Fake News Detection with Deep Diffusive Network Model

Jiawei Zhang[1], Limeng Cui[2], Yanjie Fu[3], Fisher B. Gouza[1]
[1]IFM Lab, Department of Computer Science, Florida State University, FL, USA
[2]School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China
[3]Department of Computer Science, Missouri University of Science and Technology, MO, USA
jzhang@cs.fsu.edu,lmcui932@163.com,fuyan@mst.edu,fisherbgouza@gmail.com

ABSTRACT

In recent years, due to the booming development of online social networks, fake news for various commercial and political purposes has been appearing in large numbers and widespread in the online world. With deceptive words, online social network users can get infected by these online fake news easily, which has brought about tremendous effects on the offline society already. An important goal in improving the trustworthiness of information in online social networks is to identify the fake news timely. This paper aims at investigating the principles, methodologies and algorithms for detecting fake news articles, creators and subjects from online social networks and evaluating the corresponding performance. This paper addresses the challenges introduced by the unknown characteristics of fake news and diverse connections among news articles, creators and subjects. Based on a detailed data analysis, this paper introduces a novel automatic fake news credibility inference model, namely FakeDetector. Based on a set of explicit and latent features extracted from the textual information, FakeDetector builds a deep diffusive network model to learn the representations of news articles, creators and subjects simultaneously. Extensive experiments have been done on a real-world fake news dataset to compare FakeDetector with several state-of-the-art models, and the experimental results have demonstrated the effectiveness of the proposed model.

KEYWORDS

Fake News Detection; Diffusive Network; Text Mining; Data Mining

ACM Reference Format:
Jiawei Zhang[1], Limeng Cui[2], Yanjie Fu[3], Fisher B. Gouza[1]. 2018. Fake News Detection with Deep Diffusive Network Model. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Fake news denotes a type of yellow press which intentionally presents misinformation or hoaxes spreading through both traditional print news media and recent online social media. Fake news has been existing for a long time, since the "Great moon hoax" published in 1835 [1]. In recent years, due to the booming developments of online social networks, fake news for various commercial and political purposes has been appearing in large numbers and widespread in the online world. With deceptive words, online social network users can get infected by these online fake news easily, which has brought about tremendous effects on the offline society already. During the 2016 US president election, various kinds of fake news about the candidates widely spread in the online social networks, which may have a significant effect on the election results. According to a post-election statistical report [4], online social networks account for more than 41.8% of the fake news data traffic in the election, which is much greater than the data traffic shares of both traditional TV/radio/print medium and online search engines respectively. An important goal in improving the trustworthiness of information in online social networks is to identify the fake news timely, which will be the main tasks studied in this paper.

Fake news has significant differences compared with traditional suspicious information, like spams [3, 19, 66, 67], in various aspects: (1) *impact on society*: spams usually exist in personal emails or specific review websites and merely have a local impact on a small number of audiences, while the impact fake news in online social networks can be tremendous due to the massive user numbers globally, which is further boosted by the extensive information sharing and propagation among these users [38, 58, 68]; (2) *audiences' initiative*: instead of receiving spam emails passively, users in online social networks may seek for, receive and share news information actively with no sense about its correctness; and (3) *identification difficulty*: via comparisons with abundant regular messages (in emails or review websites), spams are usually easier to be distinguished; meanwhile, identifying fake news with erroneous information is incredibly challenging, since it requires both tedious evidence-collecting and careful fact-checking due to the lack of other comparative news articles available.

These characteristics aforementioned of fake news pose new challenges on the detection task. Besides detecting fake news articles, identifying the fake news creators and subjects will actually be more important, which will help completely eradicate a large number of fake news from the origins in online social networks. Generally, for the the news creators, besides the articles written by them, we are also able to retrieve his/her profile information from either the social network website or external knowledge libraries, e.g., Wikipedia or government-internal database, which will provide fundamental complementary information for his/her background check. Meanwhile, for the news subjects, we can also obtain its textual descriptions or other related information, which can be used as the foundations for news subject credibility inference. From a higher-level perspective, the tasks of fake news article, creator and subject detection are highly correlated, since the articles written from a trustworthy person should have a higher credibility, while the person who frequently posting unauthentic information will have a lower credibility on the other hand. Similar correlations can also be observed between news articles and news subjects. In the following part of this paper, without clear specifications, we will use the general *fake news* term to denote the *fake news articles*, *creators* and *subjects* by default.

Problem Studied: In this paper, we propose to study the fake news detection (including the articles, creators and subjects) problem in online social networks. Based on various types of heterogeneous information sources, including both textual contents/profile/descriptions and the authorship and article-subject relationships among them, we aim at identifying fake news from the online social networks simultaneously. We formulate the fake news detection problem as a credibility inference problem, where the real ones will have a higher credibility while unauthentic ones will have a lower one instead.

The fake news detection problem is not easy to address due to the following reasons:

- *Problem Formulation*: The fake news detection problem studied in this paper is a new research problem, and a formal definition and formulation of the problem is required and necessary before studying the problem.
- *Textual Information Usage*: For the news articles, creators and subjects, a set of their textual information about their contents, profiles and descriptions can be collected from the online social media. To capture signals revealing their credibility, an effective feature extraction and learning model will be needed.
- *Heterogeneous Information Fusion*: In addition, as mentioned before, the credibility labels of news articles, creators and subjects have very strong correlations, which can be indicated by the authorship and article-subject relationships between them. An effective incorporation of such correlations in the framework learning will be helpful for more precise credibility inference results for fake news.

To resolve these challenges aforementioned, in this paper, we will introduce a new fake news detection framework, namely FakeDetector. In FakeDetector, the fake news detection problem is formulated as a credibility label inference problem, and FakeDetector aims at learning a prediction model to infer the credibility labels of news articles, creators and subjects simultaneously. FakeDetector deploys the bag-of-word and RNN models for learning the explicit and latent feature representations of news articles, creators and subjects respectively, and introduce a novel deep diffusive network model for the heterogeneous information fusion within the social networks.

The remaining paper is organized as follows. At first, we will introduce several important concepts and formulate the fake news detection problem in Section 2. Before introducing the proposed framework, we will provide a detailed analysis about fake news dataset in Section 3, which will provide useful signals for the framework building. The framework FakeDetector is introduced in Section 4, whose effectiveness will be evaluated in Section 5. Finally, we will talk about the related works in Section 6 and conclude this paper in Section 7.

2 TERMINOLOGY DEFINITION AND PROBLEM FORMULATION

In this section, we will introduce the definitions of several important concepts and provide the formulation of the studied problem.

2.1 Terminology Definition

In this paper, we will use the "*news article*" concept in referring to the posts either written or shared by users in online social media, and use the "*news creator*" concept to denote the set of users writing the news articles.

Definition 2.1. (News Articles): News articles published in online social networks can be represented as set $\mathcal{N} = \{n_1, n_2, \cdots, n_m\}$. For each news article $n_i \in \mathcal{N}$, it can be represented as a tuple $n_i = (n_i^t, n_i^c)$, where the entries denote its *textual content* and *credibility label* respectively.

