Deep Graph Neural Network-based Spammer Detection Under the Perspective of Heterogeneous Cyberspace

Zhiwei Guo[a], Lianggui Tang[a], Tan Guo[b], Keping Yu[c,*], Mamoun Alazab[d], Andrii Shalaginov[e]

[a]*School of Artificial Intelligence, National Research Base of Intelligent Manufacturing Service, Chongqing Technology and Business University, Chongqing 400067, China*
[b]*School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[c]*Global Information and Telecommunication Institute, Waseda University, Shinjuku, Tokyo 169-8050, Japan*
[d]*College of Engineering, IT and Environment, Charles Darwin University, Australia*
[e]*Department of Information Security and Communication Technology, Norwegian University of Science and Technology, Gjovik, Norway*

Abstract

Due to the severe threat to cyberspace security, detection of online spammers has been a universal concern of academia. Nowadays, prevailing literature of this field almost leveraged various relations to enhance feature spaces. However, they majorly focused stable or visible relations, yet neglected the existence of those which are generated occasionally. Exactly, some latent feature components can be extracted from the view of heterogeneous information networks. Thus, this paper proposes a Deep Graph neural network-based Spammer detection (DeG-Spam) model under the perspective of heterogeneous cyberspace. Specifically, representations for occasional relations and inherent relations are separately modelled. Based on this, a graph neural network framework is formulated to generate feature expressions for the social graph. With more feature components being mined, acquirement of stronger and more comprehensive feature spaces ensures the accuracy of spammer detection. At last, fruitful experiments are carried out on two benchmark datasets to compare the DeG-Spam with typical

*Corresponding author
Email addresses: zwguo@ctbu.edu.cn (Zhiwei Guo), tlg@ctbu.edu.cn (Lianggui Tang), guot@cqupt.edu.cn (Tan Guo), keping.yu@aoni.waseda.jp (Keping Yu), mamoun.alazab@cdu.edu.au (Mamoun Alazab), andrii.shalaginov@ntnu.no (Andrii Shalaginov)

spammer detection approaches. Experimental results show that it performs about 5%-10% better than baselines.

Keywords: cyberspace security, spammer detection, graph neural network, heterogeneous social graph

1. Introduction

With the rapid development of Internet technology, the cyberspace has been a novel working and living space in contemporary world [1]. Despite much convenience brought for human beings, the security threat faced by cyberspace has gradually been a serious problem that cannot be ignored [2, 3]. Most typically, a class of communities named online spammers always spread malicious statements in cyberspace to mislead public opinions [4, 5], so that some political and commercial goals can be achieved [6]. Nowadays, the issue of online spamming has already evolved into the universal trouble around the world [7], harming social stability and even national security [8, 9]. For instance, during the worldwide epidemic COVID-19, spreading of various tendentious speeches in cyberspace hindered works of fighting against the epidemic in many countries. To guarantee the strong capability of cyber defence, the significance of effective spammer detection technologies is acknowledged [10]. In essence, it remains not an easy task to make accurate identification towards spammers [11]. Because online spamming generally occurs inside complicated circumstances where direct features are quite sparse [12]. Therefore, deep extraction of fine features determines detection efficiency to a large extent [13, 14].

During the past decade, with the vigorous development of artificial intelligence, substantial progress has been acquired in the field of spammer detection [9, 15]. As mentioned in one of our previously published study [1], the most intuitive idea is to model semantic meanings of speech contents [16, 17, 18, 19, 20, 21, 22]. But the semantics-based approaches are just suitable for spammers with highly regular speeches [23]. In general, spamming is not a kind of singly linguistic activity and is accompanied by contextual information such as social relations and even financial links [24]. Hence, it is supposed to extract abstract features from fruitful contextual information to enhance feature spaces [25]. Realizing this view, contextual information-based spammer detection acts as the mainstream up to now [26], yielding a number of representative technical methods [27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38]. Although they are able to perform well in some cases, one major shortcoming still exists inside them. Almost all of them just considered relatively stable and visible relations, without noticing those which are imperceptible or generated provisionally. In particular, each social graph is heterogeneous and can be divided into multiple types of subgraphs. And many heterogeneous relation links can be extracted from it to deduce latent linkages among entities. Figure 1 gives a typical example to illustrate this view. Three types of subgraphs can be separated from the social graph: "*User-Community*" subgraph, "*User-Post*" subgraph, and "*Post-Topic*" subgraph. It is assumed that the three users are not friends in the real world. A relation link, "*User A-Community 1-User B-Community 2-User C*", can be extracted from the social graph to denote their provisional relations. In reality, cyberspace is generally such kind of heterogeneous environment filled with uncertainty and complexity. To improve spammer detection efficiency in cyberspace, It is expected to deeply mine unknown linkages from the view of heterogeneous information network [39].

To bridge such gap, relations inside each social graph are specialized into two categories: stable relations and occasional relations. And a graph embedding-based hybrid neural network architecture can be developed to realize joint modelling operation. Therefore, this paper proposes a **De**ep **G**raph neural network-based **Spam**mer detection model (DeG-Spam) under the perspective of heterogeneous cyberspace. Particularly, representation for occasional relations is inferred from initial social graphs via the parametric random walk model [40], and representation for stable relations is modelled via direct vectorized encoding. Joint modelling of them is endowed with the capability to capture deeper-level characteristics, and leads to more comprehensive feature representation for social graphs. Base upon this, detection accuracy can be promoted compared with

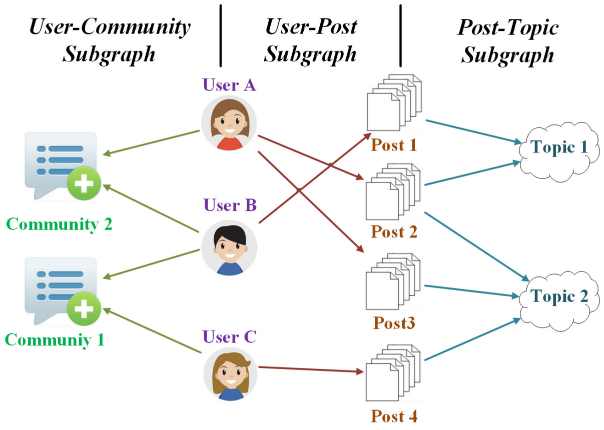


Figure 1: An example for Social Network Heterogeneity.

previous works. Besides, the methodology can be used as a stepping stone to combat the cybercrime and withstand growing number of attacks on end-users. To the best of our knowledge, the idea of mining hidden relations in heterogeneous cyberspace had never been put forward by any other researchers. Main contributions of this paper can be summarized as the following points:

1) Existence of occasional relations under the perspective of heterogeneous cyberspace is recognized, and its roles to spammer detection is illustrated.
2) A deep graph neural network-based spammer detection model is proposed to mine more comprehensive relational features.
3) The efficiency of the proposal is evaluated on two real-world datasets, showing proper performance compared with baselines.

The remainder of this paper is organized as follows. Section 2 introduces the problem scenarios and gives basic definitions. In Section 3, the detailed mathematical process of the DeG-Spam is described in detail. Experimental settings, results and analysis are displayed in Section 4. And we conclude this paper in Section 5.

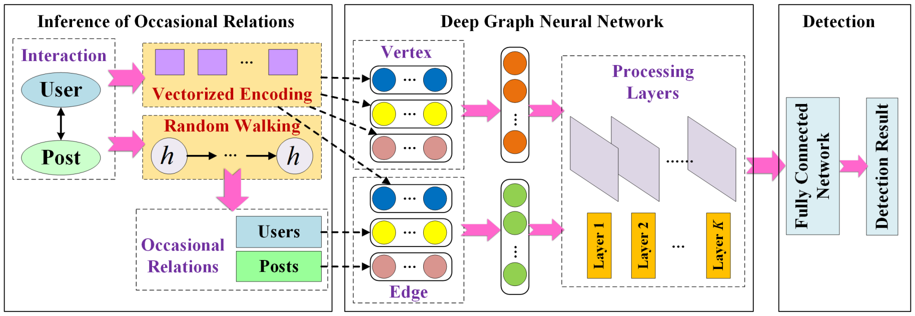


Figure 2: Main Architecture of the DeG-Spam.

