FNED: A Deep Network for Fake News Early Detection on Social Media

YANG LIU and YI-FANG BROOK WU, New Jersey Institute of Technology

The fast spreading of fake news stories on social media can cause inestimable social harm. Developing effective methods to detect them early is of paramount importance. A major challenge of fake news early detection is fully utilizing the limited data observed at the early stage of news propagation and then learning useful patterns from it for identifying fake news. In this article, we propose a novel deep neural network to detect fake news early. It has three novel components: (1) a status-sensitive crowd response feature extractor that extracts both text features and user features from combinations of users' text response and their corresponding user profiles, (2) a position-aware attention mechanism that highlights important user responses at specific ranking positions, and (3) a multi-region mean-pooling mechanism to perform feature aggregation based on multiple window sizes. Experimental results on two real-world datasets demonstrate that our proposed model can detect fake news with greater than 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. Most importantly, our approach requires only 10% labeled fake news samples to achieve this effectiveness under PU-Learning settings.

CCS Concepts: • **Information systems** → **Social networks**; • **Computing methodologies** → **Neural networks**;

Additional Key Words and Phrases: Fake news detection, social media, deep learning

ACM Reference format:
Yang Liu and Yi-Fang Brook Wu. 2020. FNED: A Deep Network for Fake News Early Detection on Social Media. *ACM Trans. Inf. Syst.* 38, 3, Article 25 (May 2020), 33 pages.
https://doi.org/10.1145/3386253

1 INTRODUCTION

As social media becomes more and more popular, people tend to consume news more often from it than from traditional news media. 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, and (iv) it requires less content censorship for a news article to be posted on social media. However, these advantages meanwhile enable "fake news," i.e., news carrying intentionally and verifiably false information to spread widely and rapidly among social media users. Researchers

Y. Liu and Y.-F. B. Wu Contributed equally to this research.

Authors' address: Y. Liu and Y.-F. B. Wu, New Jersey Institute of Technology, 323 Dr. M.L.K. Jr. Boulevard, Newark, NJ 07102; emails: {yl558, yi-fang.wu}@njit.edu.

1046-8188/2020/05-ART25 $15.00
https://doi.org/10.1145/3386253

found that fake news spreads significantly farther, faster, deeper, and more broadly than true news [Vosoughi et al. 2018]. Two different studies conducted in 2016 found that 23% of Americans say they have shared fake news stories, either knowingly or unknowingly.[1] It was reported that in 2017, 67% of American adults consumed news mainly from social media.[2] However, the fast and massive spreading of fake news can rapidly cause inestimable social harm. For example, in 2013, a hacker's false Associated Press (AP) tweet claiming that an "explosion" had injured President Obama, causing the DOW to briefly plunge more than 140 points within 6 minutes. The estimated temporary loss of market cap in the S&P 500 alone totaled $136.5 billion.[3] 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 almost three out of five Americans believe that fake news is a serious threat to their financial decision making.[4] The phrase "fake news" was declared the official Collins Dictionary Word of the Year for 2017.[5] It is crucial to stop fake news before it reaches a broad audience to mitigate and minimize its negative effects. One of the key steps to achieve this goal is *early detection of fake news*, i.e., detecting fake news at its early propagation stage.

Human efforts have been involved in detecting and combating fake news. Fact-checking sites, e.g., Snopes[6], Politifact[7], and Factcheck.org[8], rely on human experts to manually judge the truthfulness of controversial news stories. The judging results are then released to the public as a reference for fact-checking. After the 2016 election, Google and Facebook also took steps to combat fake news. Facebook enables users to mark news stories as fake.[9] A marked news story will then go through a fact-checking process: it will be attached with a warning label below its link to discourage users from sharing it, if the news story is confirmed to be fake news. Google enhanced its search function by displaying the fact-check result conducted by news publishers and fact-checking organizations under the snippet of news stories.[10] Although manual fact-checking can indeed help readers identify fake news, doing so is far from meeting the goal of fake news early detection for the following reasons: (i) manual fact-checking is inefficient to detect and report fake news early because it is time consuming, and (ii) manual fact-checking is not scalable to handle a large volume of fake news produced on the Internet.

With the fast development of machine learning and deep learning [LeCun et al. 2015] techniques in recent years, automatic machine learning–based detection approaches have become a major alternative to manual fact-checking and have attracted significant attention both from research communities and the industry. Plenty of existing studies focus on automatic detection of fake news [Kwon et al. 2017; Ma et al. 2016, 2017; Ruchansky et al. 2017; Shu et al. 2017; Wu et al. 2017], as well as closely related topics, e.g., rumor detection [Sampson et al. 2016; Wu et al. 2015, 2017], misinformation detection [Jain et al. 2016b; Qazvinian et al. 2011; Zhang et al. 2016], and social spam detection [Hu et al. 2013; Li and Liu 2017; Markines et al. 2009; Wang et al. 2011]. Most machine-learning-based detection approaches follow the standard workflow: given a news article, a series of features are extracted from some data either related to the news content or its social context, e.g., user engagements around the news article. Next, a machine-learning model is

[1] https://www.consumer-action.org/english/articles/fake_news.
[2] http://www.journalism.org/2017/09/07/news-use-across-social-media-platforms-2017/.
[3] https://www.cnbc.com/id/100646197.
[4] https://www.aicpa.org/press/pressreleases/2017/fake-financial-news-is-a-real-threat-to-majority-of-americans-new-aicpa-survey.html.
[5] http://www.newsweek.com/fake-news-word-year-collins-dictionary-699740.
[6] https://www.snopes.com/.
[7] http://www.politifact.com/.
[8] https://www.factcheck.org/.
[9] https://www.facebook.com/help/572838089565953?helpref=faq_content.
[10] https://blog.google/products/search/fact-check-now-available-google-search-and-news-around-world/.

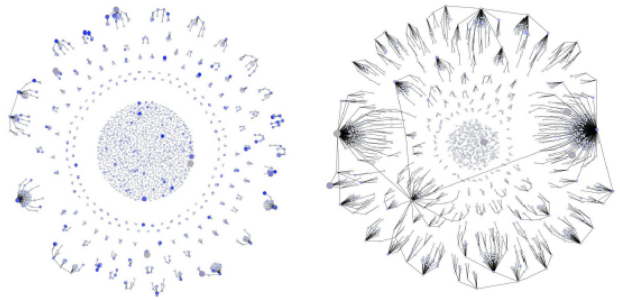
trained to classify the news as true or fake based on the extracted features. Therefore, a machine-learning-based detection approach can be characterized by the machine-learning model it uses and the features that are fed to the model.

Early approaches [Castillo et al. 2011; Gupta et al. 2014; Jin et al. 2017b; Qazvinian et al. 2011] extract textual features from news content to build classifiers for identifying fake news. However, these approaches have not been proven effective yet, as fake news is intentionally written to mislead readers to believe false information, which makes it difficult to detect solely based on news content [Shu et al. 2017]. Recent approaches extract discriminative features from the *crowd responses* to a news article posted on social media, e.g., user retweets, likes, reviews, and comments, to detect fake news. One reason is that those crowd responses consist of the social context of the concerned news. They contain information about how a news article is circulated on social media and how people interact with it. Thus, the social context information might give us more of a clue about the truthfulness of a news article compared with textual content only. Diverse crowd response types lead to a variety of detection models that utilize different types of crowd response features. Textual [Chen et al. 2017; Ma et al. 2015, 2016] and structural [Jin et al. 2013; Ma et al. 2017; Wu et al. 2015] crowd response features are the two categories that are most deeply investigated. Textual features are usually extracted from a sequence of user comments or reviews. Structural features can be extracted from an information propagation network that is constructed from users' sharing, liking, and retweeting behaviors. Then, different machine-learning models are designed to fit various types of features.

From our literature review, we found one significant limitation of the existing machine-learning-based detection approaches: 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 that no research focuses on *early detection effectiveness* when the required data is usually insufficient at this stage. The main reason is that 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 would likely be low. Without sufficient relevant data, a machine-learning model is prone to overfitting. However, 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. If a detection approach cannot effectively detect fake news shortly after it starts to spread, even if it has high optimal effectiveness in experimental settings, it will still have marginal use in the real world.

The following is an example showing why an existing detection approach that is effective given enough relevant data is ineffective in early detection when relevant data is insufficient. A recent work [Ma et al. 2016] adopts recurrent neural networks (RNNs) 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. According to the statistics of their experimental datasets, the average number of posts per event is 1,111 in the Twitter dataset and 816 in the Weibo[11] dataset. After we investigated their datasets, we found that the average number of posts per event at 24

[11] https://weibo.com.