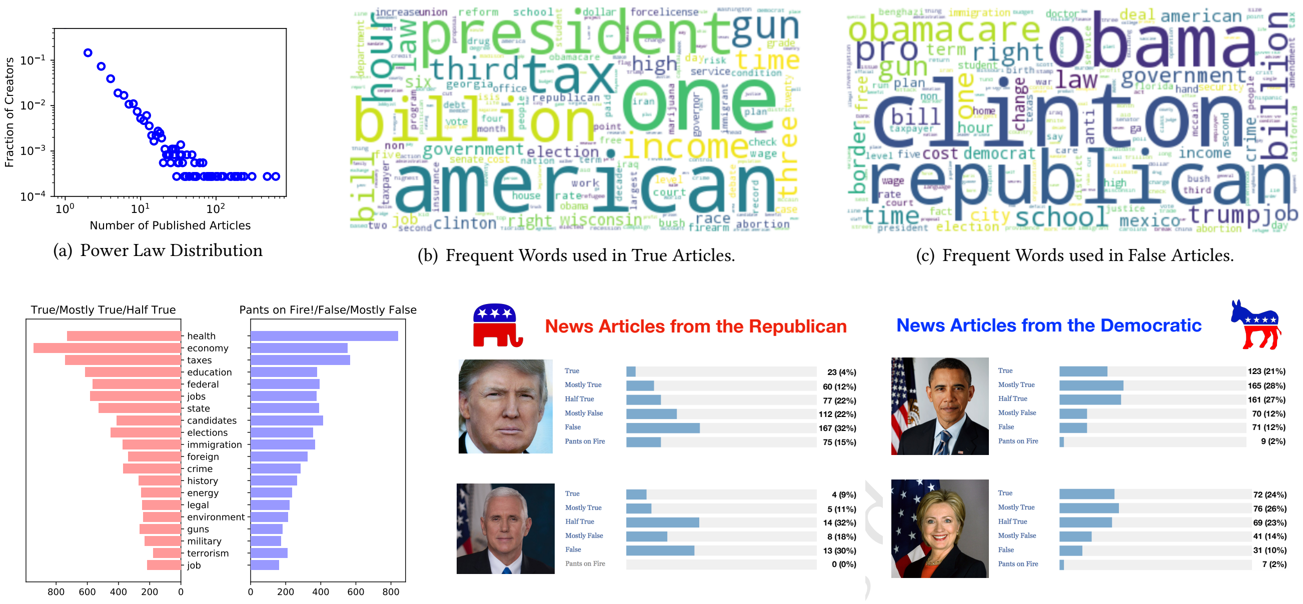
In the above definition, the news article credibility label is from set $\mathcal{Y}$, i.e., $n_i^c \in \mathcal{Y}$ for $n_i \in \mathcal{N}$. For the PolitiFact dataset to be introduced later, its label set $\mathcal{Y}$ ={True, Mostly True, Half True, Mostly False, False, Pants on Fire!} contains 6 different class labels, whose credibility ranks from high to low respectively. In addition, the news articles in online social networks are also usually about some topics, which are also called the *news subjects* in this paper. News subjects usually denote the central ideas of news articles, and they are also the main objectives of writing the news articles.

Definition 2.2. (News Subject): Formally, we can represent the set of news subjects involved in the social network as $\mathcal{S} = \{s_1, s_2, \cdots, s_k\}$. For each subject $s_i \in \mathcal{S}$, it can be represented as a tuple $s_i = (s_i^t, s_i^c)$ containing its *textual description* and *credibility label* respectively.

Definition 2.3. (News Creator): We can represent the set of news creators in the social network as $\mathcal{U} = \{u_1, u_2, \cdots, u_n\}$. To be consistent with the definition of *news articles*, we can also represent news creator $u_i \in \mathcal{U}$ as a tuple $u_i = (u_i^p, u_i^s)$, where the entries denote the *profile information* and *credibility label* of the creator respectively.

For the news article creator $u_i \in \mathcal{U}$, his/her profile information can be represented as a sequence of words describing his/her basic background. For some of the creators, we can also have his/her title representing either their jobs, political party membership, their geographical residential locations or companies they work at, e.g., "political analyst", "Democrat"/"Republican", "New York"/"Illinois" or "CNN"/"Fox". Similarly, the credibility labels of the creator u_i can also be assigned with a class label from set $\mathcal{Y}$.

Definition 2.4. (News Augmented Heterogeneous Social Networks): The online social network together with the news articles published in it can be represented as a *news augmented heterogeneous social network* (News-HSN) $G = (\mathcal{V}, \mathcal{E})$, where the node set $\mathcal{V} = \mathcal{U} \cup \mathcal{N} \cup \mathcal{S}$ covers the sets of news articles, creators and subjects, and the edge set $\mathcal{E} = \mathcal{E}_{u,n} \cup \mathcal{E}_{n,s}$ involves the authorship links between news articles and news creators, and the topic indication links between news articles and news subjects.



(a) Power Law Distribution

(b) Frequent Words used in True Articles.

(c) Frequent Words used in False Articles.

(d) Subject Credibility Distribution.

(e) News Articles from Republican.

(f) News Articles from Democratic.

Figure 1: PolitiFact Dataset Statistical Analysis.

2.2 Problem Formulation

Based on the definitions of terminologies introduced above, the fake news detection problem studied in this paper can be formally defined as follows.

Problem Formulation: Given a News-HSN $G = (\mathcal{V}, \mathcal{E})$, the fake news detection problem aims at learning an inference function $f : \mathcal{U} \cup \mathcal{N} \cup \mathcal{S} \rightarrow \mathcal{Y}$ to predict the *credibility labels* of news articles in set $\mathcal{N}$, news creators in set $\mathcal{U}$ and news subjects in set $\mathcal{S}$. In learning function f, various kinds of heterogeneous information in network G should be effectively incorporated, including both the textual content/profile/description information as well as the connections among them.

To resolve the above fake news detection problem, before introducing the proposed framework FakeDetector, we will provide an analysis about the fake news dataset first in Section 3.

3 DATASET ANALYSIS

The dataset used in this paper includes both the tweets posted by PolitiFact from its official Twitter account[1], as well as the fact-check articles written regarding these statements in the PolitiFact website[2]. In this section, we will first provide the basic statistical information about the crawled dataset, after which we will carry out a detailed analysis about the information regarding the news articles, creators and subjects respectively.

3.1 Dataset Statistical Information

PolitiFact website is operated by the Tampa Bay Times, where the reporters and editors can make fact-check regarding the statements (i.e., news articles in this paper) made by the Congress members, White House, lobbyists and other political groups (namely the "creators" in this paper). PolitiFact collects the political statements from the speech, news article report, online social media, etc., and will publish both the original statements, evaluation results, and the complete fact-check report at both PolitiFact website and via its official Twitter account. The statement evaluation results will clearly indicate the credibility rating, ranging from "True" for completely accurate statements to "Pants on Fire!" for totally false claims. In addition, PolitiFact also categorizes the statements into different groups regarding the subjects, which denote the topics that those statements are about. Based on the credibility of statements made by the creators, PolitiFact also provides the credibility evaluation for these creators and subjects as well. The crawled PolitiFact dataset can be organized as a network, involving articles, creators and subjects as the nodes, as well as the authorship link (between articles and creators) and subject indication link (between articles and subjects) as the connections. More detailed statistical information is provided as follows and in Table 1.

The number of crawled news articles is $14,055$, which are created by $3,634$ creators, and each creator has created 3.86 articles on average. These articles belong to 152 subjects respectively, and each article may belong to multiple subjects simultaneously. In the crawled dataset, the number of article-subject link is $48,756$. On average, each news article has about 3.5 associated subjects. Each news article also has a "Truth-O-Meter" rating score indicating its credibility, which takes values from {True, Mostly True, Half True, Half False, Mostly False, Pants on Fire!}.

[1] https://twitter.com/PolitiFact
[2] http://www.politifact.com

Table 1: Properties of the Heterogeneous Networks

	property	**PolitiFact Network**
# node	articles	14,055
	creators	3,634
	subjects	152
# link	creator-article	14,055
	article-subject	48,756

3.2 Dataset Detailed Analysis

In this part, we will introduce a detailed analysis of the crawled PolitiFact network dataset, which can provide necessary motivations and foundations for our proposed model to be introduced in the next section. The data analysis in this section includes 4 main parts: *creator-article publishing historical records*, *article credibility analysis with textual content*, *subject credibility analysis*, as well as *creator credibility analysis*, and the results are illustrated in Figure 1 respectively.