2. Problem Statement

The main goal of this research is to distinguish spammers from a set of users, according to their patterns of speech contents and behaviours. Their speeches are usually released in the form of short texts on social media such as Twitter and Weibo. For simplicity, the following two definitions are firstly deduced:

Definition 1 (Post). A piece of the specific speech published by a user is defined as a post, such as a tweet or a microblog.

Definition 2 (Interaction). An activity that a user releases a post, is defined as an interaction. The number of interactions of a user equals to the number of his speeches.

In general, the initial posts are short and profiles of users are not informative. It is expected to deeply mine feature components of both users and posts. Different from existing researches, each interaction is viewed as a sample in this research. All the interactions in the repository are regarded as a graph network, in which vertices refer to interactions themselves and edges refer to relations among interactions. As for vertices, their representation derives from the semantics of posts and profile features of users. As for edges, their representation comes from relation features among users. All of the interactions of a user are integrated, and the joint effect of them determines the nature of the user. Based on such an idea, this research is established based on the following assumptions:

Assumption 1. For each user, the decision of interactions is not directly affected by others. They are assumed to make decisions independently.

Assumption 2. For each post, text associated with it needs to be not too short and can be semantically modelling.

Assumption 3. For the interactions of a user, they are assumed to be not sequential. Interactions that have happened will not influence the following ones.

The major architecture of the proposed DeG-Spam is demonstrated in Figure 2. Let u_i $(i = 1, 2, \cdots, |u|)$ denote the set of users, and p_j $(j = 1, 2, \cdots, |p|)$ denote the set of posts, which are two types of entities. To deeply extract feature expressions for interactions, the following definition is deduced:

Definition 3 (Heterogeneous Social Network). As is shown in the example of Figure 1, a social network composed of users and posts is a heterogeneous social graph which contains three types of subgraphs. The whole social graph contains four types of objects in this research: communities, users, posts, and topics. Naturally, four types of objects are viewed as four types of nodes.

To model provisional relations of users and posts, random walk scheme is utilized to link two types of nodes inside subgraphs. Thus, the definition of a walking path is deduced:

Definition 4 (Walking Path). It refers to directed links inside subgraphs by randomly sampling different types of nodes in sequence. In each time of sampling, one of the other type of nodes will be selected as the next node, according to some probabilistic distributions. Note that every two adjacent nodes belong to different types.

Inside each walking path, nodes whose types are different from the starting node need to be removed from the walking path. Thus, the heterogeneous walking path is transformed into a homogeneous one. After filtering, such link indicates occasional relations between the starting node and other nodes with the same type. The homogeneous directed link can be also regarded as a sequential propagation process and modelled via the gated recurrent unit (GRU) model. Through a series of walking paths, occasional relations of users and posts can be obtained. For user u_i and post p_j, occasional relations of them are further encoded into two representative vectors: $\mathcal{R}_{ui}^{(occ)}$ and $\mathcal{R}_j$. Besides, attributes of user u_i can be divided into numerical attributes and categorical attributes. Representative vectors of them are denoted as $\mathcal{C}_i^{(nu)}$ and $\mathcal{C}_i^{(ca)}$. Semantics of post p_j is denoted as the representative vector $\mathcal{C}_j^{(se)}$. Representative vectors for social relations of user u_i is denoted as $\mathcal{R}_{ui}^{(inh)}$.

As mentioned above, an interaction between user u_i and post p_j is viewed as a vertex and its relations are viewed as edges. Representative vector for its vertex features, $\mathcal{C}_{i,j}$, is obtained by concatenation of $\mathcal{C}_i^{(nu)}$, $\mathcal{C}_i^{(ca)}$, and $\mathcal{C}_j^{(se)}$. Representative vector for its edge features, $\mathcal{R}_{i,j}$, is obtained by concatenation of $\mathcal{R}_{ui}^{(inh)}$, $\mathcal{R}_{ui}^{(occ)}$, and $\mathcal{R}_j$. Then, the $\mathcal{C}_{i,j}$ and $\mathcal{R}_{i,j}$ are input into a developed graph neural network with $\mathcal{K}$ processing layers. After that, a hidden vector $\mathcal{H}_{i,j}^{(\mathcal{K})}$ is obtained to denote encoding state of the interaction. And it can be mapped into detection result for the user u_i.

3. Methodology

This section describes the mathematical modelling procedures of the proposed DeG-Spam, which contains three parts. Firstly, representative vectors of occasional relations with parameters are deduced. Secondly, a deep graph neural network framework is formulated to generate feature expressions for interactions of users. Thirdly, the detection is viewed as a binary classification problem and results are output through a sigmoid activation function.

3.1. Inference of Occasional Relations

For the node user u_i, its occasional relations are sampled via random walking from itself to other nodes in sequence. The index number of sampling rounds is denoted as τ and ranges from 1 to q. In the τ-th round of sampling, it is expected to produce a walking path $\mathcal{V}_n^\tau(u_i)$ consisting of a sequence of nodes. The index number of nodes is denoted as n and ranges from 1 to $\mathcal{N}$. The $\mathcal{V}_n^\tau(u_i)$ takes the following format:

$$\mathcal{V}_1^\tau\left(u_i\right) \rightarrow \mathcal{V}_2^\tau\left(u_i\right) \rightarrow \cdots \rightarrow \mathcal{V}_{\mathcal{N}-1}^\tau\left(u_i\right) \rightarrow \mathcal{V}_{\mathcal{N}}^\tau\left(u_i\right) \tag{1}$$

where nodes with odd index numbers are attributes. According to Definition 4, each walking path is a directed link of heterogeneous nodes. Each two adjacent nodes belong to different types, in which the latter one belongs to the first-order neighbor of the former one. During the generative process of walking path $\mathcal{V}_n^\tau(u_i)$, transformation probability from the n-th node to the $(n+1)$-th node is drawn from the following multinomial distribution:

$$P\left[\mathcal{V}_{n+1}^\tau\left(u_i\right)|\mathcal{V}_n^\tau\left(u_i\right)\right] = \begin{cases} \frac{1}{|Nei[\mathcal{V}_n^\tau(u_i)]|}, & \mathcal{V}_{n+1}^\tau\left(u_i\right) \in Nei\left[\mathcal{V}_n^\tau\left(u_i\right)\right] \\ 0, & otherwise \end{cases} \tag{2}$$

where $Nei\left[\mathcal{V}_n^\tau\left(u_i\right)\right]$ denotes the first-order neighbor set of the n-th node. Taking Figure 1 as an example, two walking paths can be deduced from *User A* to *User C*: 1) *User A* →*Community 1*→*User C*; 2) *User A*→*Community 1*→ *User B*→*Community 2*→*User C*.

Since the spammer detection problem discriminates nature of users by modeling the historical interaction behaviors between users and posts, main focus of this part is to learn representative vectors for two types of entities: users and posts. Thus, only paths which start with these two types of nodes are employed for modeling. For a sampled walking path, it contains two types of nodes which are assigned even and odd index numbers, respectively. In order to eliminate heterogeneity of the walking path, nodes with even index numbers need to be removed from the walking path. After filtering, it is transformed into another node sequence $\mathcal{B}_m^\tau\left(u_i\right)$, where m is the index number that ranges from 1 to $\mathcal{M}$. And $\mathcal{B}_m^\tau\left(u_i\right)$ takes the following format:

$$\mathcal{B}_1^\tau\left(u_i\right) \rightarrow \mathcal{B}_2^\tau\left(u_i\right) \rightarrow \cdots \rightarrow \mathcal{B}_{\mathcal{M}-1}^\tau\left(u_i\right) \rightarrow \mathcal{B}_{\mathcal{M}}^\tau\left(u_i\right) \tag{3}$$

Thus, a heterogeneous node sequence $\mathcal{V}_n^\tau(u_i)$ is transformed into a homoge-

