{L}(a_i) \in \{0, 1\}$ for each news article $a_i \in A$ based on its first k status-sensitive crowd responses, which is formally defined as:

$$\hat{y}(a_i) = H\big(R(a_i, k)\big).$$

6.1.2 Model Overview

Our proposed fake news detection model has three major components: a *Status-Sensitive Crowd Response Feature Extractor* (shown in Figure 6.3), a *CNN-based News Classifier* (shown in Figure 6.4), and a *PU-Learning Framework* (shown in Figure 6.5).

Given a news article posted on social media, our detection model first collects its status-sensitive crowd responses, each of which is a combination of a piece of text response and a user profile of the user who sends the response. Next, a status-sensitive crowd response feature extractor extracts both texts, and user features from status-sensitive crowd responses, and then concatenate them to form a feature map that represents the news article. Then, a CNN-based news classifier is applied to produce a class label based on the extracted status-sensitive crowd response feature map. A PU-Learning framework is also utilized to enhance the performance of our detection model given unlabeled and imbalanced training data. We name our proposed detection model as *FNED* (Fake News Early Detection). Figure 6.1 shows the flowchart of our proposed detection model.

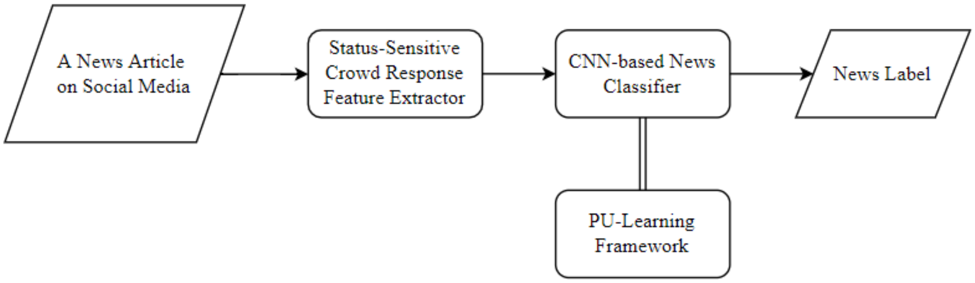

A News Article on Social Media
Status-Sensitive Crowd Response Feature Extractor
CNN-based News Classifier
News Label
PU-Learning Framework

Figure 6.1 Flowchart of our proposed fake news early detection model.

6.1.3 Status-Sensitive Crowd Response Feature Extractor

Figure 6.2 visualizes the status-sensitive crowd responses to a given news article. Given a news article posted on social media, a sequence of its crowd responses, e.g., retweets or comments, are observed. In some cases, the first crowd response consists of a news title followed by a URL. Each crowd response is associated with a user profile of the user who sends this response. The combination of a crowd response with its corresponding user profile forms a *Status-Sensitive Crowd Response*.

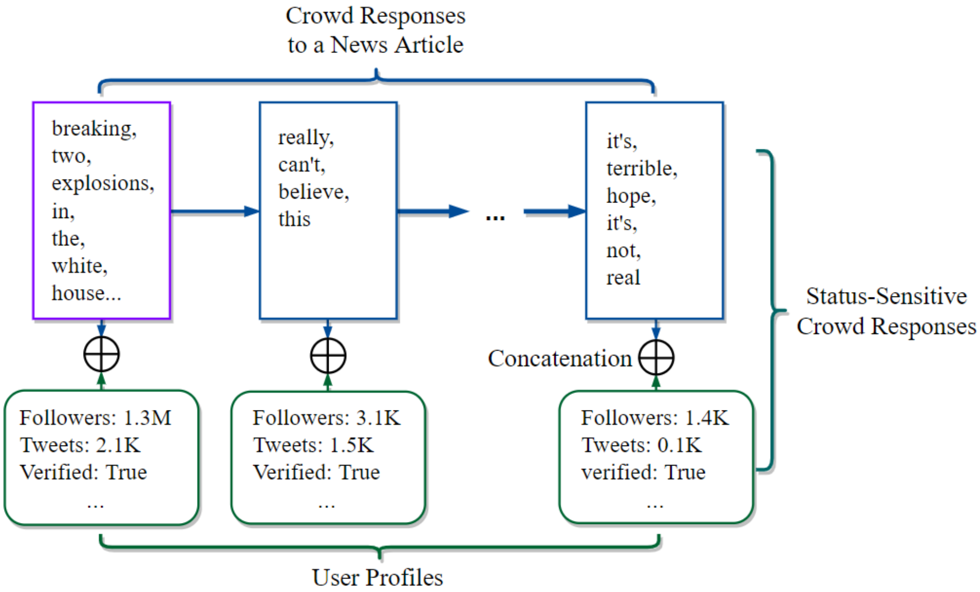


Figure 6.2 Status-Sensitive Crowd Responses to a given news article.

For each status sensitive crowd response $(u_j, S(u_j, t_j), r_j, t_j) \in R(a_i, k)$, a text feature vector $\mathbf{c}_j \in \mathbb{R}^{d_1}$ is extracted from the response text r_j via a basic Text-CNN block (Y. Wang et al., 2018), and a user feature vector $\mathbf{u}_j \in \mathbb{R}^{d_2}$ is extracted from the user status $S(u_j, t_j)$ via an embedding block. The user status $S(u_j, t_j)$ is recorded in the user profile. Next, $\mathbf{c}_j$ and $\mathbf{u}_j$ are concatenated to form a status-sensitive crowd response feature vector:

$$\mathbf{r}_j = \mathbf{c}_j \oplus \mathbf{u}_j,$$

where $\mathbf{r}_j \in \mathbb{R}^d$, $d = d_1 + d_2$ and $\oplus$ is the concatenation operator. Here d_1,d_2, and d are the dimensions of the text, user, and the concatenated status-sensitive crowd response feature vector, respectively. Then, the first k status-sensitive crowd response feature vectors are concatenated to form a feature map that represents the news article a_i:

$$\mathbf{R}_{i,k} = \mathbf{r}_1 \oplus \mathbf{r}_2 \oplus \cdots \oplus \mathbf{r}_k,$$

where $\mathbf{R}_{i,k} \in \mathbb{R}^{d \times k}$. The architecture of the proposed Status-Sensitive Crowd Response Feature Extractor is shown in Figure 6.3.

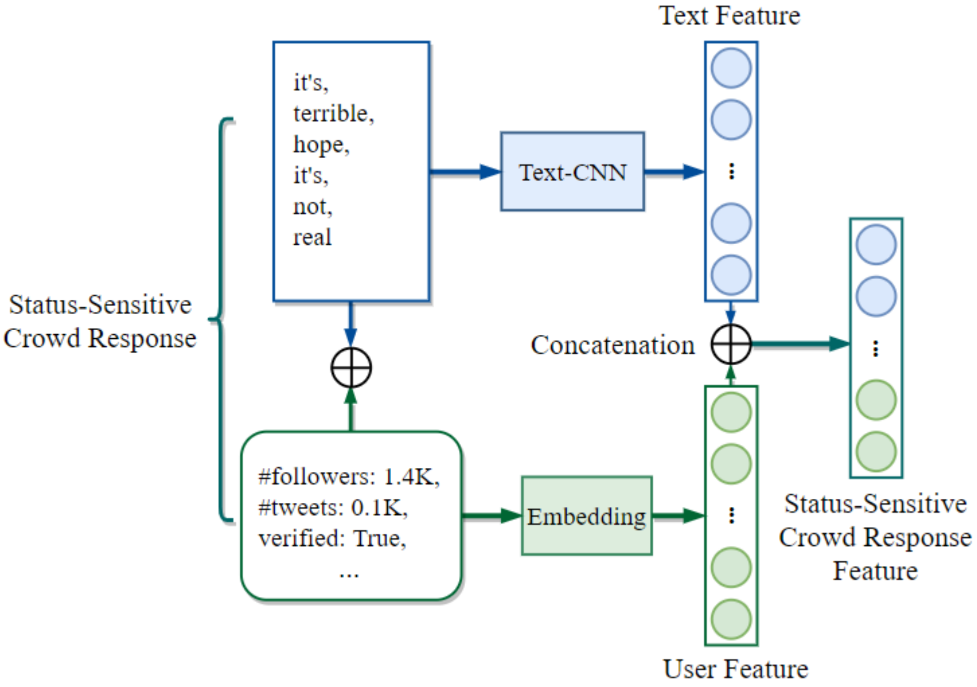


Figure 6.3 Architecture of the Status-Sensitive Crowd Responses Feature Extractor.

6.1.4 CNN-based News Classifier

The output of the Status-Sensitive Crowd Responses Feature Extractor is a feature map that consists of a sequence of k concatenation of text and user features. Our proposed CNN-based News Classifier utilizes basic convolution networks (CNNs) and two novel mechanisms proposed by ourselves, i.e., *Position-Aware Attention Mechanism* and *Multi-Region Mean-Pooling*, to produce a news label from this feature map. Figure 6.4 shows the architecture of CNN-based news classifier.