3.2.1 Creator-Article Publishing Historical Records. In Figure 1(a), we show the scatter plot about the distribution of the number of news articles regarding the fraction of creators who have published these numbers of articles in the dataset. According to the plot, the creator-article publishing records follow the power law distribution. For a large proportion of the creators, they have merely published less than 10 articles, and a very small number of creators have ever published more than 100 articles. Among all the creators, Barack Obama has the most articles, whose number is about 599.

3.2.2 Article Credibility Analysis with Textual Content. On the other hand, in Figures 1(b)-1(c), we illustrate the frequent word cloud of the true and false news articles, where the stop words have been removed already. Here, the true article set covers the news articles which are rated “True”, “Mostly True” or “Half True”; meanwhile, the false article set covers the news articles which are rated “Pants on Fire!”, “False” or “Mostly False”. According to the plots, from Figure 1(b), we can find some unique words in True-labeled articles which don’t appear often in Figure 1(c), like “President”, “income”, “tax” and “american”, ect.; meanwhile, from Figure 1(c), we can observe some unique words appearing often in the false articles, which include “Obama”, “republican” “Clinton”, “obamacare” and “gun”, but don’t appear frequently in the True-labeled articles. These textual words can provide important signals for distinguishing the true articles from the false ones.

3.2.3 Subject Credibility Analysis. In Figure 1(d), we provide the statistics about the top 20 subjects with the largest number of articles, where the red bar denotes the true articles belonging to these subjects and the blue bar corresponds to the false news articles instead. According to the plot, among all the 152 subjects, subject “health” covers the largest number of articles, whose number is about $1,572$. Among these articles, 731 (46.5%) of them are the true articles and 841 (53.5%) of them are false, and articles in this subject are heavily inclined towards the false group. The second largest subject is “economy” with $1,498$ articles in total, among which 946 (63.2%) are true and 552 (36.8%) are false. Different from “health”, the articles belonging to the “economy” subject are biased to be true instead. Among all the top 20 subjects, most of them are about the economic and livelihood issues, which are also the main topics that presidential candidates will debate about during the election.

3.2.4 Creator Credibility Analysis. Finally, in Figures 1(e)-1(f), we show 4 case studies regarding the creators’ credibility based on their published articles. We divide the case studies into two groups: republican vs democratic, where the representatives are “Donald Trump”, “Mike Pence”, “Barack Obama” and “Hillary Clinton” respectively. According to the plots, for the articles in the crawled dataset, most of the articles from “Donald Trump” in the dataset are evaluated to be false, which account for about 69% of all his statements. For “Mike Pence”, the ratio of true articles vs false articles is 52% : 48% instead. Meanwhile, for most of the articles in the dataset from “Barack Obama” and “Hillary Clinton” are evaluated to be true with fact check, which takes more than 76% and 73% of their total articles respectively.

We need to add a remark here, the above observations are merely limited to the crawled PolitiFact dataset only. Based on these observations, we will build a unified credibility inference model to identify the fake news articles, creators and subjects simultaneously from the network with a deep diffusive network model in the next section.

4 PROPOSED METHODS

Based on the important signals revealed in the previous data analysis, we will provide the detailed information about the FakeDetector framework in this section. Framework FakeDetector covers two main components: *representation feature learning*, and *credibility label inference*, which together will compose the deep diffusive network model FakeDetector.

4.1 Representation Feature Learning

According to the data analysis aforementioned, both the textual contents and the diverse relationships among news articles, creators and subjects can provide important information for inferring the credibility labels of fake news. In this part, we will focus on feature learning from the textual content information based on the *hybrid feature extraction unit* as shown in Figure 3(a), while the relationships will be used for building the deep diffusive model in the following subsection.

4.1.1 Explicit Feature Extraction. Based on the previous data analysis, the textual information of fake news can reveal important signals for their credibility inference. Besides some shared words used in both true and false articles (or creators/subjects), a set of frequently used words can also be extracted from the article contents, creator profiles and subject descriptions of each category respectively. Let $\mathcal{W}$ denotes the complete vocabulary set used in the PolitiFact dataset, and from $\mathcal{W}$ a set of unique words can also be extracted from articles, creator profile and subject textual information, which can be denoted as sets $\mathcal{W}_n \subset \mathcal{W}$, $\mathcal{W}_u \subset \mathcal{W}$ and $\mathcal{W}_s \subset \mathcal{W}$ respectively (of size d).

These extracted words have shown their stronger correlations with their fake/true labels. As shown in the left component of Figure 3(a), based on the pre-extracted word sets $\mathbf{W}_n$, given a news

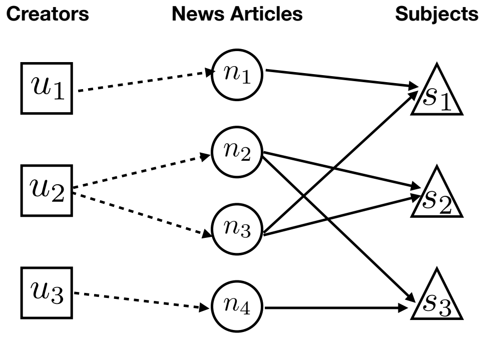


Figure 2: Relationships of Articles, Creators and Subjects.

article $n_i \in \mathcal{N}$, we can represent the extracted explicit feature vector for n_i as vector $\mathbf{x}^e_{n,i} \in \mathbb{R}^d$, where entry $x^e_{n,i}(k)$ denotes the number of appearance times of word $w_k \in \mathbf{W}_n$ in news article n_i. In a similar way, based on the extracted word set $\mathcal{W}_u$ (and $\mathcal{W}_s$), we can also represent the extracted explicit feature vectors for creator u_j as $\mathbf{x}^e_{u,j} \in \mathbb{R}^d$ (and for subject s_l as $\mathbf{x}^e_{s,l} \in \mathbb{R}^d$).

4.1.2 Latent Feature Extraction. Besides those explicitly visible words about the news article content, creator profile and subject description, there also exist some hidden signals about articles, creators and subjects, e.g., *news article content information inconsistency* and *profile/description latent patterns*, which can be effectively detected from the latent features as introduced in [32]. Based on such an intuition, in this paper, we propose to further extract a set of *latent features* for news articles, creators and subjects based on the deep recurrent neural network model.

Formally, given a news article $n_i \in \mathcal{N}$, based on its original textual contents, we can represents its content as a sequence of words represented as vectors $(\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \cdots, \mathbf{x}_{i,q})$, where q denotes the maximum length of articles (and for those with less than q words, zero-padding will be adopted). Each feature vector $\mathbf{x}_{i,k}$ corresponds to one word in the article. Based on the vocabulary set $\mathcal{W}$, it can be represented in different ways, e.g., the one-hot code representation or the binary code vector of a unique index assigned for the word. The latter representation will save the computational space cost greatly.

As shown in the right component of Figure 3(a), the latent feature extraction is based on RNN model (with the basic neuron cells), which has 3 layers (1 input layer, 1 hidden layer, and 1 fusion layer). Based on the input vectors $(\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \cdots, \mathbf{x}_{i,q})$ of the textual input string, we can represent the feature vectors at the hidden layer and the output layer as follows respectively:

$$\begin{cases} \text{\# Fusion Layer: } \mathbf{x}^l_{n,i} = \sigma(\sum_{t=1}^{q} \mathbf{W}_i \mathbf{h}_{i,t}), \\ \text{\# Hidden Layer: } \mathbf{h}_{i,t} = GRU(\mathbf{h}_{i,t-1}, \mathbf{x}_{i,t}; \mathbf{W}) \end{cases}$$

where GRU (Gated Recurrent Unit) is used as the unit model in the hidden layer and the $\mathbf{W}$ matrices denote the variables of the model to be learned.