(a) Bigfoot (rumor) (b) Summize (non-rumor)

Fig. 1. The propagation network of a fake news article (a) and a true news article (b).

hours after a news article starts to spread is around 500 in the Twitter dataset and 400 in the Weibo dataset. Therefore, their approach requires around 400 to 500 relevant posts to accurately detect fake news. However, 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 fewer than required, their approach's performance drops significantly. Recall the fake tweet example that we discussed before: fake news caused significant damage to the stock market within 5 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] extracted a series of structural features from the propagation networks, e.g., median in-degree and median out-degree, to detect fake news. Figure 1 shows the propagation network of a fake news event named *Bigfoot* and a true news event named *Summize*. These two propagation networks are constructed from a large amount of propagation data. According to the statistics of their datasets reported in their work, 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 2,054 and 4,367,672, respectively. In such a condition, the two networks have significantly different structural features, and their structural difference can be easily recognized by human eyes. Thus, it is easy for a machine-learning model to differentiate between these two networks. However, when the two concerned news articles first start to spread, only a small propagation network can be observed, which is the center circle of the two large networks. Since it is usually difficult 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 structure looks identical via human eyes. Thus, it might be difficult for a machine-learning model to differentiate between these two small networks. Moreover, their work only reported overall detection effectiveness without the corresponding detection deadline. Thus, their approach's performance on early detection remains unknown.

With a lack of early detection capability, a machine-learning-based detection approach will have marginal usefulness since 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 early detection performance, training data is another issue. Most existing studies only present detection performance results on fully labeled and balanced experimental datasets. However, real-world data is expected to be mostly unlabeled and extremely imbalanced, since verified fake news stories account for 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. However, the outcome is expected to be seen in 5 to 10 years.[12]

To address the research problem of fake news early detection, we first investigated what social context features of a news article are both readily available at the early stage of news propagation and have the ability to differentiate fake news from true news. We noticed that user profiles of news spreaders on social media are readily available even before news propagation. A user profile includes a variety of user characteristics/features that might reflect whether a user is likely to spread fake news. Existing studies have adopted user characteristics to detect fake news [Castillo et al. 2011; Yang et al. 2012]. As a recent study [Shu et al. 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 for adopting user characteristics for fake news detection. 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 frequently spread real news pieces and users who frequently spread fake news pieces in their experimental dataset. Although the study mentioned previously yields convincing results, it has the following limitations: (i) it only examines a subset of user profile features; (ii) it does not differentiate source users, i.e., users who initially post a news piece, from news retweeters; and (iii) it has not investigated whether user features can be utilized to predict whether a user is likely to spread fake news. To address these limitations, in this work, we conducted a more comprehensive user feature study based on our experimental datasets. Our statistical analysis of user features also demonstrates that many user features distribute significantly differently between users who have spread fake news and users who have never spread fake news. Our previous research [Liu and Wu 2018] showed that user characteristics of news spreaders can indeed be utilized to improve fake news early detection. In this study, we investigate how to combine user features with other social context data to further improve fake news early detection.

Besides features, data is another issue in fake news detection. To handle the inconsistency between fully labeled, balanced experimental datasets and mainly unlabeled, imbalanced real-world data, we investigate the application of PU-Learning [Li and Liu 2005] algorithms in fake news detection. PU-Learning refers to learning from positive and unlabeled examples. In such tasks, there is no labeled negative training data. Existing PU-Learning algorithms [Lee and Liu 2003; Li and Liu 2003; Liu et al. 2002, 2003] mainly use a two-step strategy: (i) identifying a set of reliable pseudo-negative examples from the unlabeled examples and (ii) building a classifier by iteratively applying a classification algorithm on the positive examples and the pseudo-negative examples.

In an attempt to address the early detection problem, in this study we investigate the following research questions:

RQ1: Whether user characteristics of news spreaders can be utilized to improve the effectiveness of fake news early detection?

If they can be, then:

RQ2: What machine learning model is suitable for detecting fake news early based on news spreaders' user characteristics?

RQ3: Can PU-Learning be utilized for fake news early detection based on mainly unlabeled and imbalanced training data?

[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.

To address these research questions, in this article we propose a novel end-to-end deep neural network model named *FNED* (Fake News Early Detection). A news article posted on social media is modeled as a sequence of status-sensitive crowd responses, each of which is a combination of a piece of text response and a user profile. Textual and user features are extracted to represent each status-sensitive crowd response by a text CNN and embedding block, respectively. A convolution network with a position-aware attention mechanism and multi-region mean pooling is then applied to learn hidden patterns from the sequence of status-sensitive crowd responses and then classify the news article. Experimental results on two real-world datasets collected from Twitter and Weibo demonstrate that our proposed model can detect fake news with greater than 90% accuracy within 5 minutes after it starts to spread, which is significantly faster than state-of-the-art baselines. Our model is also evaluated under PU-Learning settings, i.e., it is iteratively trained based on positive and unlabeled data samples only, to best mimic real-world scenarios. The results show that our model requires only 10% labeled fake news samples in the datasets to achieve the same effectiveness in the PU-Learning setting.

The main contributions of this article can be summarized as follows:

- We demonstrate that social media user characteristics distribute significantly differently between fake news spreaders and fake news ignorants. This difference is especially significant on retweeters, and it can be utilized to improve the effectiveness of fake news early detection.
- We propose a novel deep learning framework (FNED) for fake news early detection based on status-sensitive crowd responses. It incorporates two novel deep learning mechanisms that facilitate early detection: the position-aware attention mechanism and multi-region mean pooling.
- We are the first to adopt PU-Learning in fake news early detection. Moreover, we demonstrate that PU-Learning can be utilized to improve the effectiveness of fake news early detection when simulating a real-world scenario where data is unlabeled and imbalanced.

The remainder of this article is organized as follows. In Section 2, we introduce existing fake news detection approaches reported in the literature. In Section 3, we formally present our proposed approach in detail. In Section 4, we introduce our experimental procedures, and then we present and analyze the results. Finally, in Section 5, we present conclusions, discussions, and future directions of our research.

2 RELATED WORKS

Fake news refers to a news article that is intentionally and verifiably false [Shu et al. 2017]. In recent years, fake news detection on social media has been drawing growing attention from both the research community and the general public, and AI-based automatic detection has become a major focus. In this section, we present an overview of existing studies that focus on the features used in automatic fake news detection, as well as closely related topics such as rumor detection or misinformation detection. Meanwhile, we discuss their limitations on early detection of fake news.

2.1 Detection Approaches Based on News Content Features

An intuitive and straightforward approach adopted by many existing studies is to detect fake news based on the news content. Most existing studies focus on the text content of news stories, e.g., news headline and body text, whereas a few of them investigate image/video content [Jin et al. 2017b]. Castillo et al. [2011] adopted a list of rudimentary content-based features, e.g., question marks, emoticon symbols, sentiment positive/negative words, and pronouns, to gauge information credibility on Twitter. Gupta et al. [2014] adopted the number of swear words and self-pronouns as

indicators of fake news. Popat [2017] found that the language style of an article plays a crucial role in understanding its credibility. Afroz et al. [2012] detected online hoaxes, frauds, and deception based on writing styles. These studies adopt language stylistic features, e.g., assertive verbs, factive verbs, and implicatives, to assess the credibility of web claims. Those linguistic stylistic features mentioned previously do not carry semantic meaning and are prone to manipulation. Thus, those approaches are less likely to succeed in real-world applications. Rubin et al. [2016] used satirical cues to detect potentially misleading news. Natural language processing (NLP) techniques [Chowdhury 2003] have also been adopted by existing studies to discover syntactical or semantical patterns from news content to detect fake news. Qazvinian et al. [2011] adopted n-grams of lexicons and part-of-speech tags extracted from microblog content as features to identify rumors. Zubiaga et al. [2017] adopted Word2Vec [Mikolov et al. 2013] to create vector representations of words in tweets to detect rumors. A common challenge for content-based detection approaches is that the content of fake news is diverse in terms of topic, style, and platform. In addition, news content features can be event specific [Gupta et al. 2012; Sun et al. 2013]. Thus, content-based features that work well on one particular fake news dataset may not work well on another. Furthermore, machine-learning models based on news content features have the generalizability issue [Tolosi et al. 2016].

2.2 Detection Approaches Based on Social Context Features

The interactive attribute of social media enables a variety of social engagements surrounding a news story. After a news story is released on social media, users can share, comment, and discuss it with their neighborhood users. These social engagements form the social context of a news story, which contains potential insights into the truthfulness of a news story. Plenty of existing studies extract features from social context to detect fake news.

The three most common types of social context features are user-based, text-based, and structural-based features. User-based features can be extracted from user profiles on social media, which reflect the characteristics of social media users. Early studies adopted 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, and registration age, to gauge the credibility of the information posted by its source user. Besides the common user features, Yang et al. [2012] added some unique user features that are supported by Sina Weibo, a Chinese social media platform, e.g., gender and registration place, to detect rumors. Simply using user-based features of the source user who releases a news story to judge whether the news story is fake has a significant limitation: fake news producers usually mix a fake news story with more true news stories to increase the chance of their fake news being trusted. Thus, user-based features of source users alone cannot give us a complete picture of whether a news story is fake. User-based features of news spreaders, e.g., users who share/retweet a news story, however, might potentially give us more insight into the truthfulness of a news story. However, this type of feature is neglected by most existing studies.