Position-aware Attention Mechanism Given a sequence of status-sensitive crowd responses, it is intuitive to assume that not all of them have the same ability to discriminate true and fake news. Some special text response generated by some special type of user in some special ranking position may reflect the truthfulness of a concerned news article

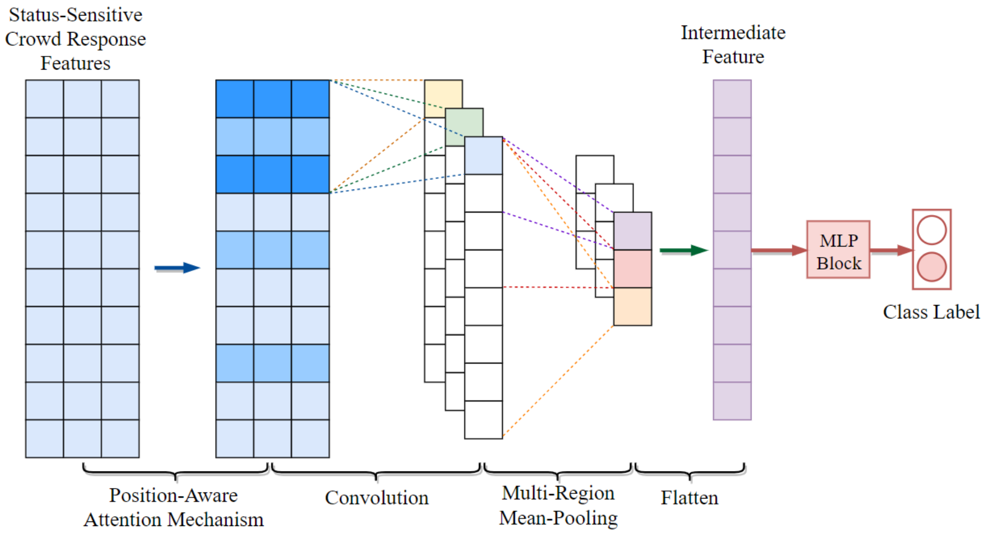

Status-Sensitive Crowd Response Features
Intermediate Feature
MLP Block
Class Label
Position-Aware Attention Mechanism
Convolution
Multi-Region Mean-Pooling
Flatten

Figure 6.4 Architecture of the CNN-based News Classifier.

more significantly, thus should be somehow highlighted in the entire propagation path. Thus, our detection model should be able to learn how much attention should be put over each status-sensitive crowd response. We propose a *Position-aware Attention Mechanism*, which is an extension of the basic Attention Mechanism (Mnih, Heess, Graves, et al., 2014; Bahdanau, Cho, & Bengio, 2014), to solve this problem.

For each status-sensitive crowd response feature vector $\mathbf{r}_j (1 \leq j \leq k)$, its attention weight and transformed vector is calculated as follows:

$$\mathbf{r}'_j = \mathbf{r}_j \oplus (j/k),$$

$$F_w(\mathbf{r}'_j) = \mathrm{Relu}(\mathbf{W}^T_{aj}\mathbf{r}'_j + \mathbf{b}_{aj}),$$

$$\alpha_j = \frac{\exp(F_w(\mathbf{r}'_j))}{\Sigma_k \exp(F_w(\mathbf{r}'_j))},$$

$$\mathbf{r}''_j = \alpha_j \mathbf{r}_j,$$

where (j/k) is the relative ranking position of the j-th status-sensitive crowd response, $\mathbf{r}'_j$ is the concatenation of the j-th status-sensitive crowd response feature vector with its relative ranking position, F_w is an attention score function with weights $\mathbf{W}_a$ and bias $\mathbf{b}_a$, α_j is the normalized attention weight of the j-th status-sensitive crowd response via a softmax function, $\mathbf{r}''_j$ is the transformed status-sensitive crowd response feature vector after our Position-aware Attention Mechanism. The difference between our proposed position-aware attention mechanism and the basic attention mechanism is that in the position-aware attention mechanism, the ranking position of each data point is considered. The difference between our proposed position-aware attention mechanism and the basic attention mechanism is that in the position-aware attention mechanism, the ranking position of each data point in a sequence of data points is considered, whereas the basic attention mechanism does not take this information into consideration. Therefore, our proposed position-aware attention mechanism can be used to classify sequential data where the ranking positions of data points are important.

Convolution Network Given the dimension of the transformed feature map $\mathbf{R}''_{i,k}$ as $d \times k$, a convolution network with kernel size $d \times h$ and number of filters l is applied to extract intermediate features. In detail, each convolutional filter with window size $d \times h$ takes the contigious h status-sensitive crowd response feature vectors as the input and outputs one scalar feature:

$$s_j = \mathrm{Relu}(\mathbf{W}_c \cdot \mathbf{R}''_{i,j:j+h-1} + \mathbf{b}_c),$$

where $\mathbf{W}_c, \mathbf{b}_c$ are the weights and bias of the convolutional filter. We perform the same convolution operation with l filters to produce a feature vector $\mathbf{s}_j \in \mathbb{R}''^{l}$. By repeating the same convolution operations for each window of h consecutive status-sensitive crowd response feature vectors, we obtain a sequence of intermediate feature vectors:

$$\mathbf{s} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{k-h+1}].$$

Multi-Region Mean Pooling Next, we propose a novel mean pooling mechanism named *Multi-Region Mean Pooling* to extract aggregated features from the feature map. Instead of one-time mean pooling over all the $k-h+1$ feature vectors, m mean pooling operations are performed, each over the first $\frac{k-h+1}{2^{m-1}}$ feature vectors:

$$\bar{\mathbf{s}}_m = \Sigma_{j=1}^{\frac{k-h+1}{2^{m-1}}} \mathbf{s}_j / \frac{k-h+1}{2^{m-1}}.$$

We propose this unique mean-pooling mechanism because of the following reasons: (1) Multi-Region Mean-Pooling can capture different granularities of aggregated features from the entire feature map, while the basic mean-pooling can only calculate one overall average; (2) If the real available number of crowd responses is less than k, zero padding is required. If the feature map $R''_{i,k}$ contains too many zero vectors, then after convolution operations, the intermediate feature vectors will contain too many zero vectors (if $b_c = 0$) or bias vectors (b_c). Thus, the basic mean-pooling approach will cause information loss from the non-zero intermediate feature vectors because they will be averaged together with lots of zero vectors or bias vectors. However, our proposed Multi-Region Mean Pooling approach does not suffer from this problem because, in several small regions, only the non-zero intermediate feature vectors will be averaged. After mean pooling, m intermediate feature vectors are flattened and then concatenated into one single intermediate feature vector:

$$\mathbf{f}_{i,k} = \mathbf{s}_1 \oplus \mathbf{s}_2 \oplus \cdots \oplus \mathbf{s}_m.$$

News Classification Finally, a multi-layer perceptron (MLP) block that consists of multiple fully-connected layers is adopted to produce a class label for the news article a_i, simply denoted as:

$$\hat{y}(a_i) = \mathrm{softmax}(\mathrm{Relu}(\mathbf{W}_m \cdot \mathbf{f}_{i,k} + \mathbf{b}_m)),$$

where $\mathbf{W}_m$, $\mathbf{b}_m$ are the weights and bias of the MLP block.

6.1.5 Optimization

We denote our CNN-based news classifier as $H(\cdot;\theta)$, where θ denotes all the included parameters. Let Y be the set of news labels. We adopt the cross entropy function to measure the detection loss:

$$L(\theta, k) = -\mathbb{E}_{(a_i,y_i)\sim(A,Y)}[y_i \log H(R(a_i, k)) + (1 - y_i) \log(1 - H(R(a_i, k)))].$$

Given the detection deadline k, the optimization goal is to find the optimal θ that minimize the detection loss:

$$\hat{\theta} = \arg\min_{\theta} L(\theta, k).$$

The optimization can be solved by stochastic gradient descent-based optimization approaches.

6.1.6 The PU-Learning Framework

Figure 6.5 shows the architecture of our proposed PU-Learning framework. It is adopted when our proposed CNN-based news classifier is trained only with positive (fake news in our context) and unlabeled news samples, in order to best mimic the real-world scenario. In the PU-Learning framework, the training data includes a collection of positive (fake) news samples (P), and a collection of unlabeled news samples (U) whose truthfulness are supposed to be unknown. The size of positive news samples is supposed to be smaller than the size of unlabeled news samples, i.e., $|P| < |U|$. And among the unlabeled news samples, the size of positive unlabeled (real fake) news samples (PU) is supposed to be smaller than the size of negative unlabeled (true) news samples (NU), i.e., $|PU| < |NU|$, and $|PU| + |NU| = |U|$. To create a balanced dataset for training a binary news classifier, we first conduct *undersampling* over the unlabeled news samples. A collection of *pseudo-true news* samples (N') is randomly selected from unlabeled news samples (U) whose size is the same as the size of positive news samples, i.e., $|N'| = |P|$. Then, we train an instance of our proposed news classification model on the combination of the pseudo-true news samples and the positive news samples ($N' \cup P$). During the model training process, we regard pseudo-true news samples as true news samples. The result of the model training process is a weak classifier. We repeat this undersampling and model training process for k times. Then, k weak classifiers are produced. Next, we ensemble those k weak classifiers by simply averaging their outputs to generate a strong classifier. Then, we use this strong classifier to classify the unlabeled news samples (U). The top n unlabeled news samples that are classified as fake consist of a collection of *machine labeled fake news* samples (P'). Next, we append the machine labeled fake news samples to the real fake news samples to update the collection of real fake news samples, i.e., $P \Leftarrow P + P'$. The procedure of undersampling, weak classifier training, ensemble classification, and positive sample updating is repeated over a number of times until the accuracy of the strong classifier on a validation dataset peaks. The parameters of our proposed PU-Learning framework are the number of weak classifiers per iteration (k), and the number of machine-labeled fake news samples produced per iteration (n).

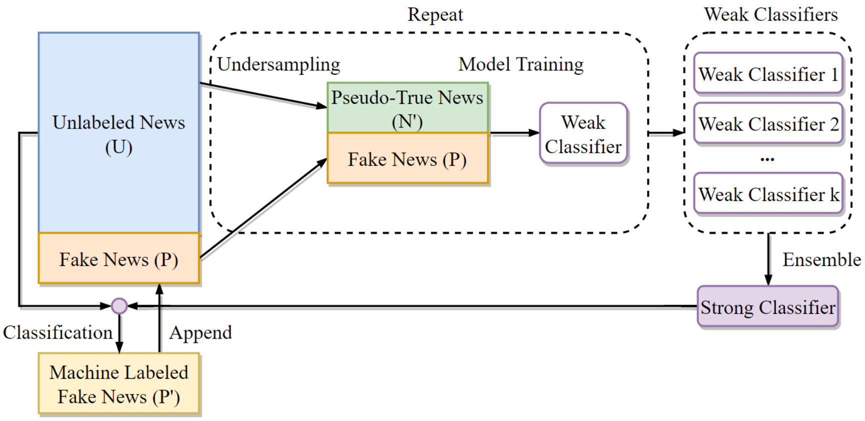


Figure 6.5 Architecture of our proposed PU-Learning framework.