Based on a component with a similar architecture, we can extract the latent feature vector for news creator $u_j \in \mathcal{U}$ (and subject $s_l \in \mathcal{S}$) as well, which can be denoted as vector $\mathbf{x}^l_{u,j}$ (and $\mathbf{x}^l_{s,l}$). By appending the explicit and latent feature vectors together, we can formally represent the extracted feature representations of news articles, creators and subjects as $\mathbf{x}_{n,i} = \left[(\mathbf{x}^e_{n,i})^\top, (\mathbf{x}^l_{n,i})^\top\right]^\top$, $\mathbf{x}_{u,j} = \left[(\mathbf{x}^e_{u,j})^\top, (\mathbf{x}^l_{u,j})^\top\right]^\top$ and $\mathbf{x}_{s,l} = \left[(\mathbf{x}^e_{s,l})^\top, (\mathbf{x}^l_{s,l})^\top\right]^\top$ respectively, which will be fed as the inputs for the deep diffusive unit model to be introduced in the next subsection.

4.2 Deep Diffusive Unit Model

According to the analysis in Section 3, the credibility of news articles are highly correlated with their subjects and creators. The relationships among news articles, creators and subjects are illustrated with an example in Figure 2. For each creator, they can write multiple news articles, and each news article has only one creator. Each news article can belong to multiple subjects, and each subject can also have multiple news articles taking it as their main topics. To model the correlation among news articles, creators and subjects, we will introduce the deep diffusive network model as follow.

The overall architecture of FakeDetector corresponding to the case study shown in Figure 2 is provided in Figure 3(c). Besides the HFLU feature learning unit model, FakeDetector also uses a *gated diffusive unit* (GDU) model for effective relationship modeling among news articles, creators and subjects, whose structure is illustrated in Figure 3(b). Formally, the GDU model accepts multiple inputs from different sources simultaneously, i.e., $\mathbf{x}_i$, $\mathbf{z}_i$ and $\mathbf{t}_i$, and outputs its learned hidden state $\mathbf{h}_i$ to the output layer and other unit models in the diffusive network architecture.

Here, let's take news articles as an example. Formally, among all the inputs of the GDU model, $\mathbf{x}_i$ denotes the extracted feature vector from HFLU for news articles, $\mathbf{z}_i$ represents the input from other GDUs corresponding to subjects, and $\mathbf{t}_i$ represents the input from other GDUs about creators. For the inputs from the subjects, GDU has a gate called the "forget gate", which may update some content of $\mathbf{z}_i$ to forget. The forget gate is important, since in the real world, different news articles may focus on different aspects about the subjects and "forgetting" part of the input from the subjects is necessary in modeling. Formally, we can represent the "forget gate" together with the updated input as

$$\tilde{\mathbf{z}}i = \mathbf{f}_i \otimes \mathbf{z}_i, \text{ where } \mathbf{f}_i = \sigma\left(\mathbf{W}_f \left[\mathbf{x}_i^\top, \mathbf{z}_i^\top, \mathbf{t}_i^\top\right]^\top\right).$$

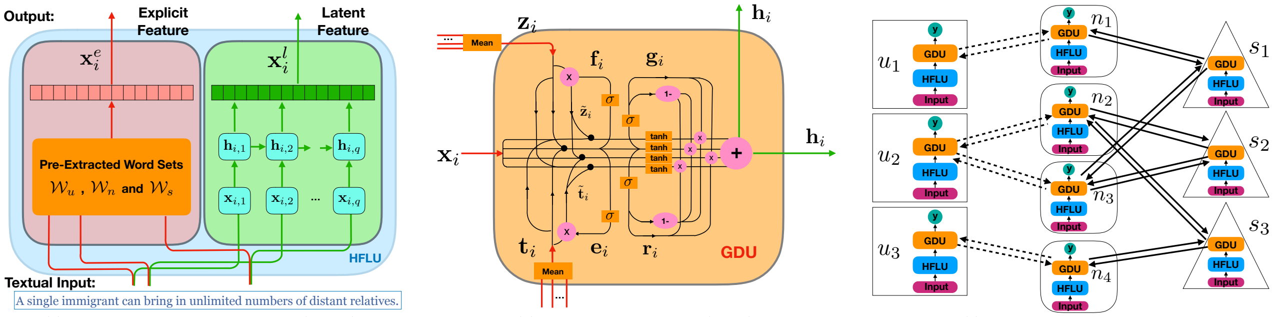
Here, operator $\otimes$ denotes the entry-wise product of vectors and $\mathbf{W}_f$ represents the variable of the forget gate in GDU.

Meanwhile, for the input from the creator nodes, a new node-type "adjust gate" is introduced in GDU. Here, the term "adjust" models the necessary changes of information between different node categories (e.g., from creators to articles). Formally, we can represent the "adjust gate" as well as the updated input as

$$\tilde{\mathbf{t}}_i = \mathbf{e}_i \otimes \mathbf{t}_i, \text{ where } \mathbf{e}_i = \sigma\left(\mathbf{W}_e \left[\mathbf{x}_i^\top, \mathbf{z}_i^\top, \mathbf{t}_i^\top\right]^\top\right),$$

where $\mathbf{W}_e$ denotes the variable matrix in the adjust gate.

GDU allows different combinations of these input/state vectors, which are controlled by the selection gates $\mathbf{g}_i$ and $\mathbf{r}_i$ respectively.



(a) Hybrid Feature Learning Unit (HFLU).

(b) Gated Diffusive Unit (GDU).

(c) Framework Architecture.

Figure 3: Unit Components and Overall Framework Architectures.

Formally, we can represent the final output of GDU as

$$\begin{aligned}\mathbf{h}_i &= \mathbf{g}_i \otimes \mathbf{r}_i \otimes \tanh\left(\mathbf{W}_u[\mathbf{x}_i^\top, \tilde{\mathbf{z}}_i^\top, \tilde{\mathbf{t}}_i^\top]^\top\right)\\ &\oplus (\mathbf{1} \ominus \mathbf{g}_i) \otimes \mathbf{r}_i \otimes \tanh\left(\mathbf{W}_u[\mathbf{x}_i^\top, \mathbf{z}_i^\top, \tilde{\mathbf{t}}_i^\top]^\top\right)\\ &\oplus \mathbf{g}_i \otimes (\mathbf{1} \ominus \mathbf{r}_i) \otimes \tanh\left(\mathbf{W}_u[\mathbf{x}_i^\top, \tilde{\mathbf{z}}_i^\top, \mathbf{t}_i^\top]^\top\right)\\ &\oplus (\mathbf{1} \ominus \mathbf{g}_i) \otimes (\mathbf{1} \ominus \mathbf{r}_i) \otimes \tanh\left(\mathbf{W}_u[\mathbf{x}_i^\top, \mathbf{z}_i^\top, \mathbf{t}_i^\top]^\top\right),\end{aligned}$$

where $\mathbf{g}_i = \sigma(\mathbf{W}_g \left[\mathbf{x}_i^\top, \mathbf{z}_i^\top, \mathbf{t}_i^\top\right]^\top)$, $\mathbf{r}_i = \sigma(\mathbf{W}_r \left[\mathbf{x}_i^\top, \mathbf{z}_i^\top, \mathbf{t}_i^\top\right]^\top)$, and $\mathbf{1}$ denotes a vector filled with value 1. Operators $\oplus$ and $\ominus$ denote the entry-wise addition and minus operation of vectors. Matrices $\mathbf{W}_u$, $\mathbf{W}_g$, $\mathbf{W}_r$ represent the variables involved in the components. Vector $\mathbf{h}_i$ will be the output of the GDU model.