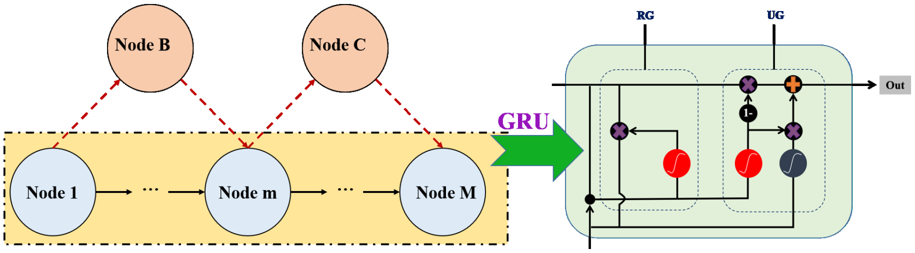


Figure 3: Workflow for Inference of occasional relations.

neous one: $\mathcal{B}_m^\tau(u_i)$. More importantly, a relation linkage is extracted from node $\mathcal{B}_1^\tau(u_i)$ to $\mathcal{B}_\mathcal{M}^\tau(u_i)$. As it is a directed walking sequence, generation of its representative vectors need to undergo a sequential transition modeling process which can be modeled by GRU model, which can be shown as Figure 3. Note that index number of the recurrent rounds in GRU is m and the total number of rounds is $\mathcal{M}$. As for the m-th round, the hidden state vector is updated via GRU operator according to that of the $(m-1)$-th round. The GRU is a state control-based neural network model and consists of two gates: update gate (UG) and reset gate (RG). UG controls the degree where state information of the previous round is brought into the current round, and RG controls the degree where state information of the previous round is neglected. During current transition round of τ-th walking path, hidden state is represented as:

$$\mathcal{H}_m^\tau = \overrightarrow{\Phi}\left[\mathcal{H}_{m-1}^\tau, \mathcal{U}_m^\tau\right] \tag{4}$$

where $\mathcal{U}_\tau^{(m)}$ is the network state vector of the m-th round. It is obtained mainly by modeling transition relation between the $(m-1)$-th round and the m-th round. Here, each directed transition state is separated into two step: output of previous state and input of current state. More concretely, node $\mathcal{B}_m^\tau(u_i)$ is associated with two transformation matrices: $\mathcal{Q}_{OUT}\left[\mathcal{B}_m^\tau(u_i)\right]$ and $\mathcal{Q}_{IN}\left[\mathcal{B}_m^\tau(u_i)\right]$. The former is of output form and the latter is of input form. Naturally, transition state from the $(m-1)$-th node $\mathcal{B}_{m-1}^\tau(u_i)$ to the m-th node $\mathcal{B}_m^\tau(u_i)$, corresponds to two transformation matrices: $\mathcal{Q}_{OUT}\left[\mathcal{B}_{m-1}^\tau(u_i)\right]$ and $\mathcal{Q}_{IN}\left[\mathcal{B}_m^\tau(u_i)\right]$. Thus,

$\mathcal{U}_m^\tau$ is calculated as:

$$\mathcal{U}_m^\tau(u_i) = \Lambda_{m-1,m}^\tau(u_i) \cdot \mathcal{H}_{m-1}^\tau(u_i) \cdot \mathcal{Q}_{OUT}\left[\mathcal{B}_{m-1}^\tau(u_i)\right] \cdot \mathcal{Q}_{IN}\left[\mathcal{B}_m^\tau(u_i)\right] \tag{5}$$

where $\Lambda_{m-1,m}^\tau(u_i)$ is the similarity weight between the $(m-1)$-th node and the m-th node, and is measured as:

$$\Lambda_{m-1,m}^\tau(u_i) = \frac{\mathcal{O}_{m-1,m}^{up}}{\mathcal{O}_{m-1,m}^{down}} \tag{6}$$

where

$$\mathcal{O}_{m-1,m}^{up} = \frac{\delta\left[\mathcal{B}_m^\tau(u_i) \cap \mathcal{B}_{m-1}^\tau(u_i)\right]}{\delta\left[\mathcal{B}_m^\tau(u_i)\right]} \tag{7}$$

$$\mathcal{O}_{m-1,m}^{down} = \sum_{\eta=1}^{\mathcal{M}} \frac{\delta\left[\mathcal{B}_m^\tau(u_i) \cap \mathcal{B}_\eta^\tau(u_i)\right]}{\delta\left[\mathcal{B}_\eta^\tau(u_i)\right]} \tag{8}$$

where $\delta\left[\mathcal{B}_m^\tau(u_i) \cap \mathcal{B}_{m-1}^\tau(u_i)\right]$ counts the number of common interactive posts between $\mathcal{B}_m^\tau(u_i)$ and $\mathcal{B}_{m+1}^\tau(u_i)$, $\delta\left[\mathcal{B}_m^\tau(u_i) \cap \mathcal{B}_\eta^\tau(u_i)\right]$ counts the total number of common interactive posts between $\mathcal{B}_m^\tau[u_i]$ and all the other $(\mathcal{M}-1)$ nodes, $\delta\left[\mathcal{B}_m^\tau(u_i)\right]$ counts the number of interactive posts for $\mathcal{B}_m^\tau(u_i)$, and $\delta\left[\mathcal{B}_\eta^\tau(u_i)\right]$ counts the number of interactive posts for all the other $(\mathcal{M}-1)$ nodes. Given above, state vectors of UG and RG for the m-th node are separately represented as:

$$\mathcal{I}_m^{\tau-UG}(u_i) = \sigma_1\left[\mathcal{W}_{\mathcal{I}1} \cdot \mathcal{U}_m^\tau(u_i) + \mathcal{W}_{\mathcal{I}2} \cdot \mathcal{H}_m^\tau(u_i) + b_{\mathcal{I}1}\right] \tag{9}$$

$$\mathcal{I}_m^{\tau-RG}(u_i) = \sigma_1\left[\mathcal{W}_{\mathcal{I}3} \cdot \mathcal{U}_m^\tau(u_i) + \mathcal{W}_{\mathcal{I}4} \cdot \mathcal{H}_m^\tau(u_i) + b_{\mathcal{I}3}\right] \tag{10}$$

where $\sigma_1(\cdot)$ is the ReLU activation function, $\mathcal{W}_{\mathcal{I}1}$, $\mathcal{W}_{\mathcal{I}2}$, $\mathcal{W}_{\mathcal{I}3}$, $\mathcal{W}_{\mathcal{I}4}$, $b_{\mathcal{I}1}$, and $b_{\mathcal{I}3}$ are parameters. Hidden state vector of the m-th node is denoted as:

$$\mathcal{H}_m^\tau(u_i) = \tilde{\mathcal{H}}_m^\tau(u_i) \odot \mathcal{I}_m^{\tau-UG}(u_i) + \mathcal{H}_{m-1}^\tau(u_i) \odot \left[1 - \mathcal{I}_m^{\tau-UG}(u_i)\right] \tag{11}$$

where $\odot$ denotes element-wise multiplication, and $\tilde{\mathcal{H}}_m^\tau(u_i)$ is calculated as:

$$\tilde{\mathcal{H}}_m^\tau(u_i) = \sigma_2\left\{\mathcal{W}_{\mathcal{I}5} \cdot \mathcal{U}_m^\tau(u_i) + \mathcal{W}_{\mathcal{I}6} \cdot \left[\mathcal{I}_m^{\tau-RG}(u_i) \odot \mathcal{H}_{m-1}^\tau(u_i)\right] + b_{\mathcal{I}5}\right\} \tag{12}$$

where $\mathcal{W}_{\mathcal{I}5}$, $\mathcal{W}_{\mathcal{I}6}$ and $b_{\mathcal{I}5}$ are parameters, and $\sigma_2(\cdot)$ is the tanh activation function which is denoted as:

$$\sigma_2(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{13}$$

Initial state of $\mathcal{H}_m^\tau(u_i)$ is a $(|u|-1)$-dimensional vector, in which each element represents social relation status between user u_i and other $(|u|-1)$ users. It can be obtained from the node sequence $\mathcal{B}_m^\tau(u_i)$. The $(\mathcal{M}-1)$-dimensional vector $\mathcal{B}_m^\tau(u_i)$ indicates that user u_i have have occasional relations with $(\mathcal{M}-1)$ users. The corresponding $(\mathcal{M}-1)$ elements in $\mathcal{H}_m^\tau(u_i)$ are set to 1, and other elements in $\mathcal{H}_m^\tau(u_i)$ are set to 0.