Text-based social context features can be extracted from users' comments and discussions under a news story on social media. As user comments are timestamped, a variety of temporal-based features extracted from time series of user comments have been proposed to detect fake news. Ma et al. [2015] detected rumors using time series of content and social context-based features, e.g., percentage of microblogs with URL and percentage of verified users. However, those aggregated-level features require a large number of observations to be statistically significant, which are not suitable for fake news early detection. Jin et al. [2017a] conducted news verification by exploiting conflicting social viewpoints in microblogs. Zhao et al. [2015] conducted early detection of rumors in social media based on inquiry posts. Lukasik et al. [2019] proposed a Gaussian process classifier

to classify users' stance toward rumors on social media, and then flagged highly disputed rumors as being potentially false. Recent works adopted deep learning techniques like RNN to extract temporal-linguistic patterns from sequences of user comments [Chen et al. 2017; Ma et al. 2016] to identify rumors. One major limitation of these approaches is that user comments can be very few at the early stage of a news story's propagation process, which can significantly affect the performance of RNN models and easily cause them to overfit.

Social media users are connected through either directed or undirected links, e.g., following and friendship. Thus, when a news story spread through these links, a propagation network can be observed. Structural features extracted from propagation networks have been investigated by existing studies 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. They also proposed a hierarchical propagation model for news credibility evaluation on microblog [Jin et al. 2014]. Liu et al. [2015b, 2016] detected rumors through modeling information propagation networks on social media. Wang and Terano [2015] proposed a graph-based pattern matching algorithm to detect rumor patterns from streaming social media data. Wu et al. [2015] proposed a graph kernel-based SVM classifier that learns high-order propagation patterns to detect fake news. Yang at al. [2015a] exploited the topology property of social networks for rumor detection. 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 that captures high-order patterns differentiating different types of fake news by evaluating the similarities between their propagation tree structures.

Liu at al. [2017a] detected rumors based on the difference of diffusion patterns of rumors and non-rumors. Propagation network-based features can also be utilized for other tasks related to fake news detection, e.g., rumor source detection [Jain et al. 2016a; Spencer and Srikant 2016; Zheng and Tan 2015], rumor spreading control [Dhar et al. 2016], and budget optimization for misinformation detection [Zhang et al. 2016]. However, detecting fake news or rumors 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.

2.3 Detection Approaches Based on Other Features

Besides the commonly used features for detecting fake news, there are other types of features explored by a few studies to detect fake news from a unique angle. Conroy et al. [2015] identified fake news by detecting deceptions. Rubin et al. [2015, 2017] also proposed methods for detecting and debunking deceptions on social media. Zhang et al. [2015] proposed several implicit features, including popularity orientation, internal and external consistency, sentiment polarity, among others, to detect rumors on social media. Language-specific features have also been explored [El Ballouli et al. 2017]. Galitsky [2015] detected rumor and disinformation by web mining, which relies on external textual information. Yang et al. [2015b] detected emerging rumors through hot topic detection. Zubiaga et al. [2016] utilized reporting dynamics during breaking news for rumor detection in social media. Jin et al. [2017b] explored visual and statistical image features for microblogs news verification. Wu et al. [2017] investigated whether knowledge learned from historical data could potentially help identify newly emerging rumors. There are also many hybrid models that detect fake news based on multiple categories of features [Jin et al. 2017a; Kwon et al. 2013; Ruchansky et al. 2017; Vosoughi 2015].

2.4 The Issue with Experimental Data

The class imbalance problem in real-world social media data is also a key challenge in detecting fake news since the volume of true news stories is assumed to be much larger than that of fake

news stories. Ensemble learning and multiple data sampling techniques have been explored to solve this problem [Liu et al. 2017b; Tang et al. 2019]. Another issue is the lack of labeled data. Labeling fake news stories requires human expertise and valid verification. It is time consuming to construct a large fully labeled news dataset. To solve this problem, unsupervised machine-learning approaches and generative models have been explored [Ma et al. 2019; Yang et al. 2019]. We find that PU-Learning, i.e., learning with positive and unlabeled data [Hsieh et al. 2015; Li and Liu 2005; Liu et al. 2003], might be more suitable for the task of detecting fake news on social media, as this approach requires only a small number of labeled fake news stories and a relatively large number of unlabeled news stories. However, few existing studies focus on this direction.

3 METHODOLOGY

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

3.1 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, e.g., "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 to 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 $R(a_i) = ((u_0, r_0, t_0), (u_1, r_1, t_1), \ldots, (u_n, r_n, t_n))$. Each tuple $(u_k, r_k, t_k) \in R(a_i)$ represents the k-th crowd response. In other words, 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 a news article. We also call the user u_0 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 refers to a set of user characteristics of that user 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 $R(a_i) = ((u_0, S(u_0, t_0), r_0, t_0), (u_1, S(u_1, t_1), r_1, t_1), \ldots, (u_n, S(u_n, t_n), r_n, t_n))$.

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 for the following two reasons. First, the number of crowd responses can be directly incorporated into the machine learning model as a parameter. Second, 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 $R(a_i, k) = ((u_0, S(u_0, t_0), r_0, t_0), \ldots, (u_k, S(u_k, t_k), r_k, t_k))$. 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

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

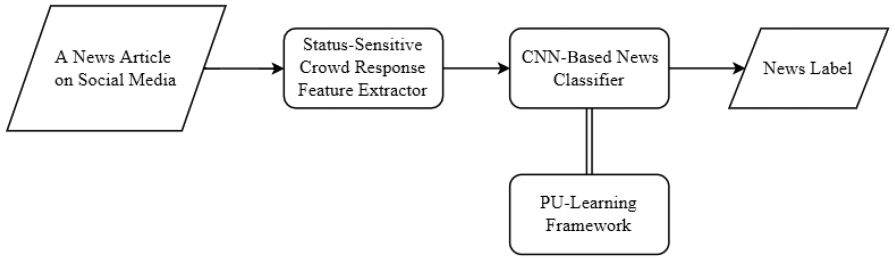


Fig. 2. The flowchart of our proposed fake news early detection model.

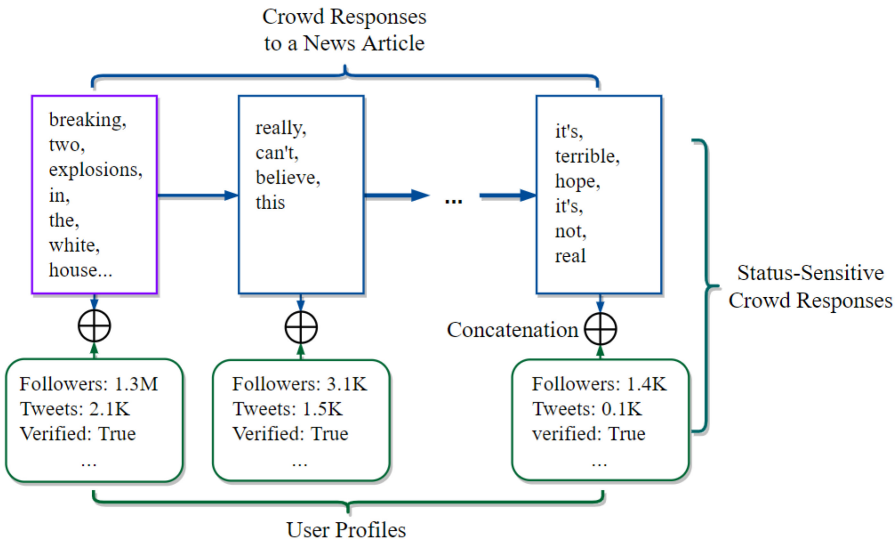


Fig. 3. The status-sensitive crowd responses to a given news article.

3.2 Model Overview

Our proposed fake news detection model has three major components: a *status-sensitive crowd response feature extractor* (shown in Figure 4), a *CNN-based news classifier* (shown in Figure 5), and a *PU-Learning framework* (shown in Figure 6).

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, the status-sensitive crowd response feature extractor extracts both text features and user features from status-sensitive crowd responses, then concatenates 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 *FNED*. Figure 2 shows the flowchart of our proposed detection model.

3.3 Status-Sensitive Crowd Response Feature Extractor

Figure 3 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.

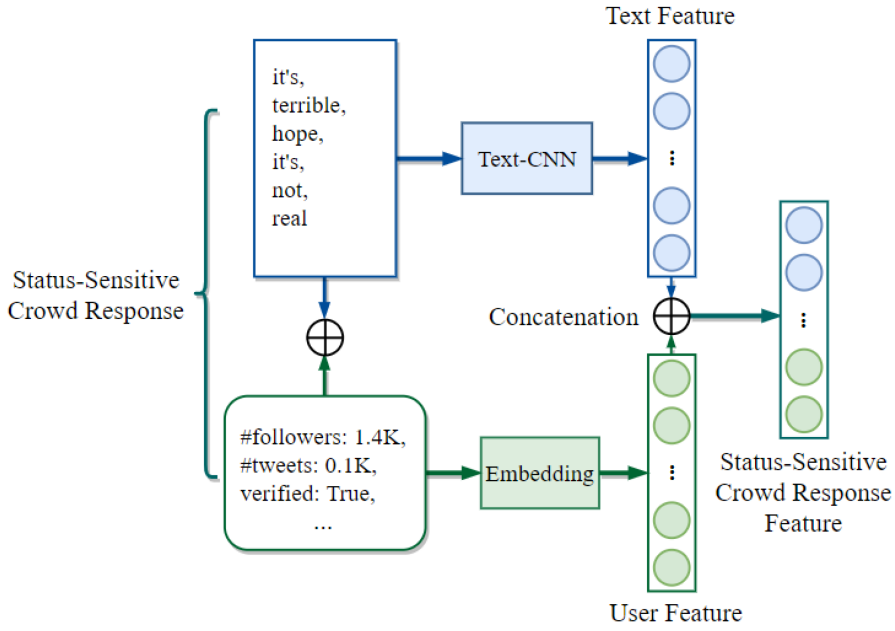


Fig. 4. The architecture of the status-sensitive crowd response feature extractor.