The introduced GDU model also works for both the news subjects and creator nodes in the network. When applying the GDU to model the states of the subject/creator nodes with two input only, the remaining input port can be assigned with a default value (usually vector $\mathbf{0}$). Based on the GDU, we can denote the overall architecture of the FakeDetector as shown in Figure 3(c), where the lines connecting the GDUs denote the data flow among the unit models. In the following section, we will introduce how to learn the parameters involved in the FakeDetector model for concurrent credibility inference of multiple nodes.

4.3 Deep Diffusive Network Model Learning

In the FakeDetector model as shown in Figure 3(c), based on the output state vectors of news articles, news creators and news subjects, the framework will project the feature vectors to their credibility labels. Formally, given the state vectors $\mathbf{h}_{n,i}$ of news article n_i, $\mathbf{h}_{u,j}$ of news creator u_j, and $\mathbf{h}_{s,l}$ of news subject s_l, we can represent their inferred credibility labels as vectors $\mathbf{y}_{n,i}, \mathbf{y}_{u,j}, \mathbf{y}_{s,l} \in \mathcal{R}^{|\mathcal{Y}|}$ respectively, which can be represented as

$$\begin{cases}\mathbf{y}_{n,i} &= softmax\left(\mathbf{W}_n \mathbf{h}_{n,i}\right),\\ \mathbf{y}_{u,j} &= softmax\left(\mathbf{W}_u \mathbf{h}_{u,j}\right),\\ \mathbf{y}_{s,l} &= softmax\left(\mathbf{W}_s \mathbf{h}_{s,l}\right).\end{cases}$$

where $\mathbf{W}_u$, $\mathbf{W}_n$ and $\mathbf{W}_s$ define the weight variables projecting state vectors to the output vectors, and function $softmax(\cdot)$ represents the softmax function.

Meanwhile, based on the news articles in the training set $\mathcal{T}_n \subset \mathcal{N}$ with the ground-truth credibility label vectors $\{\hat{\mathbf{y}}_{n,i}\}_{n_i \in \mathcal{T}_n}$, we can define the loss function of the framework for news article credibility label learning as the cross-entropy between the prediction results and the ground truth:

$$\mathcal{L}(\mathcal{T}_n) = -\sum_{n_i \in \mathcal{T}_n} \sum_{k=1}^{|\mathcal{Y}|} \hat{\mathbf{y}}_{n,i}[k] \log \mathbf{y}_{n,i}[k].$$

Similarly, we can define the loss terms introduced by news creators and subjects based on training sets $\mathcal{T}_u \subset \mathcal{U}$ and $\mathcal{T}_s \subset \mathcal{S}$ as

$$\mathcal{L}(\mathcal{T}_u) = -\sum_{u_j \in \mathcal{T}_u} \sum_{k=1}^{|\mathcal{Y}|} \hat{\mathbf{y}}_{u,j}[k] \log \mathbf{y}_{u,j}[k],$$

$$\mathcal{L}(\mathcal{T}_s) = -\sum_{s_l \in \mathcal{T}_s} \sum_{k=1}^{|\mathcal{Y}|} \hat{\mathbf{y}}_{s,l}[k] \log \mathbf{y}_{s,l}[k],$$

where $\mathbf{y}_{u,j}$ and $\hat{\mathbf{y}}_{u,j}$ (and $\mathbf{y}_{s,l}$ and $\hat{\mathbf{y}}_{s,l}$) denote the prediction result vector and ground-truth vector of creator (and subject) respectively.

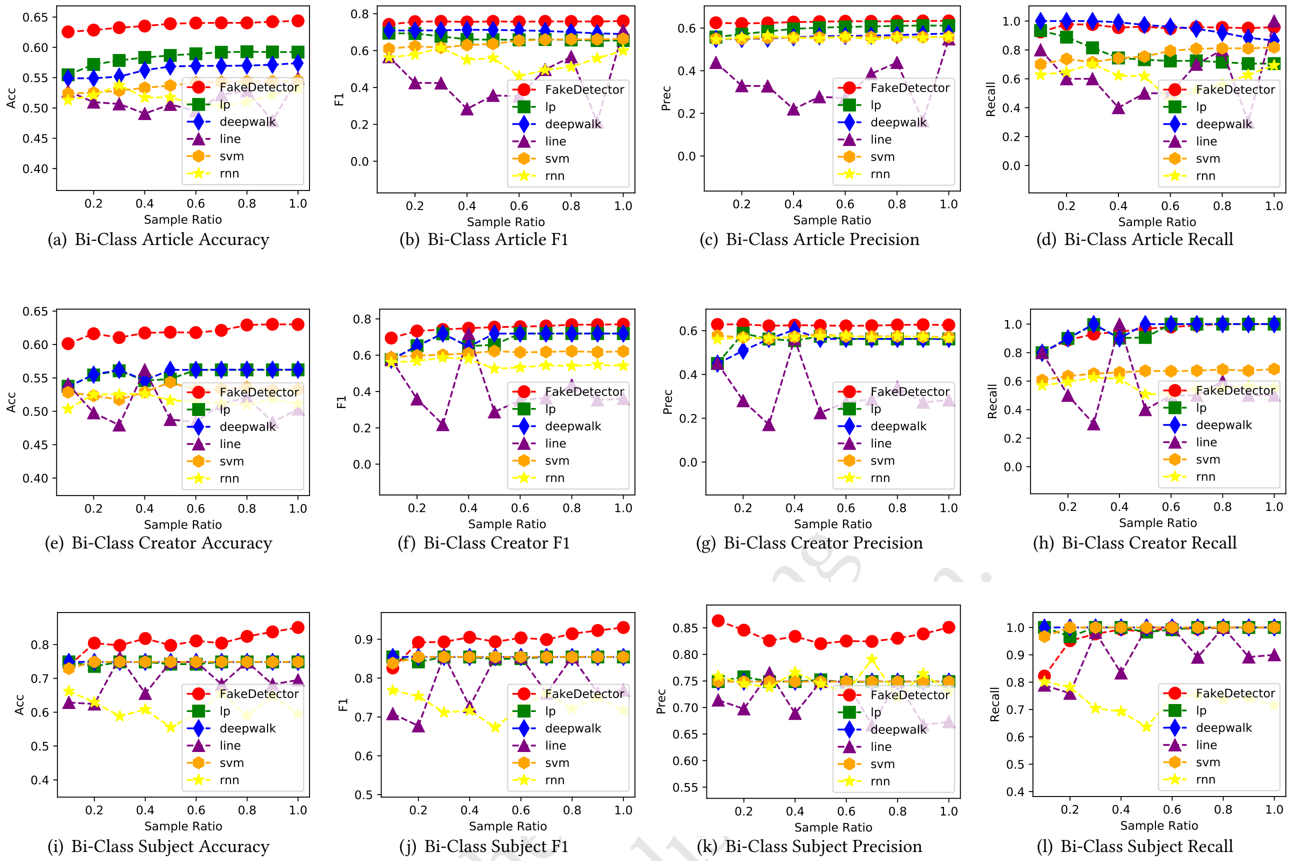
Formally, the main objective function of the FakeDetector model can be represented as follows:

$$\min_{\mathbf{W}} \mathcal{L}(\mathcal{T}_n) + \mathcal{L}(\mathcal{T}_u) + \mathcal{L}(\mathcal{T}_s) + \alpha \cdot \mathcal{L}_{reg}(\mathbf{W}),$$

where $\mathbf{W}$ denotes all the involved variables to be learned, term $\mathcal{L}_{reg}(\mathbf{W})$ represents the regularization term, and α denotes the regularization term weight. By resolving the optimization functions, we will be able to learn the variables involved in the framework. In this paper, we propose to train the framework with the *back-propagation* algorithm. For the news articles, creators and subjects in the testing set, their predicted credibility labels will be outputted as the final result.