After $\mathcal{M}$ rounds of propagations, the output hidden state vector of τ-th sampling round is $\mathcal{H}_{\mathcal{M}}^\tau(u_i)$. Outputs of all the q sampling rounds are aggregated into a final representative vector to denote occasional relation factors of user u_i. The aggregation process is implemented through an attentive neural mapping procedure, which is expressed as:

$$\mathcal{R}_{ui}^{(occ)} = \sigma_1\left\{\frac{1}{q}\sum_{\tau=1}^{q} a_1^\tau \cdot \left[\mathcal{W}_{\mathcal{I}7} \cdot \mathcal{H}_{\mathcal{M}}^\tau(u_i) + b_{\mathcal{I}7}\right]\right\} \tag{14}$$

where $\mathcal{W}_{\mathcal{I}7}$ and $b_{\mathcal{I}7}$ are parameters, and a_1^τ is the attention weight of the τ-th sampling round. Similarly, taking post p_j as the starting node, representative vector for occasional relations between post p_j and other posts can be obtained as:

$$\mathcal{R}_j = \sigma_1\left\{\frac{1}{q}\sum_{\tau=1}^{q} a_2^\tau \cdot \left[\mathcal{W}_{\mathcal{I}7} \cdot \mathcal{H}_{\mathcal{M}}^\tau(p_j) + b_{\mathcal{I}8}\right]\right\} \tag{15}$$

where $\mathcal{W}_{\mathcal{I}7}$ and $b_{\mathcal{I}8}$ are parameters, and a_2^τ is the attention weight of the τ-th sampling round. Note that $\delta(\cdot)$ in Eq. (7) and (8) for posts is realized by counting interactive users related to them. To sum up, inference of representative vectors for occasional relations not only extends the modeling perspective into heterogeneous cyberspace, but also enriches feature spaces.

3.2. Deep Graph Neural Network

To sufficiently mine interaction characteristics between users and posts, each interaction between a user and a post is assumed to generate a contribution value for a user. A higher contribution value indicates larger possibility that the user is a spammer. Contribution values of all his interaction record jointly determine his nature. All the interactions can be viewed as a graph network, in which interactions are vertices and relations among interactions are edges. Therefore, a graph neural network framework can be developed for this purpose.

Representative vector for vertex feature of an interaction is highly correlated to three aspects of factors:

1) Attributes of the user. Following the idea in one of our previously published research, attributes are divided into two types: numerical attributes and categorical attributes. The former refer to those whose contents are numerical values, and can be directly submitted into models for computation, without encoding procedures. The latter refer to those whose contents are a fixed value out of multiple optional items, such as sex, location, etc. The one-hot encoding (OHE) can be utilized to encode them into a feature vector. In rules of OHE, dimension of a feature vector corresponds to the number of all the optional items. Inside the feature vector, the element corresponding to the hit item from multiple optional items is set to 1, and other elements are set to 0. Encoding results of two types of features are aggregated into a whole feature vector to denote feature encoding of users. For the user u_i, two types of feature vectors of him are denoted as $\mathcal{C}_i^{(nu)}$ and $\mathcal{C}_i^{(ca)}$.

2) Semantics of the post. A post p_j is actually a sentence. Before modeling semantics of posts, all the words ever occurring in posts are collected into a dictionary. In the dictionary, each word is randomly assigned a number to reflect its ranking position. In post p_j, each word is transformed into a vector via the one-hot encoding scheme. Dimension of each word vector equals to size of the dictionary, and each element of it corresponds to a word in dictionary. In a word vector, the element corresponding to position of the word is set to 1, and the other elements are all set to 0. As for a sentence with $\mathcal{Z}$ words, all the word vectors are denoted as $w_z \, (z = 1, 2, \cdots, \mathcal{Z})$, where z is the index number of words in the sentence and ranges from 1 to $\mathcal{Z}$. Then, a bi-directional attention encoding structure is introduced to model the word sequence from two directions:

$$\left[h_{j,z}^{(for)}\right]^T = \sigma_1 \left\{\mathcal{W}_{h1} \cdot \left[w_z^i \oplus h_{j,z-1}^{(for)}\right]^T + b_{h1}\right\} \tag{16}$$

$$\left[h_{j,z}^{(bac)}\right]^T = \sigma_1 \left\{\mathcal{W}_{h2} \cdot \left[w_z^i \oplus h_{j,z+1}^{(bac)}\right]^T + b_{h2}\right\} \tag{17}$$

where $\oplus$ denotes concatenation operation, $h_{j,z}^{(for)}$ and $h_{j,z}^{(bac)}$ separately corresponds to representative vectors in forward and backward directions, and $\mathcal{W}_{h1}$, $\mathcal{W}_{h2}$, b_{h1} and b_{h2} are parameters. Note that $h_{j,z-1}^{(for)}$ and $h_{j,z+1}^{(bac)}$ are identity vectors in the cases where $z = 1$ and $z = \mathcal{Z}$. Therefore, the final representative vector for semantics of post p_j is calculated as:

$$\mathcal{C}_j^{(se)} = \sigma_1 \left\{\mathcal{W}_{h3} \cdot \left[\frac{\lambda_1}{\mathcal{Z}} \sum_{z=1}^{\mathcal{Z}} h_{j,z}^{(for)} \cdot r_1^T + \frac{(1-\lambda_1)}{\mathcal{Z}} \sum_{z=\mathcal{Z}}^{1} h_{j,z}^{(bac)} \cdot r_2^T\right] + b_{h3}\right\} \tag{18}$$

where r_1^T and r_2^T are transition vectors, and $\mathcal{W}_{h3}$ and b_{h3} are parameters.

As for the interaction between user u_i and post p_j, representative vector for its vertex feature is obtained by concatenating $\mathcal{C}_i^{(nu)}$, $\mathcal{C}_i^{(ca)}$ and $\mathcal{C}_j^{(se)}$:

$$\mathcal{C}_{i,j} = \left[\mathcal{C}_i^{(nu)} \oplus \mathcal{C}_i^{(ca)} \oplus \mathcal{C}_j^{(se)}\right] \tag{19}$$

Representative vector for edge features of an interaction is composed of three components: inherent relations among users, potential relations among users, and relations among posts. As the last two parts have been inferred before, modeling of inherent relations among users is given here. It is derived from inherent social relations among users which are denoted as:

$$\mathcal{S}_i = \left[s_1, s_2, \cdots, s_{i-1}, s_{i+1}, \cdots, s_{|u|}\right] \tag{20}$$

Note that $\mathcal{S}_i$ is a $(|u-1|)$-dimensional vector that indicates social relations of the user u_i. It needs to be mapped into a more abstract representative vector through the following operation:

$$\mathcal{R}_{ui}^{(inh)} = \sigma_1 \left[\mathcal{W}_{\mathcal{C}} \cdot \left(\mathcal{S}_i^T \cdot \mathcal{S}_i\right) + b_{\mathcal{C}}\right] \tag{21}$$

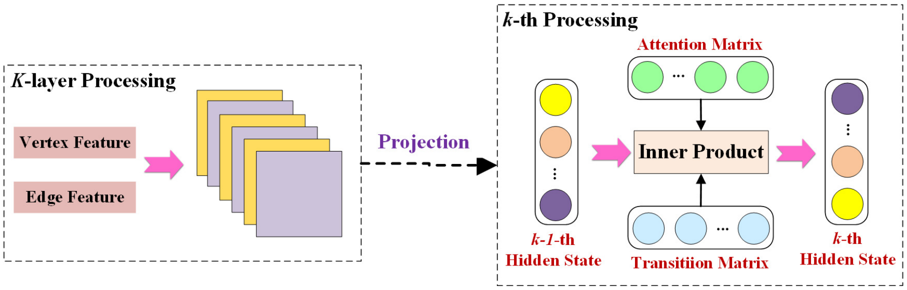


Figure 4: Workflow for Deep Graph Neural Network.

where $\mathcal{W}_{\mathcal{C}}$ and $b_{\mathcal{C}}$ are parameters. Therefore, representative vector for edge features of an interaction is obtained by concatenating $\mathcal{R}_{ui}^{(inh)}$, $\mathcal{R}_{ui}^{(occ)}$, and $\mathcal{R}_j$:

$$\mathcal{R}_{i,j} = \left[\mathcal{R}_{ui}^{(inh)} \oplus \mathcal{R}_{ui}^{(occ)} \oplus \mathcal{R}_j\right] \tag{22}$$