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.

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 [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 4.

3.4 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 convolutional neural 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 5 shows the architecture of CNN-based news classifier.

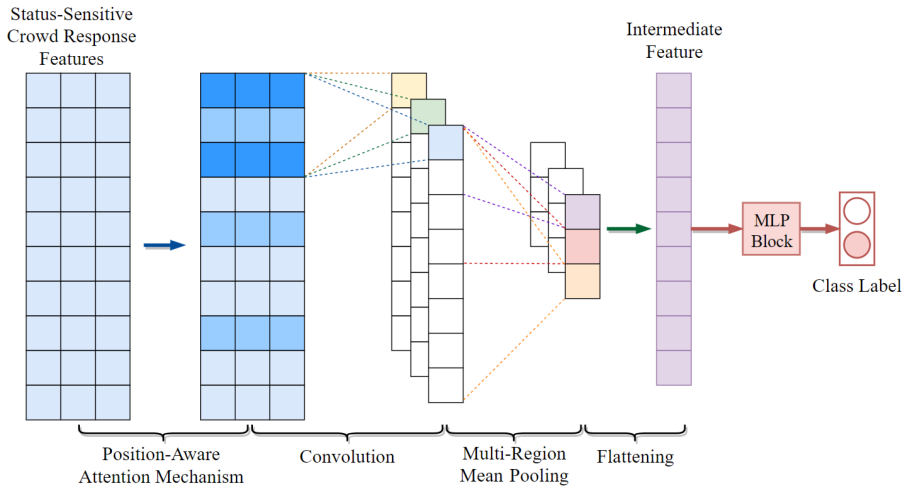


Fig. 5. The architecture of the CNN-based news classifier.

3.4.1 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 more significantly, thus should be somehow highlighted in the entire propagation path. Thus, our detection model should learn how much attention should be given to each status-sensitive crowd response. We propose a position-aware attention mechanism, which is an extension of the basic attention mechanism [Bahdanau et al. 2014; Mnih et al. 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:

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

$$F_w(\mathbf{r}'_j) = \text{Relu}\left(\mathbf{W}_{aj}^T \mathbf{r}'_j + \mathbf{b}_{aj}\right),$$

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

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

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 $\mathbf{W}_a$, bias $\mathbf{b}_a$, α_j is the normalized attention weight of the j-th status-sensitive crowd response via a softmax function, and $\mathbf{r}''_j$ is the transformed status-sensitive crowd response feature vector after our position-aware attention mechanism is applied. The difference between our proposed position-aware attention mechanism and the basic attention mechanism is that in our approach 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.

3.4.2 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 the number of filters l is applied to extract intermediate features. Specifically, each convolutional filter with window size $d \times h$ takes the contiguous h status-sensitive crowd response feature vectors as the input and produces one scalar feature as output:

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

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:

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

3.4.3 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 $k - h + 1$ 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 for the following reasons. First, multi-region mean pooling can capture different granularities of aggregated features from the entire feature map, whereas basic mean pooling can only calculate one overall average. Second, 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, 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:

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

3.4.4 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

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

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

3.5 Optimization

We denote our CNN-based news classifier as $H(\cdot;\theta)$, where θ denotes all of 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 minimizes the detection loss:

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

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

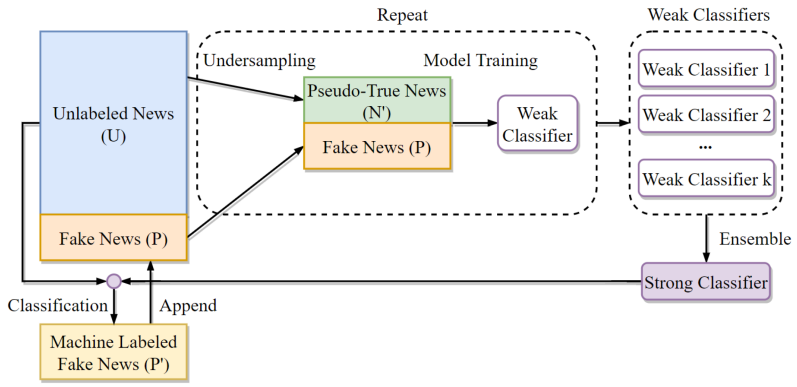


Fig. 6. The architecture of our proposed PU-Learning framework.

3.6 The PU-Learning Framework

Figure 6 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, to best simulate 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 is supposed to be unknown. The size of positive news samples is supposed to be much smaller than the size of unlabeled news samples, i.e., $|P| < |U|$. Among the unlabeled news samples, the size of positive unlabeled (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 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 positive-labeled fake news samples to update the collection of positive-labeled fake news samples, i.e., $P \Leftarrow P + P'$. Afterward, we repeat the procedure of undersampling, weak classifier training, ensemble classification, and positive sample updating again.

The reason we run our PU-Learning process iteratively to extend the fake news dataset with the machine labeled fake news is as follows. Under the PU-Learning settings, the size of positive (fake) news samples (P) is expected to be small. Thus, in the first iteration of our PU-Learning algorithm, the size of pseudo-true news samples (N') is also expected to be small since $|N'| = |P|$. As a consequence, the training data, i.e., positive (fake) news samples (P) plus pseudo-true news samples (N'), may not be sufficiently large to build the best classifier. That is why we call each classifier trained at this stage the *weak classifier*. By extending the positive (fake) news samples (P) with the machine-labeled fake news (P') that have a high confidence level, a larger training dataset can be built to train a better classifier in the next iteration.

Table 1. Statistics of the Experimental Datasets

Statistic	Twitter15	Weibo16
# News articles	680	4,664
# True news	353	2,351
# Fake news	327	2,313
# Source users	277	2,309
# Retweeters	215,691	2,818,002

However, there is a risk that some incorrect labels will be included in positive (fake) news samples (P), which may bring noise into the classification problem. To handle the trade-off between the size of the training dataset and the amount of noise it contains, we adopt an early stop mechanism to stop extending the positive (fake) news sample set (P) when the iterative training process stop improving. Early stop is triggered when the accuracy of the strong classifier on the validation dataset does not improve after five iterations. Then, we will pick the strong classifier that performs the best within the latest five iterations as our final strong classifier. Iteratively building a PU-Learning classifier by extending the training dataset with a stop condition is widely adopted by existing studies on PU-Learning problems [Li and Liu 2003; Liu et al. 2003; Yu et al. 2002].

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).

4 EXPERIMENTS AND RESULTS

4.1 Dataset

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. We directly adopted a public Weibo dataset [Ma et al. 2016] consisted of 2,351 true news and 2,313 fake news collected during 2015–2016, because it provided all necessary information for our study, especially user characteristics. We named this dataset Weibo16 in this article. We also found a public Twitter dataset [Ma et al. 2017]: it consisted of two parts, i.e., Twitter15 and Twitter16, which were constructed based on two reference datasets collected in 2015 [Liu et al. 2015a] and 2016 [Ma et al. 2016] respectively. We slightly modified this Twitter dataset by the following steps and finally regenerated our own First, we removed the tweets labeled as "unverified" or "true rumor" since they were beyond our research interest. Second, we removed the tweets that were no longer available to be accessed, as we needed to collect their corresponding features that were not available in the original dataset, for model training. Third, we eventually discarded the original Twitter16 dataset, because the number of remaining tweets was too small (309), and it was 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. Fourth, we developed a crawler to acquire corresponding user profiles for each of the remaining retweets. Retweeters whose user profiles were no longer available to be crawled were removed from news propagation paths. Fifth, we augmented the remaining dataset that contained of 353 true news and 327 fake news with user features extracted from crawled user profiles and made it publicly accessible.[13] We name the augmented dataset Twitter15 in this article. Table 1 shows some basic statistics of the two datasets.

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

4.2 User Feature Study

We conducted a comprehensive user feature study to show how the user characteristics of fake news spreaders distribute significantly differently from those of the general user population on social media, as well as how user characteristics can be utilized to detect fake news spreaders. The results of our user feature study suggest that user characteristics might be utilized to detect fake news.