5 EXPERIMENTS

To test the effectiveness of the proposed model, in this part, extensive experiments will be done on the real-world fake news dataset, PolitiFact. Detailed information about the PolitiFact dataset has been introduced in Section 3. In this section, we will first introduce the experimental settings, and the experimental results together with the detailed analysis will be provided after that.



(a) Bi-Class Article Accuracy (b) Bi-Class Article F1 (c) Bi-Class Article Precision (d) Bi-Class Article Recall

(e) Bi-Class Creator Accuracy (f) Bi-Class Creator F1 (g) Bi-Class Creator Precision (h) Bi-Class Creator Recall

(i) Bi-Class Subject Accuracy (j) Bi-Class Subject F1 (k) Bi-Class Subject Precision (l) Bi-Class Subject Recall

Figure 4: Bi-Class Credibility Inference of News Articles 4(a)-4(d), Creators 4(e)-4(h) and Subjects 4(i)-4(l).

5.1 Experimental Settings

The experimental setting covers (1) detailed experimental setups, (2) comparison methods and (3) evaluation metrics, which will be introduced as follows respectively.

5.1.1 Experimenta Setups. Based on the input PolitiFact dataset, we can represent the set of news articles, creators and subjects as $\mathcal{N}$, $\mathcal{U}$ and $\mathcal{S}$ respectively. With 10-fold cross validation, we propose to partition the news article, creator and subject sets into two subsets according to ratio $9:1$ respectively, where 9 folds are used as the training sets and 1 fold is used as the testing sets. Here, to simulate the cases with different number of training data. We further sample a subset of news articles, creators and subjects from the training sets, which is controlled by the sampling ratio parameter $\theta \in \{0.1, 0.2, \cdots, 1.0\}$. Here, $\theta = 0.1$ denotes 10% of instances in the 9 folds are used as the final training set, and $\theta = 1.0$ denotes 100% of instances in the 9 folds are used as the final training set. The known news article credibility labels will be used as the ground truth for model training and evaluation. Furthermore, based on the categorical labels of news articles, we propose to represent the 6 credibility labels with 6 numerical scores instead, and the corresponding relationships are as follows: “True”: 6, “Mostly True”: 5, “Half True”: 4, “Mostly False”: 3, “False”: 2, “Pants on Fire!”: 1. According to the known creator-article and subject-article relationships, we can also compute the credibility scores of creators and subjects, which can be denoted as the weighted sum of credibility scores of published articles (here, the weight denotes the percentage of articles in each class). And the credibility labels corresponding to the creator/subject round scores will be used as the ground truth as well. Based on the training sets of news articles, creators and subjects, we propose to build the FakeDetector model with their known textual contents, article-creator relationships, and article-subject relationships, and further apply the learned FakeDetector model to the test sets.

5.1.2 Comparison Methods. In the experiments, we compare FakeDetector extensively with many baseline methods. The list of used comparison methods are listed as follows:

- *FakeDetector*: Framework FakeDetector proposed in this paper can infer the credibility labels of news articles, creators and subjects with both explicit and latent textual features and relationship connections based on the Gdu model.
- *DeepWalk*: Model DeepWalk [48] is a network embedding model. Based on the fake news network structure, DeepWalk

embeds the news articles, creators and subjects to a latent feature space. Based on the learned embedding results, we can further build a SVM model to determine the class labels of the new articles, creators and subjects.

- *Line*: The Line model is a scalable network embedding model proposed in [57], which optimizes an objective function that preserves both the local and global network structures. Similar to DeepWalk, based on the embedding results, a classifier model can be further build to classify the news articles, creators and subjects.
- *Propagation*: In addition, merely based on the fake news heterogeneous network structure, we also propose to compare the above methods with a label-propagation based model proposed in [29], which also considers the node types and link types into consideration. The prediction score will be rounded and cast into labels according to the label-score mappings aforementioned.
- *Rnn*: In this paper, merely based on the textual contents, we propose to apply the Rnn model [42] to learn the latent representations of the textual input. Furthermore, the latent feature vectors will be fused to predict the news article, creator and subject credibility labels.
- *Svm*: Slightly different form Rnn, based on the raw text inputs, a set of explicit features can be extracted according to the descriptions in this paper, which will be used as the input for building a Svm [8] based classification model as the last baseline method..

Among these baseline methods, DeepWalk and Line use the network structure information only, and all build a classification model based on the network embedding results. Model Propagation also only uses the network structure information, but is based on the label propagation model instead. Both Rnn and Svm merely utilize the textual contents only, but their differences lies in: Rnn builds the classification model based on the latent features and Svm builds the classification model based on the explicit features instead.

5.1.3 Evaluation Metrics. Several frequently-used classification evaluation metrics will be used for the performance evaluation of the comparison methods. In the evaluation, we will cast the credibility inference problem into a binary class classification and a multi-class classification problem respectively. By grouping class labels {True, Mostly True, Half True} as the positive class and labels {Pants on Fire!, False, Mostly False} as the negative class, the credibility inference problem will be modeled as a binary-class classification problem, whose results can be evaluated by metrics, like Accuracy, Precision, Recall and F1. Meanwhile, if the model infers the original 6 class labels {True, Mostly True, Half True, Mostly False, False, Pants on Fire!} directly, the problem will be a multi-class classification problem, whose performance can be evaluated by metrics, like Accuracy, Macro Precision, Macro Recall and Macro F1 respectively.

5.2 Experimental Results

The experimental results are provided in Figures 4-5, where the plots in Figure 4 are about the binary-class inference results of articles, creators and subjects, while the plots in Figure 5 are about the multi-class inference results.

5.2.1 Bi-Class Inference Results. According to the plots in Figure 4, method FakeDetector can achieve the best performance among all the other methods in inferring the bi-class labels of news articles, creators and subjects (for all the evaluation metrics except Recall) with different sample ratios consistently. For instance, when the sample ratio $\theta = 0.1$, the Accuracy score obtained by FakeDetector in inferring the news articles is 0.63, which is more than 14.5% higher than the Accuracy score obtained by the network structure based models Propagation, DeepWalk, Line and the textual content based methods Rnn and Svm. Similar observations can be identified for the inference of creator credibility and subject credibility respectively.