Having encoded $\mathcal{C}_{i,j}$ and $\mathcal{R}_{i,j}$, as is shown in Figure 4, a graph neural network structure with multiple processing layers is designed to further map it into higher-dimensional feature spaces. Index number of layers is denoted as k which ranges from 1 to $\mathcal{K}$. Such a multi-layer structure is implemented through a series of propagation procedures that transit from a layer to the next layer. In the k-th layer, hidden state of it is deduced as:

$$\mathcal{H}_{i,j}^{(k)} = \sigma_1 \left\{\mathcal{A}_{i,j}^{(k)} \cdot \mathcal{H}_{i,j}^{(k-1)} \cdot \mathcal{E}_{i,j}^{(k)}\right\} \tag{23}$$

where $\mathcal{A}_{i,j}^{(k)}$ is the attention matrix, and $\mathcal{E}_{i,j}^{(k)}$ is the transition matrix from the $(k-1)$-th layer to the k-th layer. In the initial layer, $\mathcal{H}_{i,j}^{(k-1)}$ is obtained from $\mathcal{C}_{i,j}$ and $\mathcal{R}_{i,j}$. The multi-layer perception (MLP) network is firstly introduced to map them into two abstract matrices:

$$\mathcal{X}_{i,j} = \alpha_1^T \cdot MLP_1\left(\mathcal{C}_{i,j}\right) \tag{24}$$

$$\mathcal{Y}_{i,j} = \alpha_2^T \cdot MLP_2\left(\mathcal{R}_{i,j}\right) \tag{25}$$

In the initial layer, $\mathcal{H}_{i,j}^{(k-1)}$ is obtained as:

$$\mathcal{H}_{i,j}^{(0)} = \sigma_1 \left\{\mathcal{W}_{\mathcal{H}1} \cdot \left[\mathcal{X}_{i,j} \oplus \mathcal{Y}_{i,j}\right] + b_{\mathcal{H}1}\right\} \tag{26}$$

185 where $\mathcal{W}_{\mathcal{H}1}$ and $b_{\mathcal{H}1}$ are parameters.

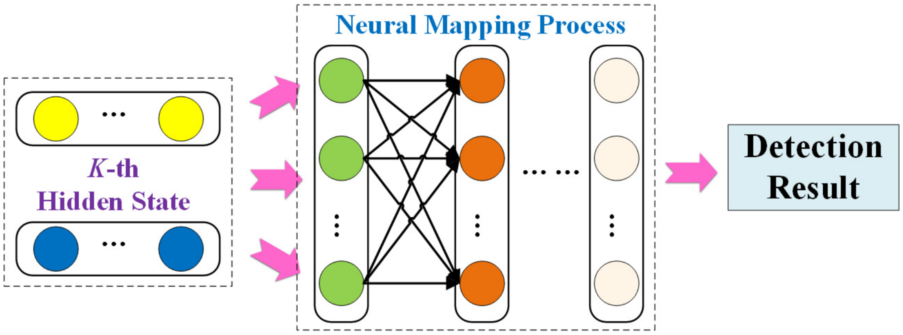


Figure 5: Generation of Detection Results.

3.3. Training and Optimization

As is shown in Figure 5, All the $\mathcal{K}$ processing layers are followed by a fully connected neural mapping function to generate an intermediate discrimination vector $\mathcal{D}_i$. The process is represented as:

$$\mathcal{D}_i = \sigma_1 \left\{ \frac{1}{|p|} \sum_{j=1}^{|p|} d_{i,j} \cdot \beta_{i,j} \cdot \left[\mathcal{W}_{\mathcal{H}2} \cdot \mathcal{H}_{i,j}^{(\mathcal{K})} + b_{\mathcal{H}2} \right] \right\} \tag{27}$$

where $\mathcal{W}_{\mathcal{H}2}$ and $b_{\mathcal{H}2}$ are parameters, $\beta_{i,j}$ is the attention vector for posts, and $d_{i,j}$ is a response function that denotes interaction status between user u_i and post p_j. The $d_{i,j}$ equals to 1 if the interaction exists, and equals to 0 otherwise. The final detection result for user u_i is calculated as:

$$\hat{y}_i = \sigma_3 \left[\mathcal{D}_i \cdot (\mathcal{D}_i)^T \right] \tag{28}$$

where $\sigma_3(\cdot)$ is the sigmoid activation function denoted as:

$$\sigma_3(x) = \frac{1}{1+e^{-x}} \tag{29}$$

Obviously, its effect is to limit the range of $\hat{y}_i$ as $(0,1)$, so that binary classification can be realized. Note that a higher value of $\hat{y}_i$ indicates that user u_i is more likely to be a spammer.

In summary, objective function of the rumor detection problem can be formulated as the following formula:

$$\min \left\{ \sum_{i=1}^{|u|} \left[-y_i \log \hat{y}_i - (1-y_i) \log (1-\hat{y}) + \lambda_2 \|\Theta_i\|_F^2 \right] \right\} \tag{30}$$

where $\|\cdot\|_F^2$ denotes the $L2$ regularization item, Θ_i denotes the set of parameters related to user u_i, and y_i is the true nature value of user u_i. The y_i is expressed as:

$$y_i = \begin{cases} 1, & \text{user } u_i \text{ is a spammer} \\ 0, & \text{user } u_i \text{ is not a spammer} \end{cases} \tag{31}$$

At last, the Adam optimizer [41] is employed to solve the above optimization problem. After all the parameters have been learned, detection results for unknown users can be directly calculated.

4. Experiments and Analysis

The proposed DeG-Spam is evaluated on two real-world benchmark datasets. The first subsection gives descriptions about datasets and main features. Then, experimental settings, evaluation metrics and benchmark methods are described in the second subsection. And the third subsection demonstrates experimental results through visualized tables and figures and makes corresponding discussions.

4.1. Datasets

Almost all of the prevalent datasets for experimental evaluation in the field of spammer detection came from two source platform: Twitter [1] and Sina Weibo [2]. And the two datasets utilized in this research were separately collected from the two platforms. Accordingly, the two datasets are named Twitter dataset and Weibo dataset, and their information are briefly introduced as follows:

Twitter dataset. This version of the Twitter dataset was firstly published by Yang et al. [42] in the year of 2012. They used official application program interactions (API) of Twitter to crawl source data from the platform. The initial dataset possesses about 10000 users, and 2060 of them are annotated as spammers after careful assessment.

[1] https://twitter.com/

[2] https://weibo.com/

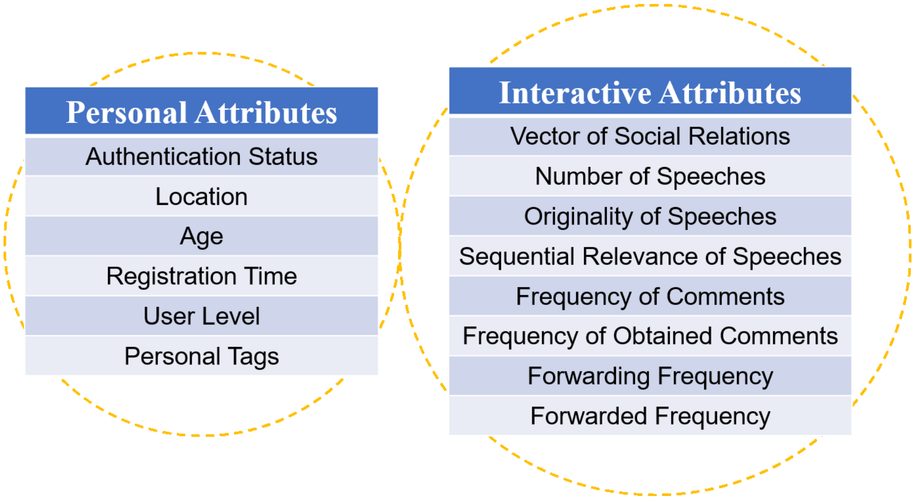


Figure 6: Details of Two Types of Attributes.

Weibo dataset. This version of Weibo dataset has been utilized in one of our previously published study [1]. In the year of 2019, it was collected by our working team with the aid of the official API. And five graduate students were provisionally recruited to annotate all the users according to their expertise experience. After the assessment, 1158 users were labelled as spammers out of totally 6072 users.