6.2 Experiments & Results

6.2.1 Dataset

We used the datasets described in Chapter 3 to evaluate our proposed model's and the baselines' detection effectiveness.

6.2.2 Baseline Approaches

We compared our proposed fake news early detection model (FNED) with a series of baseline models, including:

- DTC ((Castillo et al., 2011)) A decision tree model that detects fake news based on aggregated news characteristics.

- SVM-TS ((Ma et al., 2015)) An SVM model that detects fake news based on time-series of aggregated news characteristics.

- GRU ((Ma et al., 2016)) An RNN model that detects fake news based on temporal-linguistic patterns recognized from sequences of user comments.

- CSI ((Ruchansky et al., 2017)) A hybrid deep learning model that detects fake news based on features extracted from news content, source user, and user comments.

- BLSTM ((Guo et al., 2018)) A hierarchical social attention network for rumor detection.

- PPC ((Y. Liu & Wu, 2018)) An RNN+CNN model to detect fake news early based on news propagation path represented by a sequence of user features.

- RvNN ((Ma et al., 2018)) A deep network model based on top-down tree-structured neural networks for rumor representation learning and classification. We didn't implement the bottom-up version since its performance is lower.

6.2.3 Experimental Setup

We implemented the proposed model using *Keras*[1], which is a Python wrapper of TensorFlow[2]. When preprocessing the text responses, English characters in the Twitter15 dataset is tokenized using the NLTK toolkit[3], Chinese characters in the Weibo16 dataset is tokenized using an open-source Chinese tokenizer[4]. The model was trained and tested using 5-fold cross-validation. At each round of cross-validation, we randomly split the entire dataset into five equal-sized folds. We kept three folds as the training set, one fold as the validation set, and the remaining one fold as the testing set. Then, the model was trained for 1000 epochs to minimize its training loss. Weights and bias were updated using stochastic gradient descent with the Adadelta update rule (Zeiler, 2012). Dropout (Srivastava et al., 2014) was applied to each hidden layer of the model to avoid overfitting. Before conducting formal cross-validation, we performed 20 rounds of pre-training to configure the model's hyper-parameters based on the model's accuracy on the validation set. Table 6.1 presents a list of hyper-parameters of our proposed FNED model as well as their experimental ranges. After configuring the model, we formally performed 5-fold cross-validation for 50 rounds and reported the average performance metrics yielded on the testing set as the evaluation results. We adopt standard effectiveness metrics, including accuracy, precision, recall, and F_1 score, to evaluate all the models. And we measured the detection deadline both by the number of retweets, i.e., the first k-th crowd responses, and by propagation time. When propagation time was used as the detection deadline, we calculated the average number of crowd responses observed before the detection deadline as the model parameter k. Zero-padding[5] was applied to handle news articles that have fewer than k crowd responses. In the PU-Learning setting, we trained 50 weak classifiers per iteration and appended the top 5% of the unlabeled news samples that are classified as

[1] https://keras.io/

[2] https://www.tensorflow.org/

[3] https://www.nltk.org/

[4] https://www.npmjs.com/package/chinese-tokenizer

[5] http://www.bitweenie.com/listings/fft-zero-padding/

fake news by the strong classifier with the highest confidence score to the real fake news collection at each iteration. We trained and evaluated our proposed model under multiple combinations of the class balance ratio, i.e., $|P|/|P+N|$, and positive label ratio, i.e., $|PL|/|P|$, to mimic its performance in real-world scenarios.

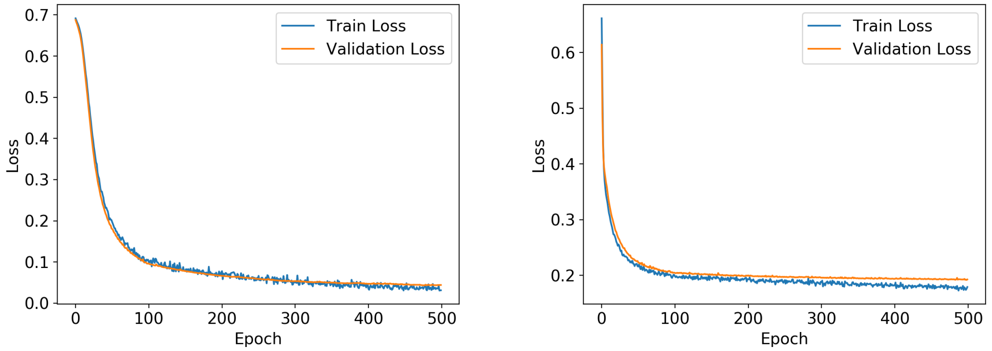
Table 6.1 Hyper-Parameters of the Proposed FNED Model

Hyper-Parameter	Value	Experimental Range
The dimension of the text feature d_1	2^7	$2^5 - 2^{10}$
The dimension of the user feature d_2	2^7	$2^5 - 2^{10}$
Convolution window height h	5	$1 - 20$
Number of multi-region mean-pooling operations m	5	$1 - 10$
Overall dropout rate	0.15	$0 - 0.5$
The number of weak classifiers per iteration k	10	$1 - 50$
The number of machine labeled fake news samples produced per iteration n	5	$1 - 20$

6.2.4 Results

Training Performance Figure 6.6 shows the learning curves of the proposed model on the two experimental datasets at a random round of cross-validation, respectively. We find that the validation loss is very close to the training loss on both two datasets, which demonstrates that there exists no overfitting or underfitting in our model.

Comparison of Optimal Performance Through our experiments, we found that our detection model's performance peaks after observing more than 150 retweets. Thus, we first compare their optimal performance by setting the detection deadline to be the first 150 retweets, i.e., $k = 150$. Table 6.2 shows the comparison of optimal detection effectiveness. From Table 6.2 we can find that our proposed FNED model outperforms the baseline models in terms of each evaluation metric, especially in the recall of the fake news.



(a) Learning curve on Twitter15.

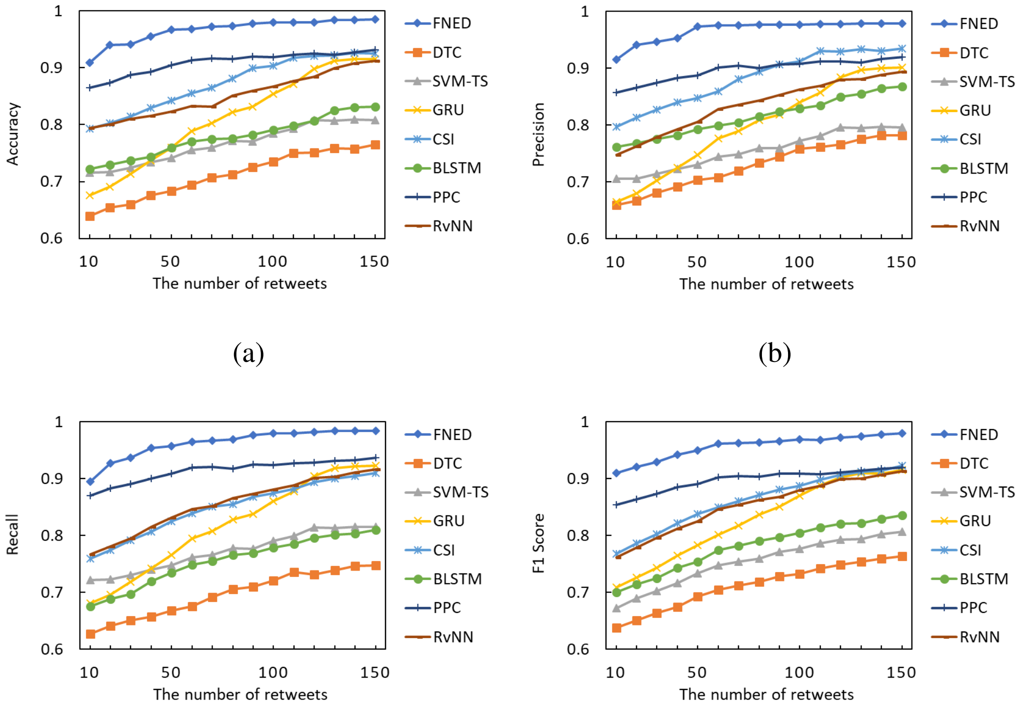
(b) Learning curve on Weibo16.

Figure 6.6 Learning curves on the two experimental datasets.

Table 6.2 Comparison of Optimal Performance when $k = 150$

Approach	Twitter15				Weibo16			
	Accuracy	Precision	Recall	F_1 Score	Accuracy	Precision	Recall	F_1 Score
DTC	0.765	0.782	0.748	0.764	0.825	0.803	0.841	0.823
SVM-TS	0.808	0.796	0.815	0.807	0.867	0.842	0.877	0.867
GRU	0.915	0.901	0.923	0.915	0.921	0.906	0.945	0.921
CSI	0.925	0.934	0.910	0.923	0.934	0.906	0.947	0.932
BLSTM	0.831	0.868	0.810	0.836	0.924	0.919	0.928	0.925
PPC	0.932	0.919	0.937	0.920	0.931	0.925	0.938	0.932
RvNN	0.912	0.894	0.916	0.913	0.919	0.910	0.932	0.915
FNED	**0.985**	**0.979**	**0.983**	**0.980**	**0.938**	**0.929**	**0.952**	**0.942**

Comparison of Early Detection Performance Figures 6.7 - 6.10 show the comparison of early detection performance on the two experimental datasets when detection deadline is measured by the number of retweets and the propagation time, respectively. Figure 6.11 shows the average propagation speed of news articles on social media calculated based on our two experimental datasets. From these figures, we can find that our proposed model outperforms the baselines significantly in terms of early detection on all metrics. And this performance difference is more significant when the detection deadline is shorter. Also, the evaluations of early detection performance using different detection deadlines are consistent based on the average propagation speed of news articles on social media.