4.2.1 Social Media Terminologies. We first briefly introduce some basic terminologies used in the context of social media, which will be used in the definition of several user characteristics:

- *User*: A user refers to a person or a computer program that registers on a social media platform.
- *Follower*: A follower of a user refers to another user who follows the concerned user and will automatically receive his or her posts.
- *Friend*: A friend of a user refers to another user who is followed by the concerned user.
- *Post*: A post refers to a social media object posted by a user, e.g., a text block, a photo, or a video.
- *Retweet*: Retweet refers to the action of reposting or forwarding a message posted by another user.
- *User characteristics*: User characteristics refer to a series of features/attributes that describe a user, e.g., the number of followers or the number of friends.
- *Status*: A status refers to a social media post plus the user characteristics of the user who posted the post observed at the time when the post was generated.
- *Source user*: The source user of a news article refers to the user who initially posted this news article on social media.
- *Spreader*: The spreaders of a news article refer to those users who retweeted this news article.

4.2.2 User Characteristics. Since the Weibo16 dataset already includeed user characteristics, we directly adopted a full list of them to construct user features. Those features include username length, screenname length, personal description length, followers count, friends count, listed count, attitudes count, favorites count, statuses count, registration age, is account verified, is GEO enabled, and gender. We also extracted a full list of user characteristics from Twitter user profiles, and most of them also appeared 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, 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 applied log scale on several numerical features entitled with "X counts," since those features have a near log-normal distribution. Registration age is measured in months 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 a character's level. Boolean features like is account verified are directly transformed to 0 or 1. We normalized all of the user features into the range of $[0, 1]$ using the Z-score. Tables 2 and 3 show the distribution of user characteristics in the Twitter15 and Weibo16 datasets, respectively.

4.2.3 User Categorization. To study whether user characteristics can reflect users' tendency to spread fake news, we first divide all social media users into the following six groups: (i) *source users* are users who initially posted news articles on social media, (ii) *fraudulent source users* are source users who have initially posted one or more fake news articles, (iii) *legitimate source users* are

Table 2. 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

source users who have never posted any fake news articles, (iv) *retweeters* are users who retweeted news articles on social media, (v) *fraudulent retweeters* are retweeters who have retweeted one or more fake news articles, and (vi) *legitimate retweeters* are retweeters who have never retweeted any fake news articles. Table 4 shows the population of the six user groups in the two experimental datasets.

4.2.4 Hypothesis Testing. We conducted hypothesis testing to show whether there is a significant difference between the distribution of each of the user characteristics among fake news spreaders (including fraudulent source users and fraudulent retweeters) and those among the general 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)/fake news ignorants (legitimate source users and retweeters) and that of the entire user population. The Z-score is calculated using the following formula:

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

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; and n is the sample size. A Z-score greater than 1.96 or less than −1.96 (two tailed Z-test critical threshold based on the significant level 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 that for the entire user population.

Table 3. 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

Table 4. Population of the Six User Groups

	Twitter15	Weibo16
# Source user	277	2,309
# Fraudulent source user	232	1,809
# 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

Table 5. Results of the *Z*-Test (Twitter15, Source Users)

Feature	All		Fraudulent		Legitimate	
	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	1,846.82	869.99	1,646.61	−3.50	2,879.01	7.95

Tables 5 through 8 show the results of *Z*-tests. From these tables, we can see 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 of the datasets.

For user features carrying binary values, we conducted chi-square goodness of fit tests. For any user feature, the null hypothesis is that there is no significant difference between the distribution of

Table 6. Results of the *Z*-Test (Twitter15, Retweeters)

Feature	All		Fraudulent		Legitimate	
	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	1,287.63	775.10	1,111.68	−64.62	1,394.26	50.38

Table 7. Results of the *Z*-Test (Weibo16, Source Users)

Feature	All		Fraudulent		Legitimate	
	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	1,005.78	13.84

this user feature among fake news spreaders (or ignorants) and that for the entire user population. The χ^2 score is calculated using the following formula:

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

where E_i, O_i are the expected counts and observed counts of users in a 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 the significance level of 0.05 and degree of freedom of 1) 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 9 through 12 show the results of the chi-square goodness of fit tests. From these tables, we can see that several binary user features distribute significantly differently across fraudulent source users and the entire source user population, as well as across the legitimate source users and the entire source user population. However, most of the binary user features distribute significantly differently across fraudulent retweeters and the entire retweeter population, as well as across the legitimate retweeters and the entire retweeter population.

From the results of our hypothesis testing, we can see that most user features distribute significantly differently across fake news spreaders and the general user population, as well as across

Table 8. Results of the Z-Test (Weibo16, Retweeters)

Feature	All		Fraudulent		Legitimate	
	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

Table 9. Results of the Chi-Square Goodness of Fit Test (Twitter15, Source Users)

Feature	All		Fraudulent			Legitimate		
	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

Table 10. Results of the Chi-Square Goodness of Fit Test (Twitter15, Retweeters)

Feature	All		Fraudulent			Legitimate		
	O_1	O_2	O_1	O_2	χ^2	O_1	O_2	χ^2
Has location info	151,366	64,097	58,216	23,086	71.22	93,150	41,014	43.32
Has personal URL	62,303	153,160	24,783	56,519	97.09	37,520	96,644	58.91
Are tweets protected	12,244	203,219	4,882	76,420	15.74	7,362	126,802	9.55
Is account verified	2,907	212,556	1,049	80,253	2.12	1,858	132,306	1.28
Is GEO enabled	121,878	93,585	48,062	33,240	215.13	73,816	60,348	130.58
Is language English	189,132	26,331	73,024	8,278	315.05	116,108	18,056	191.52
Is Contributors enabled	0	215,463	0	81,302	NA	0	134,164	NA
Has profile background tile	61,692	163,771	29,169	52,133	2,088.46	32,523	101,641	1,265.95
Has profile background image	184,857	30,606	71,462	9,840	294.68	113,395	20,769	179.11
Has default profile	81,423	134,040	24,026	57,276	2,347.11	57,397	76,767	1,421.84
Has default profile image	3,509	211,954	820	80,482	195.07	2689	131,475	118.19

Table 11. Results of the Chi-Square Goodness of Fit Test (Weibo16, Source Users)

Feature	All		Fraudulent			Legitimate		
	O_1	O_2	O_1	O_2	χ^2	O_1	O_2	χ^2
Is account verified	999	1,310	687	1,152	26.15	312	158	102.32
Is GEO enabled	1,601	708	1,304	535	0.45	297	173	8.35
Gender	828	1,481	635	1,204	0.32	193	277	5.53
Has location info	2,309	0	1,839	0	0	470	0	NA

Table 12. Results of the Chi-Square Goodness of Fit Test (Weibo16, Retweeters)

Feature	All		Fraudulent			Legitimate		
	O_1	O_2	O_1	O_2	χ^2	O_1	O_2	χ^2
Is account verified	101,680	2,716,322	60,559	1,561,865	72.18	41,121	1,154,457	97.95
Is GEO enabled	2,554,957	263,045	1,465,097	157,327	252.02	1,089,860	105,718	342.007
Gender	1,535,478	1,282,524	786,947	835,477	23,425.52	748,531	447,047	31,788.92
Has location info	2,818,002	0	1,622,424	0	NA	1,195,578	0	NA

Table 13. 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

fake news ignorants and the general user population. These results indicate that whether a social media user is a fake news spreader or fake news ignorant can be reflected from his or her user characteristics. We also built a simple machine-learning model to predict whether a user is a fake news spreader based on his or her user characteristics.

4.2.5 A Simple Machine-Learning Model to Predict a User's Tendency to Spread Fake News. Since we found that many 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 or her user characteristics. The proposed model can be formulated as follows:

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

where $\sigma(\cdot)$ is the sigmoid activation function, $\text{Relu}(\cdot)$ is the ReLU activation function, $\mathbf{W}_1, \mathbf{b}_1$ are the weights and bias of the feature input layer, and $\mathbf{W}_2, \mathbf{b}_2$ are the weights and bias of the hidden layer. The parameters are optimized to minimize training loss.

Table 13 shows the performance of the proposed user classification model. From this table, we can see that our user classification model can predict whether a user is likely to be a fake news spreader with high accuracy, except for retweeters in the Twitter15 dataset. We believe that this is because the Twitter15 dataset does not include a large number of retweeters. Compared to 2,818,002 retweeters in the Weibo16 dataset, Twitter15 only includes 215,691 retweeters. Nonethe-

less, we believe that by combining the user characteristics of a group of retweeters, we can detect fake news with higher accuracy than simply using those of individual fake news spreaders.

The results of our hypothesis testing and the performance of our user classification model indicate the usefulness of user characteristics in detecting fake news.

4.3 Baseline Approaches

After proving that user characteristics are highly useful in fake news detection, we compared our proposed fake news early detection model (FNED) to a series of baseline models discussed in Section 2, including the following:

- *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* [Liu and 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 did not implement the bottom-up version, because its performance is lower than the top-down one.

4.4 Experimental Setup

We implemented the proposed model using *Keras*,[14] which is a Python wrapper of TensorFlow.[15] When pre-processing the text responses, English characters in the Twitter15 dataset were tokenized using the NLTK toolkit,[16] and Chinese characters in the Weibo16 dataset were tokenized using an open source Chinese tokenizer.[17] The model was trained and tested using fivefold 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 1,000 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 14 presents a list of hyper-parameters of our proposed FNED model, as well as their experimental ranges. After configuring the model, we formally performed fivefold cross validation for 50 rounds and reported the average performance metrics yielded on the testing set as the evaluation results. We adopted standard effectiveness metrics, including accuracy, precision, recall, and $F1$-score, to evaluate all of the models. 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

[14] https://keras.io/.
[15] https://www.tensorflow.org/.
[16] https://www.nltk.org/.
[17] https://www.npmjs.com/package/chinese-tokenizer.