To construct a heterogeneous information network, all users are randomly assigned communities they belong to. The number of communities of each user is set as the range of 1 to 5, meaning that each user joins at least one community and at most five communities. Besides, for each piece of text associated with a tweet or a microblog, it is expected to extract a topic indicator for it with the utilization of Twitter-LDA algorithm [43]. As for Chinese texts of microblogs, they need to be translated into English texts by invoking API of Baidu Translation [3]. Although each user has been assigned a label in initial dataset, the proportion of spammers in all the users is quite small. To ensure the distribution balance of two classes of data, only some of the normal users

[3] https://api.fanyi.baidu.com/

are selected to make the number of them are close to the number of spammers. Here, the sampling operation is implemented through random sampling. After processing, the Twitter dataset contains 2080 normal users and 2000 spammers, and the Weibo dataset contains 1150 normal users and 1100 spammers.

As is shown in Figure 6, attributes of users can be classified into two types: personal attributes and interactive attributes. The former refers to attributes of users themselves, including authentication status, location, age, registration time, user level, personal tags. The latter reflects interactive characteristics between users and posts, including the vector of social relations, number of speeches, originality of speeches, sequential relevance of speeches, frequency of comments, etc. From the view of data form, these attributes can be categorized into two types: numerical attributes and categorical attributes. Note that all of them can be encoded into vectors.

4.2. Experimental Settings

To prove the superiority of the proposed DeG-Spam, some typical spammer detection methods that can represent the whole technology level need to be selected as baselines. According to the novelty mentioned in Section 1, five relative methods are utilized as baselines here. On the one hand, we construct two semantic analysis-based methods which are respectively named "LDA+K-means" and "LSTM+LoR" for short. On the other hand, we introduce three typical methods that exploit various contextual information to realize detection. The three methods are named "SVM", "CNN" and "NMF" for short. All of the five benchmark methods are briefly described as follows:

LDA+K-means. It is actually the combination of two unsupervised learning-based methods: latent Dirichlet allocation (LDA) [44] and K-means clustering [45]. The former part is to extract semantic features for posts and the latter part is to classify samples. Hence, the LDA+K-Means is an unsupervised detection method. More descriptions of the method are introduced in [20].

LSTM+LR. It is the combination of two typical methods: long short-term memory (LSTM) model [46] and logistic regression (LoR) model [47]. The LSTM is a sequential modelling method, and its role is to extract semantic features for posts. The LoR is a classical classification model that can be used for classification. Hence, the LSTM+LR is a supervised detection method.

SVM. The core of this model is the typical classification model: support vector machine (SVM) [48]. Inside the method, contextual information such as user attributes and social relations is encoded into vectors as DeG-Spam does. And SVM model is utilized to identify the nature of users.

MLP. The core of this model is an elementary neural network model named multi-layer perception (MLP) [49]. Inside the method, contextual information such as user attributes and social relations is encoded into vectors as DeG-Spam does. It directly carries out neural mapping after feature extraction and abstraction.

CNN. The core of this model is a typical neural network model named convolutional neural network (CNN) [50]. Encoding of contextual information is similar to the above two methods. The CNN and MLP are both deep feature representation-based methods. Differently, the CNN carries out multiple layers of convolutional operations after feature extraction.

To discriminate whether a user is a normal user or a spammer is essentially a binary classification problem, in which a sample with label 1 is a positive sample and a sample with label 0 is a negative sample. In this research, a positive sample indicates that the user is a spammer, while a negative sample indicates that the user is normal. In general, it is not reasonable to directly assess accuracy of classification problems. Because for samples with different labels, the cost of error prediction is different. Thus, it is supposed to assess classification effect for positive samples and negative samples, separately. The concept of true positive (TP) and false positive (FP) indicate scenarios where positive samples are correctly identified and incorrectly identified, respectively. Similarly, the concept of true negative (TN) and false negative (FN) indicate scenarios where negative samples are correctly identified and incorrectly identified, respectively. given above definitions, four metrics that are used for evaluating performance of the proposed DeG-Spam and baselines, can be deduced: precision, recall, accuracy, and F-score. And computational expressions of the four metrics are listed as follows:

$$Precision = \frac{\zeta\left(TP\right)}{\zeta\left(TP\right) + \zeta\left(FP\right)} \tag{32}$$

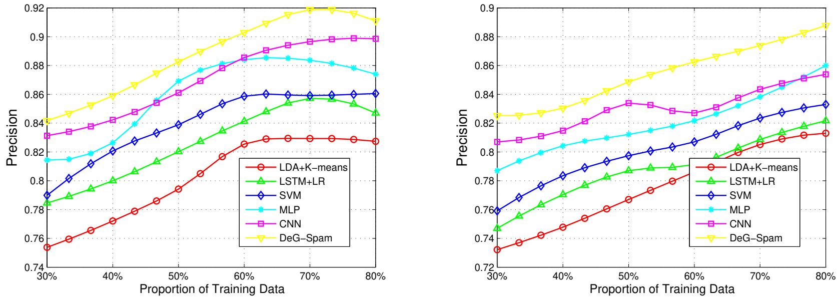
$$Recall = \frac{\zeta\left(TP\right)}{\zeta\left(TP\right) + \zeta\left(FN\right)} \tag{33}$$

$$Accuracy = \frac{\zeta\left(TP\right) + \zeta\left(TN\right)}{\zeta\left(TP\right) + \zeta\left(FP\right) + \zeta\left(TN\right) + \zeta\left(FN\right)} \tag{34}$$

$$F - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{35}$$

where $\zeta\left(x\right)$ counts the number of x.

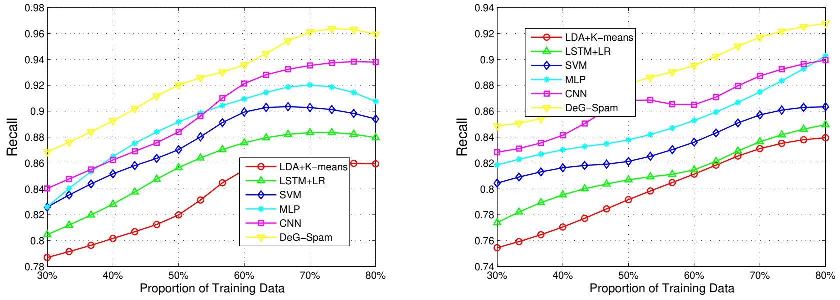
The proposed DeG-Spam is implemented with the assistance of the tool TensorFlow [4], and its running environment is a working station with 28-core CPU and a GPU (RTX-2080Ti). As for parameters, the number of sampling rounds q is set to 15, the number of nodes $\mathcal{N}$ in each round is set to 10, the number of nodes $\mathcal{M}$ in a homogeneous walking path is set to 5, and the number of convolutional processing layers $\mathcal{K}$ is set to 8. The learning rate of DeG-Spam is ordinarily set to 0.001 and will be changed multiple times during experiments. Parameters in baselines are expected to be set as the default values, the detailed settings are left out here due to the limitation of textual length. The proportion of training data is set to 60% in default, and it is tuned according to real experimental situations.



(a) Twitter

(b) Weibo

Figure 7: Precision results on two datasets



(a) Twitter

(b) Weibo

Figure 8: Recall results on two datasets

4.3. Results and Analysis

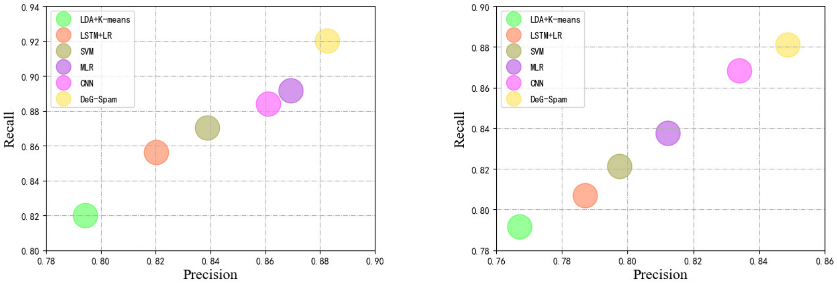
During the process of experiments, both of the datasets are divided into training part and testing part. All the baselines and the DeG-Spam are trained with the utilization of training data and their performance is assessed on the 290 testing data in terms of four aforementioned metrics. The first group of experiments evaluate the superiority of the DeG-Spam compared with baselines, and the other group of experiments evaluate the sensitivity of the DeG-Spam

[4] https://tensorflow.google.cn/

itself. Two groups of experiments collaboratively assess the performance of the DeG-Spam.