(a)

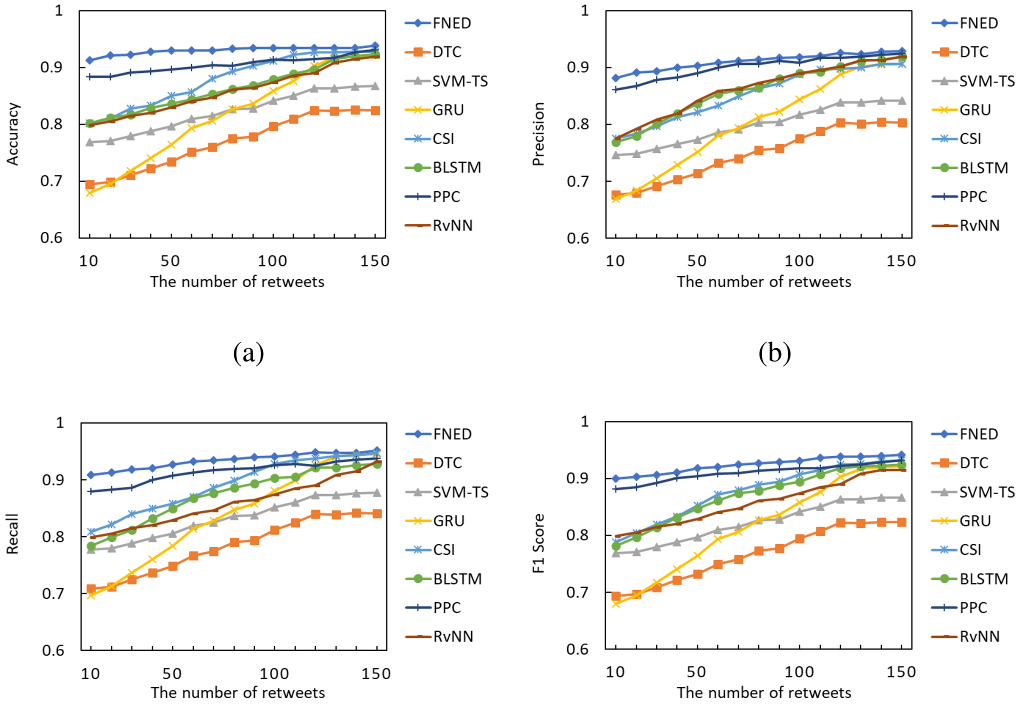
(b)

(c)

(d)

Figure 6.7 Early detection performance comparison on Twitter15 when detection deadline is measured by the number of retweets.

Ablation Study We also evaluate several simplified variations of our proposed model, each of which has one key component removed. We conduct this ablation study in order to



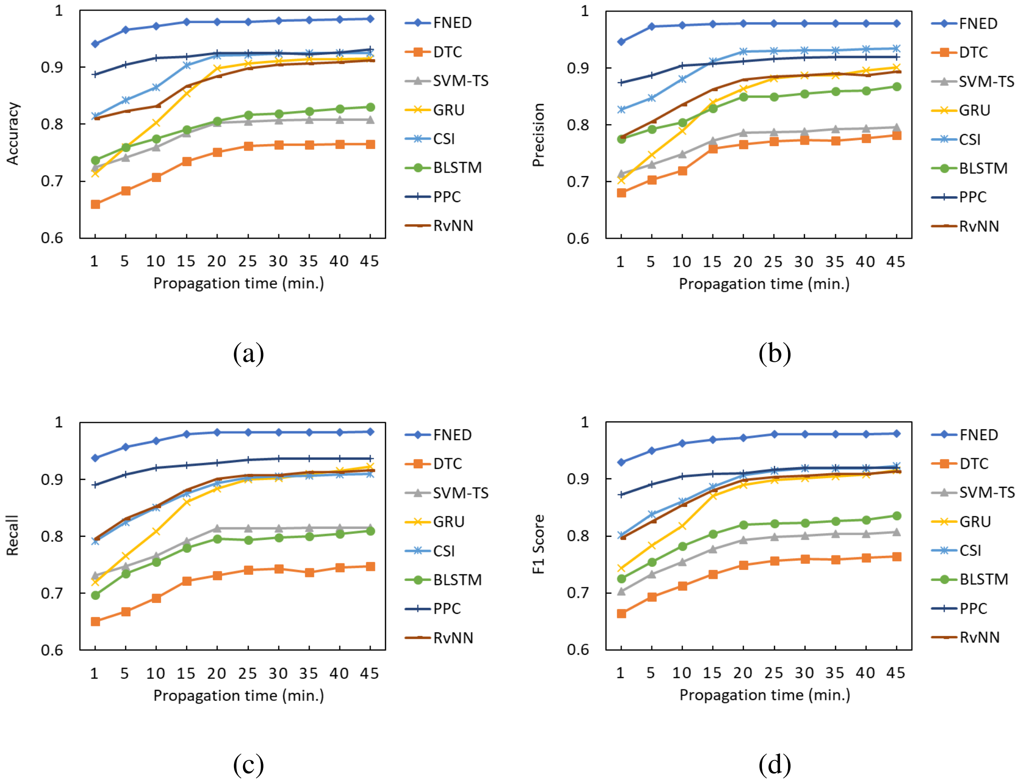
(a)

(b)

(c)

(d)

Figure 6.8 Early detection performance comparison on Weibo16 when detection deadline is measured by the number of retweets.



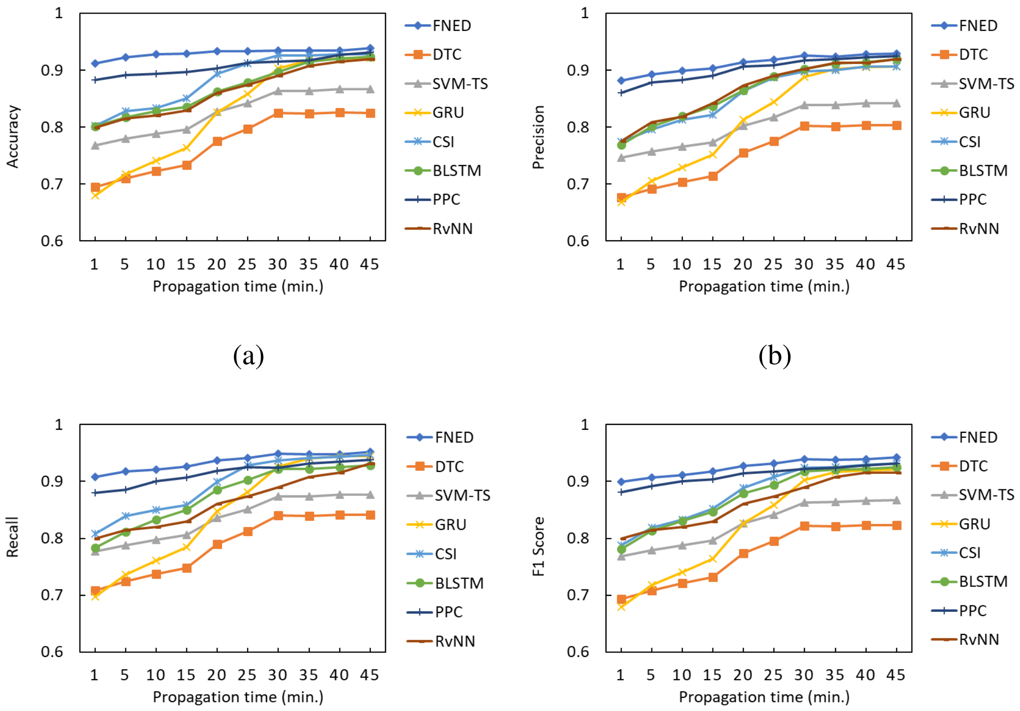
(a)

(b)

(c)

(d)

Figure 6.9 Early detection performance comparison on Twitter15 when detection deadline is measured by the propagation time.



(a)

(b)

(c)

(d)

Figure 6.10 Early detection performance comparison on Weibo16 when detection deadline is measured by the propagation time.

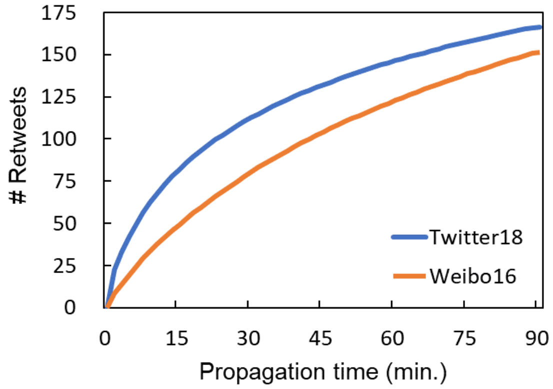


Figure 6.11 Average propagation speed of news articles on social media.

investigate the impact of each key component of our proposed model that we proposed in this study. Below is a list of reduced internal models:

- FNED-UF: User features extracted from user profiles are removed. Only text features are used to model crowd responses.
- FNED-TF: Text features extracted from the response text are removed. Only user features are used to model crowd responses.
- FNED-PAAM: Position-aware attention mechanism is removed. All crowd responses are treated identically.
- FNED-MRMP: Multi-region mean-pooling is replaced with the basic global average-pooling.
- FNED: The full model.