Table 14. Hyper-Parameters of the Proposed FNED Model

Hyper-parameter	Value	Experimental Range
Dimension of the text feature d_1	2^7	2^5–2^{10}
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
Number of weak classifiers per iteration k	10	1–50
Number of machine-labeled fake news samples produced per iteration n	5	1–20

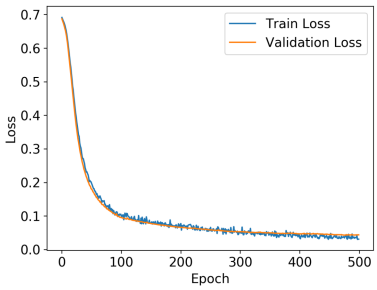


Fig. 7. Learning curve on Twitter15.

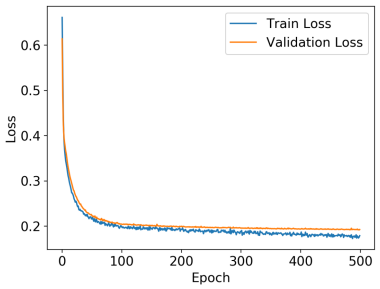


Fig. 8. Learning curve on Weibo16.

before the detection deadline as the model parameter k. Zero-padding[18] was applied to handle news articles that had 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 were classified as fake news by the strong classifier with the highest confidence score to the positive labeled 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 simulate real-world scenarios.

4.5 Results

4.5.1 Training Performance. Figures 7 and 8 show the learning curves of the proposed model on the two experimental datasets at a random round of cross validation. 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.

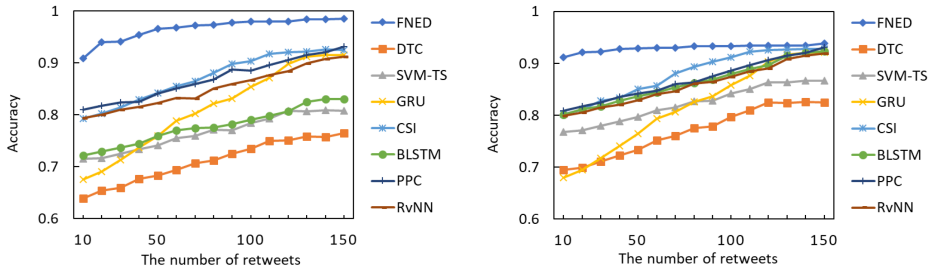
4.5.2 Comparison of Overall Performance. Through our experiments, we found that our detection model's performance peaked after observing more than 150 retweets. Later, in Figure 11, we can see that it requires roughly 1 hour to observe 150 retweets after a news story starts to spread in the two experimental datasets. We first compared our method with the baselines when the detection deadline was the first 150 retweets, i.e., $k = 150$, to get an overall image on their effectiveness at this moment. Table 15 shows the comparison results. From Table 15, we can see that our proposed FNED model outperforms the baseline models in terms of each evaluation metric, especially in the recall of fake news.

4.5.3 Comparison of Early Detection Performance. Figures 9 and 10 show the comparison of early detection performance on the two experimental datasets when the detection deadline is

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

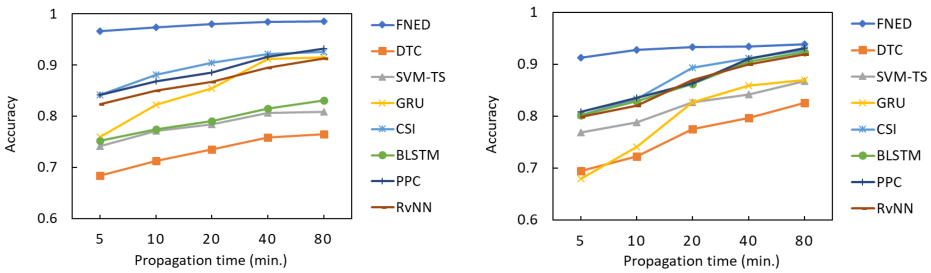
Table 15. Comparison of Overall Performance When $k = 150$

Approach	Twitter15				Weibo16			
	Accuracy	Precision	Recall	$F1$-Score	Accuracy	Precision	Recall	$F1$-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**



(a) Twitter15 (b) Weibo16

Fig. 9. Early detection performance comparison when detection deadline is measured by the number of retweeters.



(a) Twitter15 (b) Weibo16

Fig. 10. Early detection performance comparison when detection deadline is measured by the propagation time.

measured by the number of retweets and the propagation time. Figure 11 shows the average propagation speed of news articles on social media calculated based on our two experimental datasets. From these figures, we can see that our proposed model outperforms the baselines significantly in terms of early detection accuracy. Moreover, this performance difference is more significant when the detection deadline is shorter. In addition, the early detection performances based on different detection deadlines are consistent according to the average propagation speed of news articles on social media.

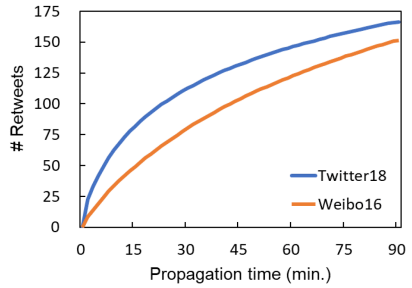


Fig. 11. The average propagation speed of news articles on social media.

Table 16. Comparison of Optimal Performance of the Reduced Models and the Full FNED 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**

4.5.4 Ablation Study. We also evaluated several simplified variations of our proposed model, each of which has one key component removed; the purpose of this ablation study was to investigate the impact of each key component on the performance. The following is a list of reduced internal models:

- *FNED-UF*: User features were not included. Only text features were used to model crowd responses.
- *FNED-TF*: Text features extracted from response text were not included. Only user features were used to model crowd responses.
- *FNED-PAAM*: Position-aware attention mechanism was removed. All crowd responses were treated identically.
- *FNED-MRMP*: Multi-region mean pooling was replaced with the basic global average pooling.
- *FNED*: The full model.

Table 16 shows the comparison of the optimal performance of the reduced models and the full FNED model. From the results, we can see that when one key component was removed, our proposed model's performance would drop. Among the four key components, user features affected the detection accuracy most significantly, whereas the text feature affected it least significantly.

4.6 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 12 and 13 show the results. From these figures, we can see that when the class distribution is more balanced and more positive labeled news samples are available, the detection accuracy of our models and that of the baselines increases. Among all of the models, our proposed model still performs the best. Compared with Table 15, which shows the optimal detection performances, we can see that when

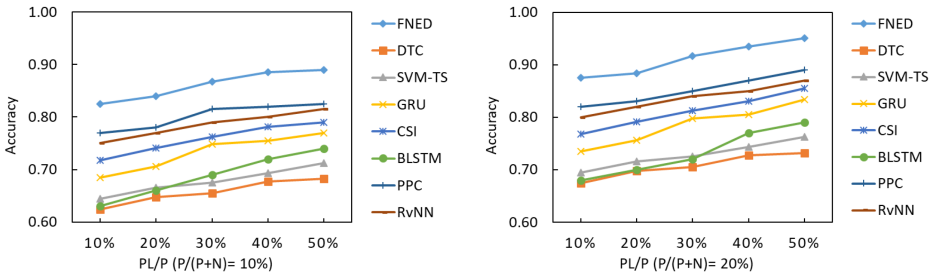


Fig. 12. Performance of PU-Learning on the Twitter15 dataset.

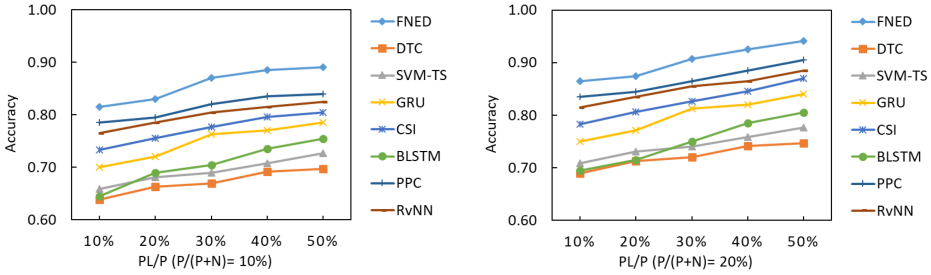


Fig. 13. Performance of PU-Learning on the Weibo16 dataset.