Precision results and recall results are reflected in Figure 7 and Figure 8, respectively. Each figure has two subfigures which correspond to results on two datasets. In each subfigure, X-axis denotes the proportion of training data ranging from 30% to 80%, and Y-axis denotes values of evaluation metrics obtained by baselines and DeG-Spam. Macroscopically, it can be easily observed that experimental results for almost all of the methods get better while the proportions of training data increase. When 70% is reached by the proportion, the ascending tendency of performance tends to be gentle. Especially on the Twitter dataset, experimental results get worse to some extent when such proportion switches from 70% to 80%. Thus, 70% is a proper value for the proportion of training data, as ideal experimental results can be acquired. Microscopically, two semantics modelling-based methods are weaker than other methods and followed by the SVM. The two deep learning-based benchmark methods are better than other baselines. But the proposed DeG-Spam always obtains the best performance compared to others. As for precision results, the DeG-Spam is about 5% better than CNN, 6% better than MLP, 8% better than SVM, 10 % better than the two semantic modelling-based methods. As for recall results, the DeG-Spam is about 6-7 % better than CNN and MLP, 9% better than SVM, and 10-11% better than two semantic modelling-based methods.

To explore the integrated effect of precision results and recall results, the group of scatter plots are introduced. The proportion of training data is set to three typical values: 50%, 60% and 70%. Corresponding scatter plots concerning two datasets are illustrated in Figure 9 , Figure 10 and Figure 11. Each figure has two subfigures that correspond to results on two datasets. Inside each subfigure, X-axis denotes value range of precision results and Y-axis denotes value range of recall results. A scatter denotes a group of precision value and recall value of a method, and its location indicates performance. In other words, a larger distance between the origin and a scatter indicates that the corresponding method possesses better performance. It can be observed from



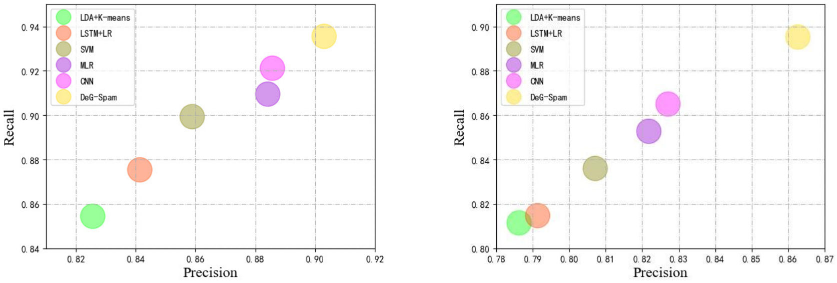
(a) Twitter

(b) Weibo

Figure 9: Joint Effect of Precision Results and Recall results When Proportion of Training Data is Set to 50%

these six subfigures that the scatter corresponding to DeG-Spam is always the farthest from the origin. No matter how the scenarios change, its distance from the origin is always larger than others. As for scattering of DeG-Spam, its distance to the origin is about twice than that of MLP, and is about three times than that of LSTM+LR. Therefore, the performance of DeG-Spam remarkably exceeds baselines from the visualization of scatter plots.

It can be concluded from above subfigures that the proposed DeG-Spam always performs better than benchmark methods under the measurement of precision and recall, regardless of any proportions of training data. The obtainment of the above results can be attributed as two aspects of reasons. Firstly, it explores hidden relations to enhance feature spaces, which is the main difference from previous methods. Secondly, it leverages a deep graph neural network framework to model interaction characteristics between users and posts, so that more precise detection can be realized. And it is also noticed that the DeG-Spam and benchmark methods can achieve better performance on Twitter dataset than on Weibo dataset. This phenomenon may be caused by the gap between Chinese texts and English texts. This is because the proposed DeG-Spam is developed with oriented interactions with English texts. The Chinese texts of Weibo dataset are required to be processed into English texts, which



(a) Twitter

(b) Weibo

Figure 10: Joint Effect of Precision Results and Recall results When Proportion of Training Data is Set to 60%

may inevitably bring about some error. Although semantic modelling for texts is just a small part in models, the influence still exists. In summary, because of its unique features, the proposed DeG-Spam is superior to benchmark methods from the view of precision and recall results.

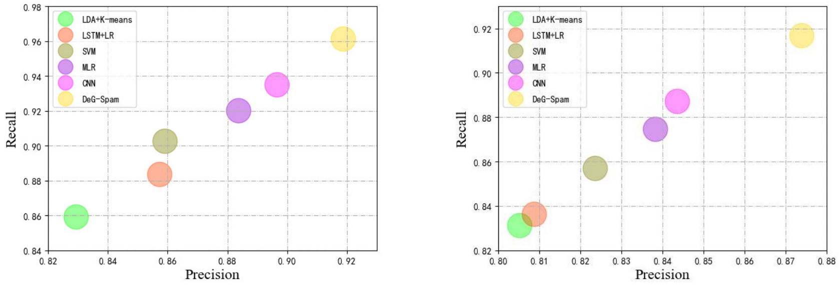
Besides precision results and recall results, another evaluation metric named F-score is introduced for further assessment. According to Eq. (35), it is constructed with the aid of precision and recall and can be viewed as an overall fusion of the two. F-score results on two datasets are illustrated in Table 1 and Table 2. Each table contains six columns and six lines, in which the first column lists the six methods and the first line lists six proportions of training data. Similar to precision results and recall results, three deep learning-based methods are better than the other three. Apart from the above three metrics, we also leverage another typical metric named accuracy. Accuracy results on two datasets under different proportions of training data are illustrated in Figure 12. It is composed of six subfigures which correspond to six proportions of training data: 30%, 40%, 50%, 60%, 70% and 80%. Among, each subfigure possesses two clusters of values, which corresponds to results on Twitter dataset and Weibo dataset separately. Of all the six methods, it can be easily observed from these subfigures that three deep learning-based methods universally perform better

Table 1: F-score Results on Twitter Dataset

Algorithms	Different Sizes of Training Data					
	30%	40%	50%	60%	70%	80%
LDA+K-means	0.7701	0.7866	0.8068	0.8398	0.8440	0.8430
LSTM+LR	0.7945	0.8138	0.8379	0.8581	0.8702	0.8630
SVM	0.8075	0.8359	0.8543	0.8786	0.8804	0.8770
MLP	0.8202	0.8453	0.8804	0.8966	0.9016	0.8905
CNN	0.8357	0.8522	0.8724	0.9032	0.9156	0.9178
Deg-Spam	**0.8549**	**0.8755**	**0.9011**	**0.9190**	**0.9395**	**0.9346**

Table 2: F-score Results on Weibo Dataset

Algorithms	Different Sizes of Training Data					
	30%	40%	50%	60%	70%	80%
LDA+K-means	0.7432	0.7590	0.7791	0.7985	0.8180	0.8261
LSTM+LR	0.7602	0.7827	0.7969	0.8028	0.8223	0.8354
SVM	0.7811	0.7995	0.8091	0.8213	0.8400	0.8479
MLP	0.8024	0.8171	0.8248	0.8370	0.8561	0.8808
CNN	0.8175	0.8279	0.8508	0.8455	0.8648	0.8761
DeG-Spam	**0.8367**	**0.8445**	**0.8646**	**0.8786**	**0.8949**	**0.9074**