Table 6.3 shows the comparison of the optimal performance of the reduced internal models and the full model. From the results, we can find that if one key component is removed, our proposed model's performance will drop. Among the four key components, user features affect the detection accuracy most significantly, while text feature affects it most insignificantly. These results demonstrate that all proposed features in the FNED model contribute to its effective early detection.

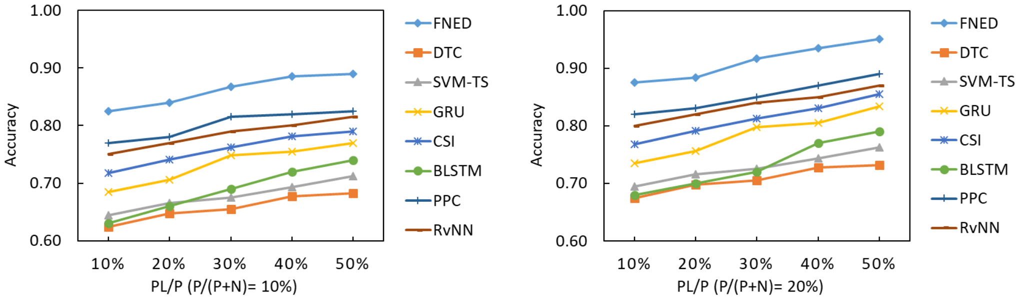
6.2.5 Performance of PU-Learning

In this section, we report our proposed model's and the baseline models' performance under the PU-Learning scenario, i.e., when training data is imbalanced and not fully labeled. Figures 6.12 and Figure 6.13 show the results. From these figures, We can find that when the class-distribution is more balanced and more positive labeled news samples are available, our models and the baselines' detection accuracy increases. Among all the models, our proposed model still performs the best. Compared with Table 6.2 which

Table 6.3 Comparison of Optimal Performance of the Reduced Internal Models and the Full Model

Approach	Twitter15				Weibo16			
	Accuracy	Precision	Recall	F_1 Score	Accuracy	Precision	Recall	F_1 Score
FNED-UF	0.905	0.892	0.913	0.901	0.889	0.862	0.913	0.905
FNED-TF	0.962	0.958	0.963	0.961	0.921	0.914	0.927	0.923
FNED-PAAM	0.952	0.943	0.976	0.953	0.915	0.907	0.931	0.918
FNED-MRMP	0.932	0.914	0.946	0.933	0.921	0.911	0.942	0.915
FNED	**0.985**	**0.979**	**0.983**	**0.980**	**0.938**	**0.929**	**0.952**	**0.942**

shows the optimal detection performances, we can find that when the class balance ratio ($P/(P+N)$) is 20% and the positive label ratio (PL/P) is 50%, our proposed model can yield a similar accuracy as the model trained using the complete dataset. However, only 10% of the labeled fake news samples in the original datasets are used for training our model. Thus, it proves our model's effectiveness under PU-Learning settings.



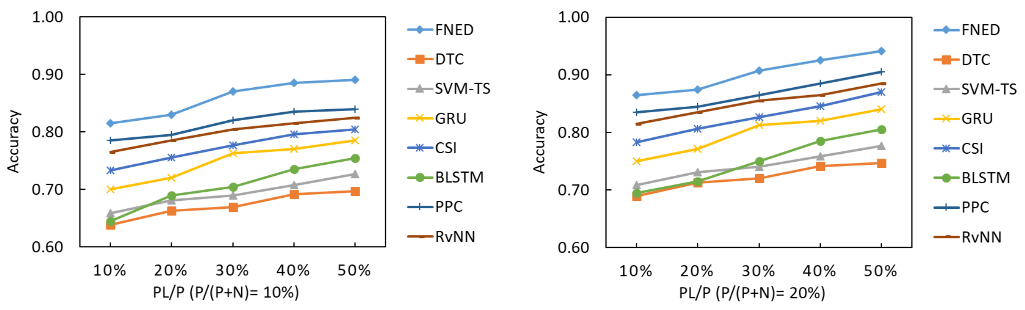
(a)

(b)

Figure 6.12 Performance of PU-Learning on *Twitter15* dataset.

6.2.6 Discussion

In this section, we further discuss our experimental results to explain why our proposed model outperforms the baselines, as well as the implications of our proposed fake news detection model and its key components.



(a) (b)

Figure 6.13 Performance of PU-Learning on *Weibo16* dataset.

As we can see from our user feature study, the user characteristics of fake news spreaders distribute significantly differently from those of the general user population on social media. We also built a simple neural network model to predict whether a social media user is likely to spread fake news based on his/her user characteristics. The model's prediction accuracy is high except on the Twitter15 dataset, which does not include a large number of retweeters. Based on the above findings, we assume that user characteristics of news spreaders can be utilized to detect fake news. Thus, we combine users' text response to a news article with their corresponding user profiles to generate status-sensitive crowd responses. Status-sensitive crowd responses can give us more information about the truthfulness of a news article than text response only. For example, the text response "I believe this is true" generated by a user who has never spread fake news and the same comment generated by another user who have spread some fake news pieces might give us an entirely different clue about whether the concerned news is fake. Our user classification model can predict whether a user is likely to spread fake news with an acceptable accuracy in most cases. We also believe its performance can be improved given larger datasets. The rich information contained in the user characteristics of news spreaders has not been fully utilized by existing approaches yet. Many existing approaches model the crowd response to a news article by textual and linguistic features, e.g., GRU (Ma et al., 2016).

Although recent approaches (T. Chen et al., 2017; Ruchansky et al., 2017; Guo et al., 2018) incorporate user features, they treat them independently with text responses instead of combining them together, as our proposed model does. Compared with the baselines, our proposed model fully utilizes the information encoded in status-sensitive crowd responses to detect fake news. That is one reason why our model outperforms the baselines.

Early detection of fake news is of critical importance. Although many existing approaches have decent performances when a large amount of data is observed, their early detection performance is low and thus will be of marginal use in the real world. The reason is that at the early stage of fake news propagation, the data required by these models is usually insufficient. For instance, the baseline model RvNN (Ma et al., 2018) detects fake news based on a recursive neural representation of the news propagation tree. However, at the early stage of news propagation, the structure of the propagation tree is usually very simple, e.g., only one root node with several child nodes. It is difficult to identify significant differences between the propagation tree structure of fake news and that of true news. Another example is the baseline mode GRU (Ma et al., 2016). It adopts recurrent neural networks to learn linguist patterns from a sequence of users' response text to a news article to identify fake news. However, users may retweet a news article without any response text to it. At the early stage of news propagation, users' response text is often insufficient. This affects this model's early detection performance. Compared with the baselines, our proposed model can fully utilize the data observed at the early stage of news propagation, which is a sequence of status-sensitive crowd responses. Therefore, our proposed model outperforms the baselines significantly in fake news early detection. The results of PU-Learning also indicate our model's robust performance when training data is imbalanced and not fully-labeled.

To effectively learn hidden patterns from a sequence of status-sensitive crowd responses that can be used to detect fake news, we propose two novel deep learning mechanisms in our CNN-based model, i.e., position-aware attention mechanism and multi-region mean-pooling. A news article usually receives a number of status-sensitive crowd responses, but not all of them have the same ability to differentiate fake news from true news. Therefore, our detection model is designed to pay more attention to those status-sensitive crowd responses that can reflect the truthfulness of the news article more significantly. Compared with the basic attention mechanism, our proposed position-aware attention mechanism takes the ranking position of each status-sensitive crowd response into consideration. Ranking position is important when modeling users' response to a news article. Some specific response generated by some specific user at some specific ranking position might give us an important clue as to whether a concerned news article is fake. However, the basic attention mechanism without the position information cannot learn this pattern. Another novel deep learning mechanism we proposed is multi-region mean-pooling. Compared with the basic mean-pooling, it can extract aggregated features from a feature map in multiple-granularity. To detect fake news early, it is necessary to model early status-sensitive crowd responses differently from the late ones. Our multi-region mean-pooling mechanism gradually calculates an average of the first several hidden representations of the status-sensitive crowd responses, i.e., first 5, 10, 15, ..., to achieve this purpose. Another advantage of multi-region mean-pooling is that it can handle missing data better. Assume that a model is trained based on sequences of 50 status-sensitive crowd responses. When it is applied to classify a sequence of 10 status-sensitive crowd responses, zero-padding is applied to extend the length of this sequence. In this case, the basic mean-pooling will average the feature vectors learned by CNN with lots of zeros. This will cause some information loss. Our proposed multi-region mean-pooling does not suffer from this problem because the first ten feature vectors learned by CNN will be averaged separately from the later 40 vectors. Our ablation study proves the effectiveness of our proposed novel position-aware attention mechanism and multi-region mean-pooling.

The advantages of our proposed FNED model compared with baseline models indicate promising potential for our model to be implemented in real-world social media platforms for fake news early detection. It can be applied on social media sites as a filter to automatically label potential fake news articles. Then, the labeled articles can be sent to social media administrators who will decide how to handle them. Beyond the task of fake news detection, our proposed position-aware attention mechanism and multi-region mean-pooling provide a solution to model sequential data in other machine learning tasks where the ranking position of each data point is important.

6.3 Summary

In this chapter, we propose a novel deep neural network to detect fake news early. Our experimental results demonstrate that status-sensitive crowd response, i.e., a user response to a news article combined with user characteristics, is more useful for fake news early detection than a user response alone. Our proposed detection model includes two novel deep learning mechanisms that facilitate early detection, i.e., position-aware attention mechanism and multi-region mean-pooling. We also demonstrate that PU-Learning can be utilized for fake news early detection based on mainly-unlabeled and imbalanced training data. The advantages of our proposed FNED model compared with baseline models indicate a promising potential for our model to be implemented in real-world social media platforms for fake news early detection. It can be applied on social media sites as a filter to automatically label potential fake news articles. Then, the labeled articles can be sent to social media administrators who will decide how to handle them afterwards. In addition, our proposed Position-Aware Attention Mechanism and Multi-Region Mean-Pooling mechanism provide novel solutions to model sequential data where time and ranking positions are important in deep learning.