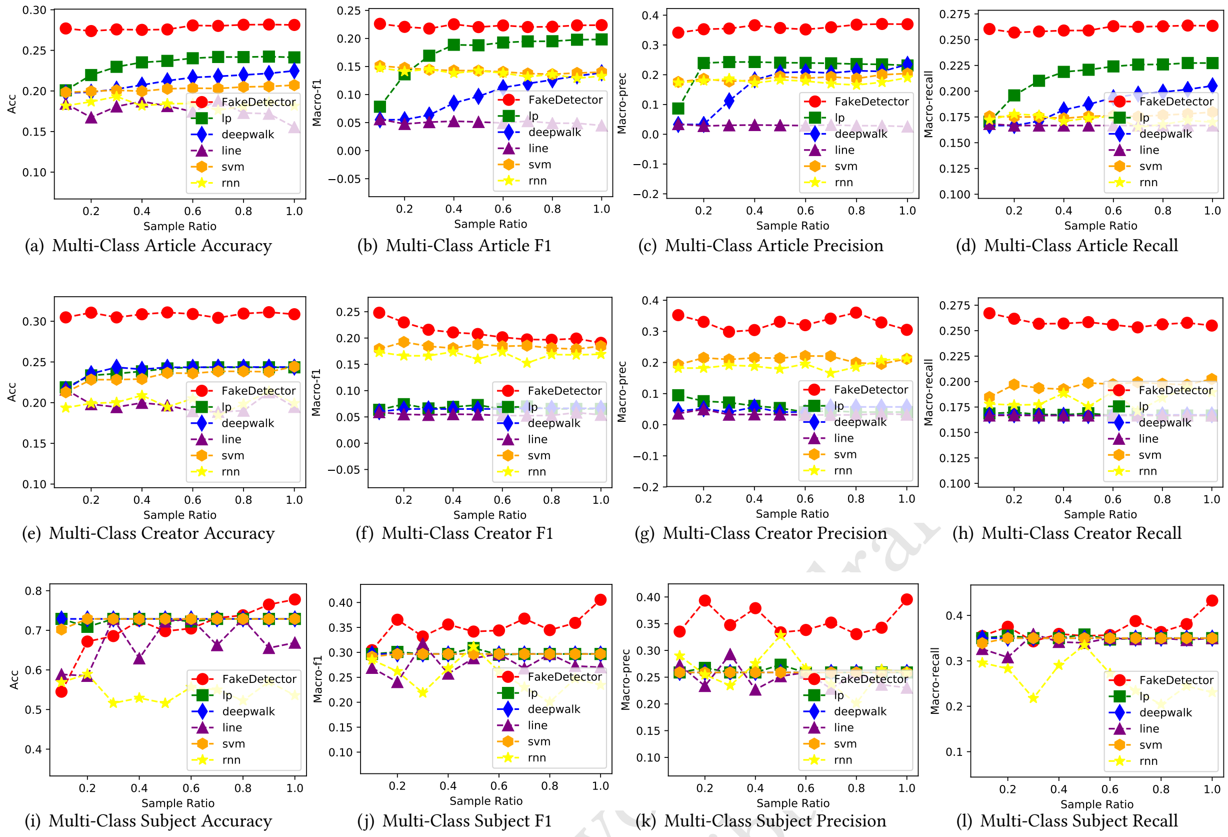
Among all the True news articles, creators and subjects identified by FakeDetector, a large proportion of them are the correct predictions. As shown in the plots, method FakeDetector can achieve the highest Precision score among all these methods, especially for the subjects. Meanwhile, the Recall obtained by FakeDetector is slightly lower than the other methods. By studying the prediction results, we observe that FakeDetector does predict less instances with the “True” label, compared with the other methods. The overall performance of FakeDetector (by balancing Recall and Precision) will surpass the other methods, and the F1 score obtained by FakeDetector greatly outperforms the other methods.

5.2.2 Multi-Class Inference Results. Besides the simplified bi-class inference problem setting, we further infer the information entity credibility at a finer granularity: infer the labels of instances based the original 6-class label space. The inference results of all the comparison methods are available in Figure 5. Generally, according to the performance, the advantages of FakeDetector are much more significant compared with the other methods in the multi-class prediction setting. For instance, when $\theta = 0.1$, the Accuracy score achieved by FakeDetector in inferring news article credibility score is 0.28, which is more than 40% higher than the Accuracy obtained by the other methods. In the multi-class scenario, both the Macro-Precision and Macro-Recall scores of FakeDetector are also much higher than the other methods. Meanwhile, by comparing the inference scores obtained by the methods in Figures 4 and Figure 5, the multi-class credibility inference scenario is much more difficult and the scores obtained by the methods are much lower than the bi-class inference setting.

6 RELATED WORK

Several research topics are closely correlated with this paper, including *fake news analysis*, *spam detection* and *deep learning*, which will be briefly introduced as follows.

Fake News Preliminary Works: Due the increasingly realized impacts of fake news since the 2016 election, some preliminary research works have been done on fake news detection. The first work on online social network fake news analysis for the election comes from Allcott et al. [4]. The other published preliminary works mainly focus on fake news detection instead [51, 53, 54, 56]. Rubin et al. [51] provides a conceptual overview to illustrate the unique



(a) Multi-Class Article Accuracy (b) Multi-Class Article F1 (c) Multi-Class Article Precision (d) Multi-Class Article Recall

(e) Multi-Class Creator Accuracy (f) Multi-Class Creator F1 (g) Multi-Class Creator Precision (h) Multi-Class Creator Recall

(i) Multi-Class Subject Accuracy (j) Multi-Class Subject F1 (k) Multi-Class Subject Precision (l) Multi-Class Subject Recall

Figure 5: Multi-Class Credibility Inference of News Articles 5(a)-5(d), Creators 5(e)-5(h) and Subjects 5(i)-5(l).

features of fake news, which tends to mimic the format and style of journalistic reporting. Singh et al. [54] propose a novel text analysis based computational approach to automatically detect fake news articles, and they also release a public dataset of valid new articles. Tacchini et al. [56] present a technical report on fake news detection with various classification models, and a comprehensive review of detecting spam and rumor is presented by Shu et al. in [53]. In this paper, we are the first to provide the systematic formulation of fake news detection problems, illustrate the fake news presentation and factual defects, and introduce unified frameworks for fake news article and creator detection tasks based on deep learning models and heterogeneous network analysis techniques.

Spam Detection Research and Applications: Spams usually denote unsolicited messages or emails with unconfirmed information sent to a large number of recipients on the Internet. The concept *web spam* was first introduced by Convey in [10] and soon became recognized by the industry as a key challenge [25]. Spam on the Internet can be categorized into *content spam* [14, 41, 50], *link spam* [2, 23, 69], *cloaking and redirection* [9, 37, 64, 65], and *click spam* [11, 13, 30, 47, 49]. Existing detection algorithms for these spams can be roughly divided into three main groups. The first group involves the techniques using content based features, like word/language model [17, 45, 55] and duplicated content analysis [15, 16, 59]. The second group of techniques mainly rely on the graph connectivity information [7, 18, 20, 21], like link-based trust/distrust propagation [24, 33, 46], pruning of connections [6, 36, 44]. The last group of techniques use data like click stream [13, 49], user behavior [39, 40], and HTTP session information [61] for spam detection. The differences between fake news and conventional spams have been clearly illustrated in Section 1, which also make these existing spam detection techniques inapplicable to detect fake news articles.

Deep Learning Research and Applications: The essence of deep learning is to compute hierarchical features or representations of the observational data [22, 35]. With the surge of deep learning research and applications in recent years, lots of research works have appeared to apply the deep learning methods, like deep belief network [28], deep Boltzmann machine [52], Deep neural network [31, 34] and Deep autoencoder model [60], in various applications, like speech and audio processing [12, 27], language modeling and processing [5, 43], information retrieval [26, 52], objective recognition and computer vision [35], as well as multimodal and multi-task learning [62, 63].

7 CONCLUSION

In this paper, we have studied the fake news article, creator and subject detection problem. According to the data analysis, a set of explicit and latent features can be extracted from the textual information of news articles, creators and subjects respectively. Furthermore, based on the connections among news articles, creators and news subjects, a deep diffusive network model has been proposed for incorporate the network structure information into model learning. In this paper, we also introduce a new diffusive unit model, namely GDU. Model GDU accepts multiple inputs from different sources simultaneously, and can effectively fuse these input for output generation with content "forget" and "adjust" gates. Extensive experiments done on a real-world fake news dataset, i.e., PolitiFact, have demonstrated the outstanding performance of the proposed model in identifying the fake news articles, creators and subjects in the network.

REFERENCES

[1] Great moon hoax. https://en.wikipedia.org/wiki/Great_Moon_Hoax. [Online; accessed 25-September-2017].