Table 17. Examples of Incorrectly and Correctly Classified News

Category	News Title
False Negative	No baes allowed: a Chick-fil-A manager bans 22 hip words.
	Jamie Dornan leaving Fifty Shades of Grey?! Yeah, not so much.
	French police commissioner commits suicide after meeting CharlieHebdo bereaved.
	Banksy account joins cartoonists support for CharlieHebdo, JeSuisCharlie.
	Mexico: Iguala mass grave bodies.
	Darren Wilson was paid $500,000 for his ABC interview.
	Some iPhone 6 Plus Owners Accidentally Bending Their iPhones in Pockets.
	Obama says being a stay-at-home mom'not a choice we want Americans to make.
	Someone spray painted a penis on a 1.9m Bugatti Veyron.
	The first General killed in a combat zone since Vietnam.
	Betty White Is Victim of Death Hoax After Fans Misread Satirical Story.
True Positive	Macklemore joined ISIS.
	Vladimir Putin's Motorcade As Seen From The Air.

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 were used for training our model. Thus, it proves our model's effectiveness under PU-Learning settings.

4.7 Qualitative Evaluation

We qualitatively examined all 11 false negatives and some true positives, produced by our model from the Twitter dataset, which are listed in Table 17. As described in Section 5, we intend for our

framework to be implemented by social media platforms, which will ultimately review and take actions on machine-labeled "positive" fake news articles. Under such a scenario, minimizing false negatives to limit their social harm is a top priority. We investigated the retweets of the 11 false negatives, trying to ascertain how many of them were missed by our model but were identified by users as fake. We found that most users only retweeted them without any comment. For those who did post comments, most of them supported the fake news or did not explicitly express any judgment. There is only one exception out of 11 cases—the fake news story "Banksy account joins cartoonists support for CharlieHebdo, JeSuisCharlie" had a user comment: "It's a fan account—Banksy isn't on Twitter." This user's comment implies that he or she was able to discern this new story based upon relevant knowledge that is beyond the scope of the news event itself. Most social media users would not possess enough knowledge beyond the scope of each news event to judge whether it is fake news. Currently, no machine-learning model, including ours, is able to incorporate knowledge from open-ended domains, e.g., in this case, whether Banksy has a Twitter account, for early detection of fake news. Thus, this provides new directions for future fake news detection research.

Regarding true positives, there are too many of them to review individually. We selectively reviewed some true positives with posted user comments. From their retweets, we found that some of them had comments indicating that users believed them to be true. For instance, the fake news "Macklemore joined ISIS" was retweeted with comments like "It's real," "Guys. The evidence is all there. Wake up, people!," and "Wow, disgusting." With such user comments, our model can still effectively detect these fake news stories. Another interesting finding is that our model can detect fake news that consists of a photo only and without any text content, e.g., the news story "Vladimir Putin's Motorcade As Seen From The Air." It demonstrates that our model is truly content independent and suggests that it can potentially be utilized to detect Deepfake [Blitz 2018].

4.8 Discussion on the Results

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.

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–based user classification model to predict whether a social media user is likely to spread fake news based on his or her user characteristics. The model's prediction accuracy is generally high except for the retweeters on the Twitter15 dataset, which only includes a very small number of retweeters. Although the performances were not always high in all four datasets, they proved that user characteristics were useful in predicting a user's likelihood of spreading fake news. We also believe that its performance can be further improved given larger datasets. Based on these findings, 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 response generated by another user who has spread fake news pieces might give us an entirely different clue about whether the concerned news is fake. 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 [Chen et al. 2017; Guo et al. 2018; Ruchansky et al. 2017] incorporate user features, they treat them

independently from the text responses. Compared with the baselines, our proposed model fully utilizes the information encoded in status-sensitive crowd responses to detect fake news. That is one of the reasons why our model outperforms the baselines.

Early detection of fake news is of critical importance. Although many existing approaches have decent performance 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 network 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 model GRU [Ma et al. 2016], which adopts RNNs 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, which 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: the 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 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, because a specific response generated by a specific user at a 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 that 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 accomplish this goal. The window sizes are optimized through parameter tuning. 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. However, our proposed multi-region mean pooling does not suffer from this problem, because the first 10 feature vectors learned by CNN will be averaged separately from the later 40 vectors. Our ablation study proves the effectiveness of our proposed 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 afterward. 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.

5 CONCLUSION AND DISCUSSION

In this article, we proposed a novel deep learning framework for fake news early detection. We first conducted a comprehensive statistical analysis of user characteristics and found that user characteristics distribute significantly differently between fake news spreaders and normal users, and user characteristics can be used to predict a user's tendency to spread fake news. Based on these findings, we proposed a deep neural network model named *FNED*, which takes status-sensitive crowd response as the input, CNN as the classification module, and incorporates two novel mechanisms, i.e., the position-aware attention mechanism and multi-region mean pooling, to further facilitate early detection. FNED combines multiple deep learning mechanisms and their extensions to solve the fake news early detection problem. Experimental results demonstrated that our FNED model significantly outperforms the baselines regarding effective early detection. Our ablation study demonstrates that incorporating user characteristics contributes most to our model's effectiveness on early detection. We also demonstrated that PU-Learning can be utilized to improve fake news early detection given unlabeled and imbalanced data.

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 label potential fake news articles automatically. Then, the labeled articles can be sent to social media administrators who will decide how to handle them. Compared with existing detection approaches, our approach has three significant advantages regarding business practices. The first advantage is that it has broader applicability: our proposed fake news early detection framework 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. The second advantage is that it has higher 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, because by then, fake news has already been widely spread. Compared with existing detection approaches, our proposed detection framework is significantly more effective with regard to early detection mainly because of the following reasons. The first reason is that we extract useful features from user characteristics of news spreaders to differentiate fake news from true news. User characteristics of news spreaders are readily available at the early stage of news propagation compared with other features utilized by existing approaches like response text or propagation network. The second reason is that 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 position-aware attention mechanism and multi-region mean pooling. These mechanisms also provide novel solutions to model sequential data where time and ranking positions are important in deep learning. The third advantage is that it has higher 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 learning models' structure is also not very complex. It does not require long training time.

Our proposed models can be trained offline and run in real time for early detection. Training data only needs to be updated periodically.

One limitation of our study is that the current publicly accessible experimental datasets we adopted are small and a small portion (<4%) of user profiles are no longer available for us to crawl. However, this limitation only exists in our experimental scenarios. If social media administrators decide to implement our model on their social media platform using full data, they will not have the preceding issue.

One future direction of our study is to extend the modeling of user status. First, we can incorporate user connections in a social network and users' historical activity data. User connections information can reflect the social group a user belongs to and how the social network structure facilitates fake news propagation. Second, we can incorporate user historical activity data to model the user status better, as a user's tendency to spread fake news might change over time.

REFERENCES

Sadia Afroz, Michael Brennan, and Rachel Greenstadt. 2012. Detecting hoaxes, frauds, and deception in writing style online. In *Proceedings of the IEEE Symposium on Security and Privacy (SP'12)*. IEEE, Los Alamitos, CA, 461–475.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv:1409.0473.

Marc Jonathan Blitz. 2018. Lies, line drawing, and deep fake news. *Oklahoma Law Review* 71 (2018), 59.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on Twitter. In *Proceedings of the 20th International Conference on World Wide Web*. ACM, New York, NY, 675–684.

Tong Chen, Lin Wu, Xue Li, Jun Zhang, Hongzhi Yin, and Yang Wang. 2017. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. arXiv:1704.05973.

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

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

Joydip Dhar, Ankur Jain, and Vijay K. Gupta. 2016. A mathematical model of news propagation on online social network and a control strategy for rumor spreading. *Social Network Analysis and Mining* 6, 1 (2016), 57.

Rim El Ballouli, Wassim El-Hajj, Ahmad Ghandour, Shady Elbassuoni, Hazem Hajj, and Khaled Shaban. 2017. CAT: Credibility analysis of Arabic content on Twitter. In *Proceedings of the 3rd Arabic Natural Language Processing Workshop (WANLP'17) Co-Located with EACL 2017)*. 62.

Boris Galitsky. 2015. Detecting rumor and disinformation by web mining. In *Proceedings of the 2015 AAAI Spring Symposium Series*.

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

Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. 2014. TweetCred: Real-time credibility assessment of content on Twitter. In *Proceedings of the International Conference on Social Informatics*. 228–243.

Manish Gupta, Peixiang Zhao, and Jiawei Han. 2012. Evaluating event credibility on Twitter. In *Proceedings of the 2012 SIAM International Conference on Data Mining*. 153–164.

Cho-Jui Hsieh, Nagarajan Natarajan, and Inderjit S. Dhillon. 2015. PU Learning for matrix completion. In *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*. 2445–2453.

Xia Hu, Jiliang Tang, Yanchao Zhang, and Huan Liu. 2013. Social spammer detection in microblogging. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*, Vol. 13. 2633–2639.

Alankar Jain, Vivek Borkar, and Dinesh Garg. 2016a. Fast rumor source identification via random walks. *Social Network Analysis and Mining* 6, 1 (2016), 62.

Suchita Jain, Vanya Sharma, and Rishabh Kaushal. 2016b. Towards automated real-time detection of misinformation on Twitter. In *Proceedings of the International Conference on Advances in Computing, Communications, and Informatics (ICACCI'16)*. IEEE, Los Alamitos, CA, 2015–2020.