CHAPTER 7

LIMITATIONS, DISCUSSIONS, CONTRIBUTIONS, FUTURE STUDIES, AND SUMMARY

In this chapter, we first discuss the limitations of our current research framework. Second, we discuss several overall key aspects regarding our entire research framework Next, we summarize the contributions of our research. Then, we propose our future research plan. Finally, we summarize our research.

7.1 Limitations

In this section, we will discuss the limitations of the proposed detection approach.

- **Small Data Size** One problem with the experimental datasets is that their size is small. Both the Weibo and Twitter dataset we used is relatively small (includes no more than 10,000 fake news samples). Small datasets can potentially cause the problem of overfitting in a machine learning task. There are public datasets for fake news detection that are larger than our experimental datasets, e.g., FakeNewsNet (Shu, Mahudeswaran, Wang, Lee, & Liu, 2018). However, they are not suitable for our study because of the following reason. In our research, we propose machine learning models to detect fake news based on user characteristics of news spreaders. Our models require that each user response to a news article is a direct retweet of the original post of the concerned news article. However, not all of the user responses collected in the FakeNewsNet dataset satisfy this requirement. In the FakeNewsNet dataset, the user responses of a news article are collected by searching all tweets that contain the keywords in the news title instead of by gathering direct retweets. Those collected user responses are not guaranteed as direct retweets to the concerned news article; furthermore, not all retweets of a concerned new article were collected by their approach. Therefore, FakeNewsNet dataset is not suitable for our study. Many other datasets have the same issue as FakeNewsNet thus are also not suitable either.

- **Data Incompleteness** Another problem with the experimental datasets is data incompleteness: a small portion of user accounts have been suspended by Twitter. In this case, their user profiles are no longer available. All data associated with these user accounts had to be removed. Thus, in our experiments, we were not able to model those users in news propagation paths, and also this resulted in even smaller than the original datasets, which might affect our models' detection effectiveness. If our detection models are implemented in real-world social media platforms, this limitation can be easily addressed since they have all the suspended users' data in the backend database.

- **Data Availability** Another limitation of this research is data availability. In this research, we only adopt user characteristics in user profiles to model source users and retweeters. However, users can be better modeled by adopting more relevant information, such as social connections and activity history. However, these data is not fully accessible by the public. For example, Twitter API can only crawl a collection of the most recent 200 tweets and retweets posted by a user. Therefore, this user's previous tweets are not accessible.

Note that the limitation of data incompleteness and data availability only exist in our experimental scenarios. For real-world social media platforms, they have full data. Therefore, if social media administrators decide to implement our model on their social media platform using full data and expand our model by incorporating user activity history, they will not have the issue of data incompleteness and data availability.

7.2 Discussions

In this section, we discuss several issues and implications for our proposed fake news early detection framework in this dissertation.

- **Applicability** The first unique feature of our proposed fake news early detection framework is that it is **content-independent**. Our models do not rely on news content to detect fake news. Thus, it is applicable to detect fake news in any format, e.g., a picture, a video, or a URL link with a short text description, as long as it is spread on a social media platform where user profiles of news spreaders are available.

- **Effectiveness on Early Detection** Early detection of fake news is of critical importance. If a detection model can only detect fake news after observing a large amount of data, it will be of marginal use in the real world, since at this moment, fake news has already been widely spread. Compared with existing detection approaches, our proposed detection framework is significantly more effective in the task of early detection mainly because of the following reasons: (1) We extract useful features from user characteristics of news spreaders to differentiate fake news from true news. User characteristics of news spreaders are much more available at the early stage of news propagation compared with other features utilized by existing approaches such as response text or propagation network. (2) We propose several novel deep learning mechanisms and incorporate them into our detection model to better extract features and learn patterns from user characteristics of news spreaders, including fake news likelihood score, position-aware attention mechanism, and multi-region mean-pooling.

- **Efficiency** Compared with most existing detection approaches, our proposed framework is more efficient. Our models do not depend on complex features that require a long time to calculate, e.g., graph decomposition of a social network (Ruchansky et al., 2017). Our deep models' structure is also not so complex. It does not require long training time. Our proposed models can be trained off-line and run in real time for early detection. Training data only needs to be updated periodically.

- **Utility** The proposed fake news early detection framework can be easily implemented in social media sites as a backend administrative tool. Detection models can be trained off-line. When a news article is posted, a pre-trained detection model will be applied to estimate its probability to be fake news. All the detected potential fake news articles can then be sent to social media administrators for verification. They will then decide how to handle verified fake news articles. Since our proposed fake news early detection framework is content-independent, it performs the first step in combating fake news, i.e., "fake news early detection". It is used to detect whether a news article is potentially fake as a whole. However, it can not tell which part of the news article is fake and why it is fake, which is the second step in combating fake news, i.e., "fake news verification".

  In addition, our user model is also useful for detecting potential fake news spreaders. Although its performances were not high in one of the four experimental datasets, we still believe its performance can be further improved, given a larger dataset and more fine-tuning. In that case, it can then be implemented in real-world applications. A social media platform can attach a label to each user's profile that indicates whether this user is likely to spread fake news. For those users who are identified as potential fake news spreaders, their future behavior needs to be paid with additional attention. For instance, news articles posted by those users will have a higher priority to be sent to our fake news detection model.

  The outcomes and findings of the proposed research can also be applied to related research topics and other more general theoretical research problems. In our FNED model, we proposed two novel deep learning mechanisms, i.e., position-aware attention mechanism and multi-region mean-pooling. These two novel mechanisms can be used to improve the classification of time-series data, where the ranking position of each data point is important.

- **Security and Robustness** Compared with existing detection approaches, our proposed detection framework is more robust against possible attacks because user profiles are more difficult to be manipulated. For example, many existing approaches detect fake news based on user comments. As a consequence, a fake news producer can easily post fake comments under the fake news articles he/she posts to cheat this kind of detection models. However, our proposed detection framework relies on user characteristics of news spreaders as a main source of data to detect fake news. If a fake news producer aims to cheat our models, then he/she needs to maintain a lot of user profiles that look normal to spread fake news. These profiles need to have a certain number of followers and followees, tweets and retweets, and interactions with other normal users instead of fake accounts. This process is both expensive and time-consuming. However, our proposed models might still be attacked if the attacker spends a large amount of money and time. This issue is beyond the scope of our current study.

7.3 Contributions

The major contributions of this research are summarized as follows:

- We are the first to systematically focus on improving the effectiveness of fake news early detection based on insufficient data.

- We demonstrated that many social media user characteristics distribute significantly differently across fake news spreaders and fake news ignorants (normal users).

- We demonstrated that by combining users' response text with user characteristics as status-sensitive crowd response, we could detect fake news more effectively than by utilizing user response or user characteristics alone.

- We demonstrated that by incorporating a user classification model to predict users' tendency to spread fake news (through the proposed fake news spreader likelihood score), our proposed fake news detection model could yield better performance.

- We demonstrated that when a news article just starts to spread, its truthfulness can be predicted based on its source user; after a news article has been retweeted many times, retweeters' user characteristics can be used to conduct more robust prediction on its truthfulness.

- We propose a novel deep learning model to detect fake news early based on a sequence of status-sensitive crowd responses. The model includes a novel position-aware attention mechanism that can learn to highlight key status-sensitive crowd responses at key ranking positions, and a novel multi-region mean-pooling mechanism that can conduct feature aggregation from multiple timeframes of the propagation path.

- We are the first to adopt PU-Learning in the problem of fake news detection to solve the issue of unlabeled and imbalanced distributed data.

7.4 Future Plans

In this section, we will introduce our future research plan, which might further improve the proposed detection approach by addressing the limitations discussed previously.

7.4.1 Adopting Dynamic User Profiling to Utilize Users' Historical Behaviors

We plan to adopt a dynamic user profiling mechanism (shown in Figure 7.1) to utilize users' historical behaviors, in order to further improve the proposed detection approach.

To generate a dynamic user profile for a specific user, we first retrieve his/her user profile and a certain number of his/her historical tweets. We do not retrieve all historical tweets because different user has a different number of historical tweets. Our model should

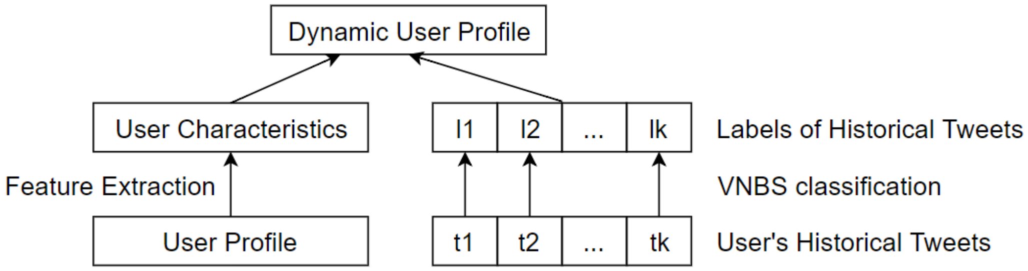


Figure 7.1 Proposed dynamic user profiling mechanism.

have a fixed parameter, i.e., the number of historical tweets. And this parameter will be tuned during the model's training process. Then, we extract user characteristics from the retrieved user profile. In the meanwhile, we use detection approach to classify each retrieved historical tweet. Then, their class labels ("true" or "fake") will be merged with the extracted user characteristics to generate a dynamic user profile.

Based on dynamic user profiling, the input of a detection model will be the news spreaders' dynamic user profiles instead of static user characteristics. Note that the detection model is used in the proposed dynamic user profiling mechanism. Thus, with the dynamic user profiling mechanism, the modified detection approach will be an iterative system.