[2] S. Adali, T. Liu, and M. Magdon-Ismail. Optimal link bombs are uncoordinated. In *AIRWeb*, 2005.

[3] L. Akoglu, R. Chandy, and C. Faloutsos. Opinion fraud detection in online reviews by network effects. In *ICWSM*, 2013.

[4] H. Allcott and M. Gentzkow. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 2017.

[5] E. Arisoy, T. Sainath, B. Kingsbury, and B. Ramabhadran. Deep neural network language models. In *WLM*, 2012.

[6] K. Bharat and M. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *SIGIR*, 1998.

[7] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: web spam detection using the web topology. In *SIGIR*, 2007.

[8] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[9] K. Chellapilla and D. Chickering. Improving cloaking detection using search query popularity and monetizability. In *AIRWeb*, 2006.

[10] E. Convey. Porn sneaks way back on web. *The Boston Herald*, 1996.

[11] N. Daswani and M. Stoppelman. The anatomy of clickbot.a. In *HotBots*, 2007.

[12] L. Deng, G. Hinton, and B. Kingsbury. New types of deep neural network learning for speech recognition and related applications: An overview. In *ICASSP*, 2013.

[13] Z. Dou, R. Song, X. Yuan, and J. Wen. Are click-through data adequate for learning web search rankings? In *CIKM*, 2008.

[14] I. Drost and T. Scheffer. Thwarting the nigritude ultramarine: Learning to identify link spam. In *ECML*, 2005.

[15] D. Fetterly, M. Manasse, , and M. Najork. On the evolution of clusters of near-duplicate web pages. In *LA-WEB*, 2003.

[16] D. Fetterly, M. Manasse, , and M. Najork. Detecting phrase-level duplication on the world wide web. In *SIGIR*, 2005.

[17] D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *WebDB*, 2004.

[18] Q. Gan and T. Suel. Improving web spam classifiers using link structure. In *AIRWeb*, 2007.

[19] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Zhao. Detecting and characterizing social spam campaigns. In *IMC*, 2010.

[20] G. Geng, Q. Li, and X. Zhang. Link based small sample learning for web spam detection. In *WWW*, 2009.

[21] G. Geng, C. Wang, and Q. Li. Improving web spam detection with re-extracted features. In *WWW*, 2008.

[22] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[23] Z. Gyöngyi and H. Garcia-Molina. Link spam alliances. In *VLDB*, 2005.

[24] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *VLDB*, 2004.

[25] M. Henzinger, R. Motwani, and C. Silverstein. Challenges in web search engines. In *SIGIR Forum*. 2002.

[26] G. Hinton. A practical guide to training restricted boltzmann machines. In *Neural Networks: Tricks of the Trade (2nd ed.)*. 2012.

[27] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, 2012.

[28] G. Hinton, S. Osindero, and Y. Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 2006.

[29] Q. Hu, S. Xie, J. Zhang, Q. Zhu, S. Guo, and P. Yu. Heterosales: Utilizing heterogeneous social networks to identify the next enterprise customer. In *WWW*, 2016.

[30] N. Immorlica, K. Jain, M. Mahdian, and K. Talwar. Click fraud resistant methods for learning click-through rates. In *WINE*, 2005.

[31] H. Jaeger. Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach. Technical report, Fraunhofer Institute for Autonomous Intelligent Systems (AIS), 2002.

[32] Y. Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.

[33] V. Krishnan and R. Raj. Web spam detection with anti-trust rank. In *AIRWeb*, 2006.

[34] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[35] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521, 2015. http://dx.doi.org/10.1038/nature14539.

[36] R. Lempel and S. Moran. Salsa: the stochastic approach for link-structure analysis. *TIST*, 2001.

[37] J. Lin. Detection of cloaked web spam by using tag-based methods. *Expert Systems with Applications*, 2009.

[38] S. Lin, Q. Hu, J. Zhang, and P. Yu. *Discovering Audience Groups and Group-Specific Influencers*. 2015.

[39] Y. Liu, B. Gao, T. Liu, Y. Zhang, Z. Ma, S. He, and H. Li. Browserank: letting web users vote for page importance. In *SIGIR*, 2008.

[40] Y. Liu, M. Zhang, S. Ma, and L. Ru. User behavior oriented web spam detection. In *WWW*, 2008.

[41] O. Mcbryan. Genvl and wwww: Tools for taming the web. In *WWW*, 1994.

[42] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, 2010.

[43] A. Mnih and G. Hinton. A scalable hierarchical distributed language model. In *NIPS*. 2009.

[44] S. Nomura, S. Oyama, T. Hayamizu, and T. Ishida. Analysis and improvement of hits algorithm for detecting web communities. *Syst. Comput. Japan*, 2004.

[45] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *WWW*, 2006.

[46] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *WWW*, 1998.

[47] Y. Peng, L. Zhang, J. M. Chang, and Y. Guan. An effective method for combating malicious scripts clickbots. In *ESORICS*, 2009.

[48] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *KDD*, 2014.

[49] F. Radlinski. Addressing malicious noise in click-through data. In *AIRWeb*, 2007.

[50] S. Robertson, H. Zaragoza, and M. Taylor. Simple bm25 extension to multiple weighted fields. In *CIKM*, 2004.

[51] V. Rubin, N. Conroy, Y. Chen, and S. Cornwell. Fake news or truth? using satirical cues to detect potentially misleading news. In *NAACL-CADD*, 2016.

[52] R. Salakhutdinov and G. Hinton. Semantic hashing. *International Journal of Approximate Reasoning*, 2009.

[53] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu. Fake news detection on social media: A data mining perspective. *SIGKDD Explor. Newsl.*, 2017.

[54] V. Singh, R. Dasgupta, D. Sonagra, K. Raman, and I. Ghosh. Automated fake news detection using linguistic analy- sis and machine learning. In *SBP-BRiMS*, 2017.

[55] K. Svore, Q. Wu, C. Burges, and A. Raman. Improving web spam classification using rank-time features. In *AIRWeb*, 2007.

[56] E. Tacchini, G. Ballarin, M. Della Vedova, S. Moret, and L. de Alfaro. Some like it hoax: Automated fake news detection in social networks. *CoRR*, abs/1704.07506, 2017.

[57] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *WWW*, 2015.

[58] Y. Teng, C. Tai, P. Yu, and M. Chen. Modeling and utilizing dynamic influence strength for personalized promotion. In *ASONAM*, 2015.

[59] T. Urvoy, T. Lavergne, and P. Filoche. Tracking web spam with hidden style similarity. In *AIRWeb*, 2006.

[60] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 2010.

[61] S. Webb, J. Caverlee, and C. Pu. Predicting web spam with http session information. In *CIKM*, 2008.

[62] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: Learning to rank with joint word-image embeddings. *Journal of Machine Learning*, 2010.

[63] J. Weston, S. Bengio, and N. Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *IJCAI*, 2011.

[64] B. Wu and B. Davison. Cloaking and redirection: A preliminary study. In *AIRWeb*, 2005.

[65] B. Wu and B. Davison. Detecting semantic cloaking on the web. In *WWW*, 2006.

[66] S. Xie, G. Wang, S. Lin, and P. Yu. Review spam detection via temporal pattern discovery. In *KDD*, 2012.

[67] S. Xie, G. Wang, S. Lin, and P. Yu. Review spam detection via time series pattern discovery. In *WWW*, 2012.

[68] Q. Zhan, J. Zhang, S. Wang, P. Yu, and J. Xie. Influence maximization across partially aligned heterogenous social networks. In *PAKDD*. 2015.

[69] B. Zhou and J. Pei. Sketching landscapes of page farms. In *SDM*, 2007.