F. Jin, E. Dougherty, P. Saraf, Y. Cao, and N. Ramakrishnan. 2013. Epidemiological modeling of news and rumors on Twitter. In *Proceedings of the 7th Workshop on Social Network Mining and Analysis*. ACM, New York, NY, Article 8, 9 pages.

Zhiwei Jin, Juan Cao, Han Guo, Yongdong Zhang, and Jiebo Luo. 2017a. Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In *Proceedings of the 2017 ACM Multimedia Conference*. ACM, New York, NY, 795–816.

Zhiwei Jin, Juan Cao, Yu-Gang Jiang, and Yongdong Zhang. 2014. News credibility evaluation on microblog with a hierarchical propagation model. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'14)*. IEEE, Los Alamitos, CA, 230–239.

Zhiwei Jin, Juan Cao, Yongdong Zhang, Jianshe Zhou, and Qi Tian. 2017b. Novel visual and statistical image features for microblogs news verification. *IEEE Transactions on Multimedia* 19, 3 (2017), 598–608.

Sejeong Kwon, Meeyoung Cha, and Kyomin Jung. 2017. Rumor detection over varying time windows. *PLoS One* 12, 1 (2017), e0168344.

Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. 2013. Prominent features of rumor propagation in online social media. In *Proceedings of the IEEE 13th International Conference on Data Mining (ICDM'13)*. IEEE, Los Alamitos, CA, 1103–1108.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436.

Wee Sun Lee and Bing Liu. 2003. Learning with positive and unlabeled examples using weighted logistic regression. In *Proceedings of the 20th International Conference on Machine Learning (ICML'03)*, Vol. 3. 448–455.

Chaoliang Li and Shigang Liu. 2017. A comparative study of the class imbalance problem in Twitter spam detection. *Concurrency and Computation: Practice and Experience* 30, 5 (2017), e4281.

Xiaoli Li and Bing Liu. 2003. Learning to classify texts using positive and unlabeled data. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, Vol. 3. 587–592.

Xiao-Li Li and Bing Liu. 2005. Learning from positive and unlabeled examples with different data distributions. In *Proceedings of the 16th European Conference on Machine Learning (ECML'05)*. 218–229.

Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. 2003. Building text classifiers using positive and unlabeled examples. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM'03)*. IEEE, Los Alamitos, CA, 179–186.

Bing Liu, Wee Sun Lee, Philip S. Yu, and Xiaoli Li. 2002. Partially supervised classification of text documents. In *Proceedings of the 19th International Conference on Machine Learning (ICML'02)*, Vol. 2. 387–394.

Shigang Liu, Yu Wang, Jun Zhang, Chao Chen, and Yang Xiang. 2017b. Addressing the class imbalance problem in Twitter spam detection using ensemble learning. *Computers & Security* 69 (2017), 35–49.

Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. 2015a. Real-time rumor debunking on Twitter. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*. ACM, New York, NY, 1867–1870.

Yahui Liu, Xiaolong Jin, Huawei Shen, and Xueqi Cheng. 2017a. Do rumors diffuse differently from non-rumors? A systematically empirical analysis in Sina Weibo for rumor identification. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. 407–420.

Yang Liu and Yi-Fang Brook Wu. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

Yang Liu and Songhua Xu. 2016. Detecting rumors through modeling information propagation networks in a social media environment. *IEEE Transactions on Computational Social Systems* 3, 2 (2016), 46–62.

Yang Liu, Songhua Xu, and Georgia Tourassi. 2015b. Detecting rumors through modeling information propagation networks in a social media environment. In *Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*. 121–130.

Michal Lukasik, Kalina Bontcheva, Trevor Cohn, Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2019. Gaussian processes for rumour stance classification in social media. *ACM Transactions on Information Systems* 37, 2 (2019), 20.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16)*. 3818–3824.

Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. 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*. ACM, New York, NY, 1751–1754.

Jing Ma, Wei Gao, and Kam-Fai Wong. 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. 708–717.

Jing Ma, Wei Gao, and Kam-Fai Wong. 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)*. 1980–1989.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2019. Detect rumors on Twitter by promoting information campaigns with generative adversarial learning. In *Proceedings of the World Wide Web Conference (WWW'19)*. 3049–3055.

Benjamin Markines, Ciro Cattuto, and Filippo Menczer. 2009. Social spam detection. In *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*. ACM, New York, NY, 41–48.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. 3111–3119.

Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. 2014. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*. 2204–2212.

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

Vahed Qazvinian, Emily Rosengren, Dragomir R. Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 1589–1599.

Victoria Rubin, Niall Conroy, Yimin Chen, and Sarah Cornwell. 2016. Fake news or truth? Using satirical cues to detect potentially misleading news. In *Proceedings of the 2nd Workshop on Computational Approaches to Deception Detection*. 7–17.

Victoria L. Rubin. 2017. Deception detection and rumor debunking for social media. In *The SAGE Handbook of Social Media Research Methods,* L. Sloan and A. Quan-Haase (Eds.). SAGE, London, UK, 342.

Victoria L. Rubin, Yimin Chen, and Niall J. Conroy. 2015. Deception detection for news: Three types of fakes. *Proceedings of the Association for Information Science and Technology* 52, 1 (2015), 1–4.

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

Justin Sampson, Fred Morstatter, Liang Wu, and Huan Liu. 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*. ACM, New York, NY, 2377–2382.

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

Kai Shu, Suhang Wang, and Huan Liu. 2018. Understanding user profiles on social media for fake news detection. In *Proceedings of the 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR'18)*. IEEE, Los Alamitos, CA, 430–435.

Sam Spencer and R. Srikant. 2016. Maximum likelihood rumor source detection in a star network. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'16)*. IEEE, Los Alamitos, CA, 2199–2203.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

S. Y. Sun, H. Y. Liu, J. He, and X. Y. Du. 2013. Detecting event rumors on Sina Weibo automatically. In *Web Technologies and Applications*. Springer, 120–131.

Wenbing Tang, Zuohua Ding, and Mengchu Zhou. 2019. A spammer identification method for class imbalanced Weibo datasets. *IEEE Access* 7 (2019), 29193–29201.

Laura Tolosi, Andrey Tagarev, and Georgi Georgiev. 2016. An analysis of event-agnostic features for rumour classification in Twitter. In *Proceedings of the 10th International AAAI Conference on Web and Social Media*.

Soroush Vosoughi. 2015. *Automatic Detection and Verification of Rumors on Twitter*. Ph.D. Dissertation. Massachusetts Institute of Technology, Cambridge, MA.

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

De Wang, Danesh Irani, and Calton Pu. 2011. A social-spam detection framework. In *Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-Abuse, and Spam Conference*. ACM, New York, NY, 46–54.

Shihan Wang and Takao Terano. 2015. Detecting rumor patterns in streaming social media. In *Proceedings of the IEEE International Conference on Big Data (Big Data'15)*. IEEE, Los Alamitos, CA, 2709–2715.

Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. EANN: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 849–857.

Ke Wu, Song Yang, and Kenny Q. Zhu. 2015. False rumors detection on Sina Weibo by propagation structures. In *Proceedings of the 31st IEEE International Conference on Data Engineering*.

Liang Wu, Jundong Li, Xia Hu, and Huan Liu. 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*. 99–107.

Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. 2012. Automatic detection of rumor on Sina Weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*. ACM, New York, NY, Article 13, 7 pages.

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

YeKang Yang, Kai Niu, and ZhiQiang He. 2015a. Exploiting the topology property of social network for rumor detection. In *Proceedings of the 12th International Joint Conference on Computer Science and Software Engineering (JCSSE'15)*. IEEE, Los Alamitos, CA, 41–46.

Zhifan Yang, Chao Wang, Fan Zhang, Ying Zhang, and Haiwei Zhang. 2015b. Emerging rumor identification for social media with hot topic detection. In *Proceedings of the 2015 12th Web Information System and Application Conference (WISA'15)*. IEEE, Los Alamitos, CA, 53–58.

Hwanjo Yu, Jiawei Han, and Kevin Chen-Chuan Chang. 2002. PEBL: Positive example based learning for web page classification using SVM. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 239–248.

Matthew D. Zeiler. 2012. ADADELTA: An adaptive learning rate method. arXiv:1212.5701.

Huiling Zhang, Md Abdul Alim, Xiang Li, My T. Thai, and Hien T. Nguyen. 2016. Misinformation in online social networks: Detect them all with a limited budget. *ACM Transactions on Information Systems* 34, 3 (2016), 18.

Qiao Zhang, Shuiyuan Zhang, Jian Dong, Jinhua Xiong, and Xueqi Cheng. 2015. Automatic detection of rumor on social network. In *Natural Language Processing and Chinese Computing*. Springer, 113–122.

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

Liang Zheng and Chee Wei Tan. 2015. A probabilistic characterization of the rumor graph boundary in rumor source detection. In *Proceedings of the IEEE International Conference on Digital Signal Processing (DSP'15)*. IEEE, Los Alamitos, CA, 765–769.

Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2016. Learning reporting dynamics during breaking news for rumour detection in social media. arXiv:1610.07363.

Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2017. Exploiting context for rumour detection in social media. In *Proceedings of the International Conference on Social Informatics*. 109–123.

Received June 2019; revised February 2020; accepted March 2020