7.5 Summary

In this dissertation, we proposed a research framework for fake news early detection on social media. Through our literature review, we found that existing machine learning-based detection approaches have a major limitation on the efficiency of early detection. They rely on either content or social context features that are usually insufficient at the early stage of news propagation.

To solve this problem, we first investigated what features are readily available at the early stage of news propagation and can also be utilized to detect fake news. We found that on social media, user characteristics of news spreaders are readily available at the early stage of news propagation since each user has a user profile, which includes his/her user characteristics. And those user profiles can be directly accessed via social media APIs. Next, we investigated whether the user characteristics of news spreaders can tell us the truthfulness of the concerned news article. Before investigating this question, we first investigated whether there exists a significant difference between the distribution of user characteristics of fake news spreaders and that of the general user population. If this significant difference indeed exists, then it will be possible to utilize user characteristics of news spreaders to detect fake news, since a fake news story often has several intentional spreaders, who often ranked top in its propagation path. We conducted hypothesis tests on the distributions of both continuous and binary user characteristics in our experimental datasets. The results demonstrated that there is indeed a significant difference between the user characteristics of fake news spreaders and that of the general user population (as well as the user characteristics of fake news ignorants and that of the general user population). Based on this finding, we further proposed a machine learning model to predict whether a user is s fake news spreader based on his/her user characteristics. The model yielded an acceptable performance. The results of our user feature study implied us to propose a machine learning model to detect fake news based on the user characteristics of its spreaders, which lead to our next researches on detection models.

After a news article is posted on social media, there will usually be a number of users who retweet it. The source user and each of its retweeters can be characterized by his/her user characteristics, in the form of a feature vector. Thus, a news propagation path can be constructed as a sequence of vectors. Since we formulate the problem of fake news detection as a binary classification problem, a machine learning model to classify sequences of vectors is required. Based on our literature review, we found that convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are both suitable for this task. Thus, we combine them together in our first proposed fake news detection model named Propagation Path Classification (PPC). We evaluated the proposed PPC model and compared it with several baselines. The experimental results showed that our PPC model outperforms the baselines on the task of fake news early detection. The main reason is that compared with text content or social context features utilized by the baselines, user characteristics are more readily available at the early stage of news propagation. Thus, our proposed detection model does not suffer from insufficient data observed at this stage. However, we still found two limitations of the first proposed PPC model. First, it simply treated source users and retweeters identically while not taking the difference of their roles in news propagation into consideration. Second, the PPC model is too sensitive to the exact retweet paths, which can be unstable because of network delays in real-world applications. Thus, we then proposed a second detection model to address these two limitations.

In our second detection model named Social Media Content Classification (SMCC), we first proposed a fake news spreader likelihood score as its intermediate output. This score incorporates the machine learning model to predict a user's tendency to spread fake news that was proposed in our user feature study (Chapter 3). Our experimental results showed that this fake news spreader likelihood score could improve the effectiveness of the SMCC model when the observed retweeters are very few. The SMCC model also has an embedding and integration mechanism that make it less sensitive to the exact retweet sequences. The experimental results showed that our SMCC model outperforms those baselines that are very sensitive to the exact retweet sequences. However, our SMCC model also has several limitations. First, it does not take users' response text to a news story into consideration. Second, it can not handle the problem of unlabeled and imbalanced data.

To further address the two limitations of our proposed SMCC model, we proposed our third detection model named Fake News Early Detection (FNED). It combines users' response text to a news story with their user characteristics to generate status sensitive crowd responses. The proposed FNED model has three major components: a Status-Sensitive Crowd Response Collector, a CNN-based News Classifier, and a PU-Learning Framework. Given a news article posted on social media, the status-sensitive crowd response collector collects each of its crowd responses and then combines them with the corresponding user profiles to generate a sequence of status-sensitive crowd responses. Then, the CNN-based news classifier first extracts both text and user features from status-sensitive crowd responses, and then concatenate them to form a feature map that represents the news article. Next, convolutional networks (CNNs) with different kernel sizes and the number of filters are applied on the feature map to extract intermediate features, which are then fed to a multi-layer perceptron (MLP) block to classify the news. The PU-Learning framework is adopted when our model is trained only with positive (fake) and unlabeled news samples. Compared with the first two models, our third model yielded better effectiveness on fake news early detection because of the following reasons. First, besides user characteristics, it also incorporates users' response text, which is another useful feature for detecting fake news. Unlike many existing detection approaches that treat user responses and user characteristics separately, our FNED model combines them together as status sensitive crowd responses to capture more accurate information from users' response to a news story. For instance, the same response text posted by different users may carry different meanings. Thus, treating user responses and user characteristics separately can not capture this difference. Another unique mechanism of our proposed FNED model is the multi-region mean pooling. It conducts mean pooling on different lengths of retweet sequences to capture different granularities of latent features from a sequence of status sensitive crowd responses. These two unique mechanisms further improved our model's effectiveness on fake news early detection. Moreover, a PU-Learning framework is also incorporated to handle the problem of unlabeled and imbalanced data. Our experimental results showed that the FNED model could also perform well in a simulated real-world scenario with unlabeled and imbalanced data.

We conducted comprehensive experiments to evaluate the proposed models on two datasets collected from Twitter and Sina Weibo, respectively. Experimental results demonstrate that our proposed models can detect fake news with over 90% accuracy within

5 minutes after it starts to spread and before it is retweeted 50 times, which is significantly faster than state-of-the-art baselines. Also, our third model requires only 10% labeled fake news samples to achieve this effectiveness under PU-Learning settings. Those advantages indicate promising potential for our models to be implemented in real-world social media platforms for fake news detection. Our proposed detection model can be applied on social media sites as a filter to label potential fake news threads automatically. Then, the labeled potential fake news threads can be sent to social media administrators who will decide how to handle them afterwards.

REFERENCES

Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, *31*(2), 211–236.

Asch, S. E., & Guetzkow, H. (1951). Effects of group pressure upon the modification and distortion of judgments. *Groups, Leadership, and Men*, 222–236.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *ArXiv PreprintarXiv:1409.0473*.

Balmas, M. (2014). When fake news becomes real: Combined exposure to multiple news sources and political attitudes of inefficacy, alienation, and cynicism. *Communication Research*, *41*(3), 430–454.

Berghel, H. (2017). Lies, damn lies, and fake news. *Computer* (2), 80–85.

Bessi, A., & Ferrara, E. (2016). Social bots distort the 2016 us presidential election online discussion.

Biyani, P., Tsioutsiouliklis, K., & Blackmer, J. (2016). “8 amazing secrets for getting more clicks”: Detecting clickbaits in news streams using article informality. In *Aaai* (pp. 94–100).

Borden, S. L., & Tew, C. (2007). The role of journalist and the performance of journalism: Ethical lessons from “fake” news (seriously). *Journal of Mass Media Ethics*, *22*(4), 300–314.

Brewer, P. R., Young, D. G., & Morreale, M. (2013). The impact of real news about “fake news”:intertextual processes and political satire. *International Journal of Public Opinion Research*, *25*(3), 323–343.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., . . . Wiener, J. (2000). Graph structure in the web. *Computer networks*, *33*(1-6), 309–320.

Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, *18*(4), 467–479.

Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on twitter. In *Proceedings of the 20th International Conference on World Wide Web* (pp. 675–684). Chen, T., Wu, L., Li, X., Zhang, J., Yin, H., & Wang, Y. (2017).

Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. ArXiv PreprintarXiv:1704.05973.

Chen, W., Zhang, Y., Yeo, C. K., Lau, C. T., & Lee, B. S. (2017). Unsupervised rumor detection based on users’ behaviors using neural networks. *Pattern Recognition Letters*.

Chen, Y., Conroy, N. J., & Rubin, V. L. (2015). Misleading online content: Recognizing clickbait as false news. In *Proceedings of the 2015 ACM Workshop on Multimodal Deception Detection* (pp. 15–19).

Chowdhury, G. G. (2003). Natural language processing. *Annual Review of Information Science and Technology, 37*(1), 51–89.

Chu, Z., Gianvecchio, S., Wang, H., & Jajodia, S. (2012). Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *IEEE Transactions on Dependable and Secure Computing, 9*(6), 811–824.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *ArXiv PreprintarXiv:1412.3555*.

Cohen, M. (2017). Fake news and manipulated data, the new gdpr, and the future of information. *Business Information Review, 34*(2), 81–85.

Conroy, N. J., Rubin, V. L., & Chen, Y. (2015). Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology, 52*(1), 1–4.

De Lathauwer, L., De Moor, B., Vandewalle, J., & by Higher-Order, B. S. S. (1994). Singular value decomposition. In *Proc. Eusipco-94, edinburgh, scotland, uk* (Vol. 1, pp. 175–178).

Del Vicario, M., Bessi, A., Zollo, F., Petroni, F., Scala, A., Caldarelli, G., . . . Quattrociocchi, W. (2016). The spreading of misinformation online. In *Proceedings of the National Academy of Sciences*, 113(3), 554–559.

Del Vicario, M., Vivaldo, G., Bessi, A., Zollo, F., Scala, A., Caldarelli, G., & Quattrociocchi, W. (2016). Echo chambers: Emotional contagion and group polarization on facebook. *Scientific Reports, 6*, 37825.

Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2016). The rise of social bots. *Communications of the ACM, 59*(7), 96–104.

Fürnkranz, J. (1998). A study using n-gram features for text categorization. *Austrian Research Institute for Artificial Intelligence, 3*(1998), 1–10.

Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with lstm.

Goldberg, Y., & Levy, O. (2014). word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *ArXiv PreprintarXiv:1402.3722*.

Guo, H., Cao, J., Zhang, Y., Guo, J., & Li, J. (2018). Rumor detection with hierarchical social attention network. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (pp. 943–951).

Gupta, A., Lamba, H., Kumaraguru, P., & Joshi, A. (2013). Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In *Proceedings of the 22nd International Conference on World Wide Web* (pp. 729–736).

Hu, X., Tang, J., Zhang, Y., & Liu, H. (2013). Social spammer detection in microblogging. In *IJCAI* (Vol. 13, pp. 2633–2639).

Jain, S., Sharma, V., & Kaushal, R. (2016). Towards automated real-time detection of misinformation on twitter. In *Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on* (pp. 2015–2020).

Jin, F., Dougherty, E., Saraf, P., Cao, Y., & Ramakrishnan, N. (2013). Epidemiological modeling of news and rumors on twitter. In *Proceedings of the 7th Workshop on Social Network Mining and Analysis* (pp. 8:1–8:9).

Jin, Z., Cao, J., Guo, H., Zhang, Y., & Luo, J. (2017). Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In *Proceedings of the 2017 ACM on Multimedia Conference* (pp. 795–816).

Jin, Z., Cao, J., Zhang, Y., Zhou, J., & Tian, Q. (2017). Novel visual and statistical image features for microblogs news verification. *IEEE Transactions on Multimedia, 19*(3), 598–608.

Kahneman, D., & Tversky, A. (2013). Prospect theory: An analysis of decision under risk. In *Handbook of the Fundamentals of Financial Decision Making: Part I* (pp. 99–127). World Scientific.

Klein, D. O., & Wueller, J. R. (2017). Fake news: A legal perspective.

Kwon, S., Cha, M., & Jung, K. (2017). Rumor detection over varying time windows. *PloS one*, *12*(1), e0168344.

Kwon, S., Cha, M., Jung, K., Chen, W., & Wang, Y. (2013). Prominent features of rumor propagation in online social media. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on* (pp. 1103–1108).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436.

Lee, J. Y., & Dernoncourt, F. (2016). Sequential short-text classification with recurrent and convolutional neural networks. *ArXiv PreprintarXiv:1603.03827*.

Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems* (pp. 2177–2185).

Li, C., & Liu, S. (2017). A comparative study of the class imbalance problem in twitter spam detection. *Concurrency and Computation: Practice and Experience*.

Liang, G., Yang, J., & Xu, C. (2016). Automatic rumors identification on sina weibo. In *Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016 12th International Conference on* (pp. 1523–1531).

Liu, X., Nourbakhsh, A., Li, Q., Fang, R., & Shah, S. (2015). Real-time rumor debunking on twitter. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management* (pp. 1867–1870).

Liu, Y., & Wu, Y. B. (2018). Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018.*

Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B. J., Wong, K.-F., & Cha, M. (2016). Detecting rumors from microblogs with recurrent neural networks. In *IJCAI* (pp. 3818–3824).

Ma, J., Gao, W., Wei, Z., Lu, Y., & Wong, K.-F. (2015). Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management* (pp. 1751–1754).

Ma, J., Gao, W., & Wong, K.-F. (2017). Detect rumors in microblog posts using propagation structure via kernel learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (Vol. 1, pp. 708–717).

Ma, J., Gao, W., & Wong, K.-F. (2018). Rumor detection on twitter with tree-structured recursive neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (volume 1: Long papers) (pp. 1980–1989).

Markines, B., Cattuto, C., & Menczer, F. (2009). Social spam detection. In Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web (pp.41-48).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems* (pp. 3111–3119).

Mnih, V., Heess, N., Graves, A., et al. (2014). Recurrent models of visual attention. In *Advances in Neural Information Processing Systems* (pp. 2204–2212).

Mustafaraj, E., & Metaxas, P. T. (2017). The fake news spreading plague: was it preventable? In *Proceedings of the 2017 ACM on Web Science Conference* (pp.

235–239).

Nguyen, T. N., Li, C., & Niederée, C. (2017). On early-stage debunking rumors on twitter: Leveraging the wisdom of weak learners. In *International Conference on Social Informatics* (pp. 141–158).

Nickerson, R. S. (1998). Confirmation bias: A ubiquitous phenomenon in many guises. *Review of General Psychology*, *2*(2), 175.

Nyhan, B., & Reifler, J. (2010). When corrections fail: The persistence of political misperceptions. *Political Behavior*, *32*(2), 303–330.

Paul, C., & Matthews, M. (2016). The Russian “firehose of falsehood” propaganda model. *RAND Corporation*.

Popat, K. (2017). Assessing the credibility of claims on the web. In *Proceedings of the 26th International Conference on World Wide Web Companion* (pp. 735–739).

Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., & Stein, B. (2017). A stylometric inquiry into hyperpartisan and fake news. *ArXiv PreprintarXiv:1702.05638*.

Qazvinian, V., Rosengren, E., Radev, D. R., & Mei, Q. (2011). Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 1589–1599).

Reed, E. S., Turiel, E., & Brown, T. (2013). Naive realism in everyday life: Implications for social conflict and misunderstanding. In *Values and Knowledge* (pp. 113–146). Psychology Press.

Rubin, V., Conroy, N., Chen, Y., & Cornwell, S. (2016). Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the Second Workshop on Computational Approaches to Deception Detection* (pp. 7–17).

Rubin, V. L. (2017). Deception detection and rumor debunking for social media. *The SAGE Handbook of Social Media Research Methods*, 342.

Rubin, V. L., Chen, Y., & Conroy, N. J. (2015). Deception detection for news: three types of fakes. *Proceedings of the Association for Information Science and Technology*, *52*(1), 1–4.

Ruchansky, N., Seo, S., & Liu, Y. (2017). Csi: A hybrid deep model for fake news. *ArXiv PreprintarXiv:1703.06959*.

Sampson, J., Morstatter, F., Wu, L., & Liu, H. (2016). Leveraging the implicit structure within social media for emergent rumor detection. In *Proceedings of the 25th Acm International Conference on Information and Knowledge Management* (pp. 2377–2382).

Shannon, C. E., & Weaver, W. (1963). The mathematical theory of communication. 1949. *Urbana, IL: University of Illinois Press*.

Shi, S.-R., Liu, C., Perez, J., & Taylor, C. R. (2005). Protein-embedding technique: a potential approach to standardization of immunohistochemistry for formalin-fixed, paraffin-embedded tissue sections. *Journal of Histochemistry & Cytochemistry*, *53*(9), 1167–1170.

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2018). Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *ArXiv PreprintarXiv:1809.01286*.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, *19*(1), 22–36.

Shu, K., Wang, S., & Liu, H. (2018). Understanding user profiles on social media for fake news detection. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)* (pp. 430–435).

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.

Sun, S., Liu, H., He, J., & Du, X. (2013). Detecting event rumors on sina weibo automatically. In *Web Technologies and Applications* (pp. 120–131). Springer.

Sundar, S. S., & Nass, C. (2001). Conceptualizing sources in online news. *Journal of Communication, 51*(1), 52–72.

Tajfel, H., & Turner, J. C. (1979). An integrative theory of intergroup conflict. *The Social Psychology of Intergroup Relations*, *33*(47), 74.

Tajfel, H., & Turner, J. C. (1986). The social identity theory of intergroup behavior. *The Social Psychology of Intergroup Relations*, 7–24.

Tversky, A., & Kahneman, D. (1992). Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, *5*(4), 297–323.

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

Wang, D., Irani, D., & Pu, C. (2011). A social-spam detection framework. In *Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference* (pp. 46–54).

Wang, Y., Ma, F., Jin, Z., Yuan, Y., Xun, G., Jha, K., . . . Gao, J. (2018). Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 849–857).

Willnat, L., & Weaver, D. H. (2014). *The American Journalist in the Digital Age: Key Findings. Bloomington, in: School of Journalism, Indiana University*. Abgerufen unter: http://news. indiana. edu/releases/iu/2014/05/2013-american-journalist-key-findings. pdf [18.03. 2016].

Wu, K., Yang, S., & Zhu, K. Q. (2015). False rumors detection on sina weibo by propagation structures. In *Proceedings of the 31st IEEE International Conference on Data Engineering.*

Wu, L., Li, J., Hu, X., & Liu, H. (2017). Gleaning wisdom from the past: Early detection of emerging rumors in social media. In *Proceedings of the 2017 SIAM*

International Conference on Data Mining (pp. 99–107).

Wu, L., & Liu, H. (2018). Tracing fake-news footprints: Characterizing social media messages by how they propagate. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (pp. 637–645).

Wu, L., Morstatter, F., Hu, X., & Liu, H. (2016). Mining misinformation in social media. *Big Data in Complex and Social Networks,* 123–152.

Yan, S., Xu, D., Zhang, B., & Zhang, H.-J. (2005). Graph embedding: A general framework for dimensionality reduction. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (Vol. 2, pp. 830–837).

Yang, F., Liu, Y., Yu, X., & Yang, M. (2012). Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics* (p. 13).

Zajonc, R. B. (1968). Attitudinal effects of mere exposure. *Journal of Personality and Social Psychology*, *9*(2p2), 1.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *ArXiv PreprintarXiv:1212.5701*.

Zhang, H., Alim, M. A., Li, X., Thai, M. T., & Nguyen, H. T. (2016). Misinformation in online social networks: Detect them all with a limited budget. *ACM Transactions on Information Systems (TOIS)*, *34*(3), 18.

Zhang, Q., Zhang, S., Dong, J., Xiong, J., & Cheng, X. (2015). Automatic detection of rumor on social network. In *Natural Language Processing and Chinese Computing* (pp. 113–122). Springer.

Zhao, Z., Resnick, P., & Mei, Q. (2015). Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 1395–1405).

Zhou, X., Cao, J., Jin, Z., Xie, F., Su, Y., Chu, D., . . . Zhang, J. (2015). Real-time news certification system on sina weibo. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 983–988).

Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., & Procter, R. (2018). Detection and

resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR), 51*(2), 32.

Zubiaga, A., Liakata, M., & Procter, R. (2017). Exploiting context for rumour detection in social media. In *International Conference on Social Informatics* (pp. 109–123).