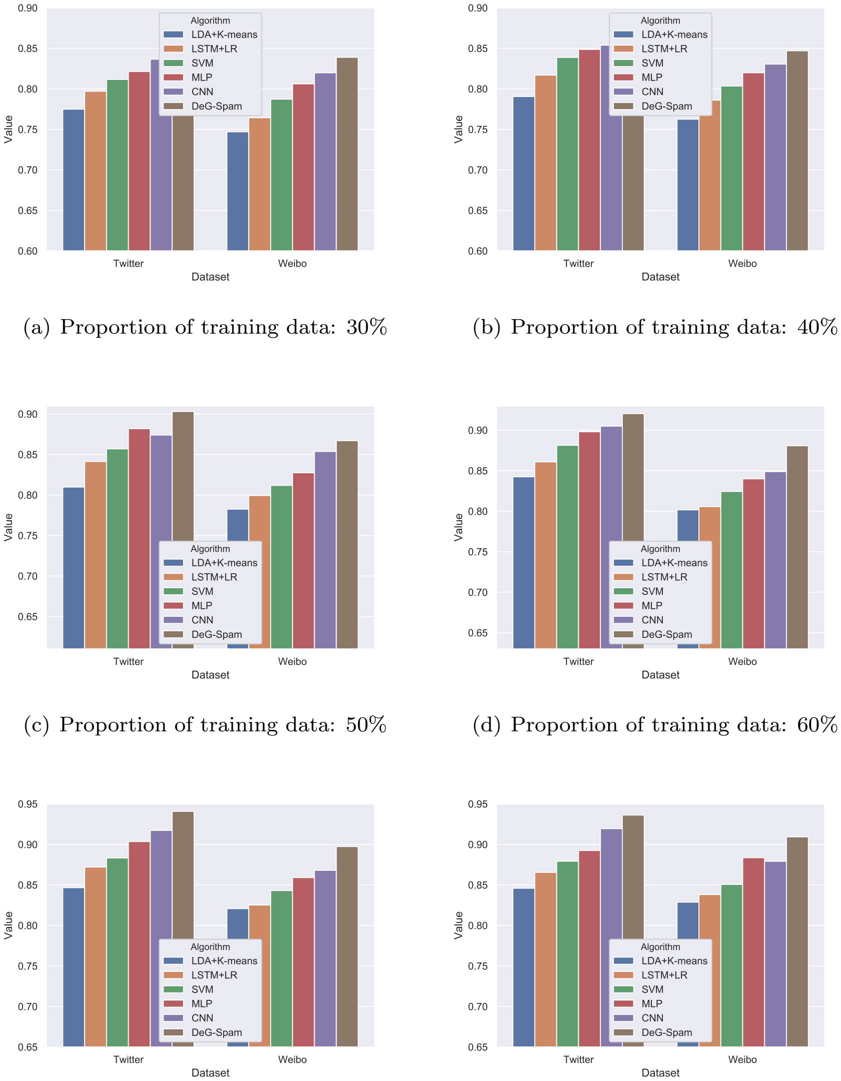
(a) Twitter

(b) Weibo

Figure 11: Joint Effect of Precision Results and Recall results When Proportion of Training Data is Set to 70%

than the other three. The reason lies in the fact that the three methods exploit the idea of deep feature representation which can extract feature components with more representative ability. Without deep feature abstraction, the other three methods cannot acquire relatively good experimental results. Of CNN and MLP, the CNN is better than MLP in most of the cases but is not better than MLP in a few cases.

It can be observed from the aforementioned tables and figure that the proposed DeG-Spam always performs better than benchmark methods under the measurement of F-score and accuracy. Two aspects of reasons can be deduced to explain the observed phenomenon. Firstly, the proposed DeG-Spam extends feature spaces by investigating the utilization of occasional relations which are usually ignored by previous methods. To this end, more fine-grained feature spaces can be obtained. Secondly, deep representation has been proved effective in improvement of spammer detection. The DeG-Spam develops a deep graph neural network to deeply encode internal features of social networks. Thus, more robust feature components can be extracted to express interaction characteristics. Combined with the above reasons, this group of experiments also well prove that the proposed DeG-Spam can promote detection efficiency. And the above two groups of experiments jointly reveal that the proposed DeG-Spam is



(a) Proportion of training data: 30%

(b) Proportion of training data: 40%

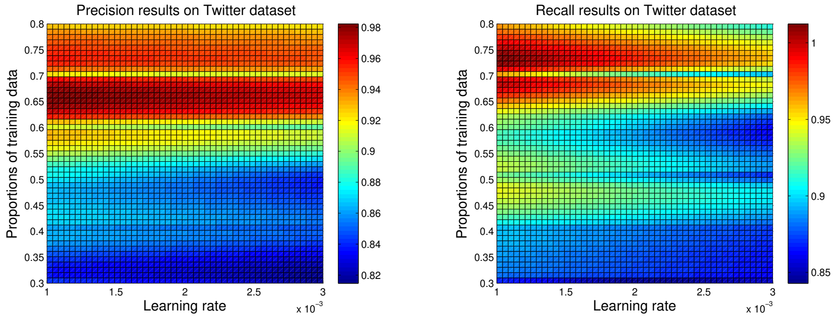
(c) Proportion of training data: 50%

(d) Proportion of training data: 60%

(e) Proportion of training data: 70%

(f) Proportion of training data: 80%

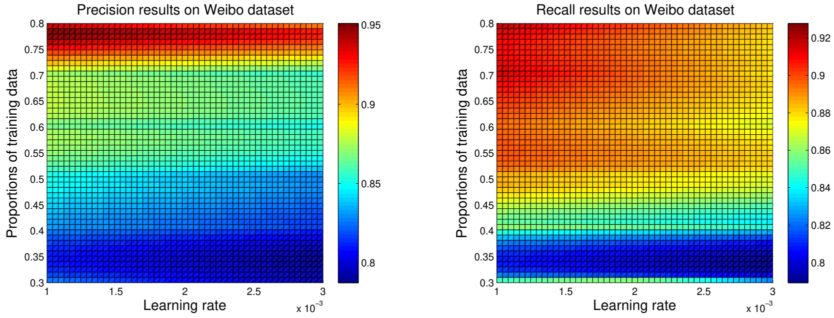
Figure 12: Accuracy results on two datasets Under Different Proportions of Training Data



(a) Precision Results

(b) Recall Results

Figure 13: Parameter Sensitivity Results on Twitter Dataset



(a) Precision Results

(b) Recall Results

Figure 14: Parameter Sensitivity Results on Twitter Dataset

well suitable for spammer detection problem.

Also, another group of experiments are further conducted to testify parameter sensitivity of the proposed Co-Spam. Specifically, we visualize the evolving tendency of performance with the simultaneous change of two parameters: proportions of training data and learning rate of Adam optimizer. Among, the learning rate is set to three different values: 0.001, 0.002, and 0.003, and the proportion of training data is set to six values: 30%, 40%, 50%, 60%, 70% and 80%. According to the value ranges of two variables, eighteen value combinations can be deduced. This group of experiments manage to demonstrate the fluctuation tendency of performance under different value combinations of those two variables. The parameter sensitivity results on two datasets are illustrated in Figure 13 and Figure 14 which are exactly four heat maps. Each figure contains two subfigures which are two heat maps. In each subfigure, the X-axis lists value range of learning rate, and the Y-axis lists value range of proportions of training data. Obviously, a smaller colour difference indicates better stability because it is not susceptible to parameter changes. As for Twitter dataset, the experimental results can be divided into two scenarios where the proportion of training data is higher or lower than 60%. Inside both scenarios, the performance remains relatively stable. And for Weibo dataset, precision results and recall results go to two different extremes. Precision results are universally small and recall results are universally large. Despite this, they still show relatively stable fluctuation status, as the color difference in them is not large. Three possible explanations can be deduced for the above results. Firstly, the introduction of occasional relations enhance feature spaces to some extent, improving the robustness of detection methods. Secondly, a developed graph neural network framework can excellently capture various relations inside interactions. With abstract relations being better represented, the proposed DeG-Spam is well suitable for different scenarios. Thirdly, the parameters inside the model are properly settled, which is also able to promote stability. To sum up, the proposed DeG-Spam is not susceptible to parameter changes and possesses considerable stability.

In a word, above several groups of experiments not only verify detection the efficiency of the proposed DeG-Spam, but also proves that it is a relatively the stable model that can be used for complicated and changeable situations.

5. Conclusions

Nowadays, more and more researchers manage to improve spammer detection efficiency by modelling various relational information inside feature spaces. Almost all of the existing methods resort to stable or explicit relations, yet ignoring relations that are implicit or generated provisionally. Such type of relations exist in many real cases and have a considerable effect on spammer detection. To overcome the current challenge, this paper proposes a two-stage method named DeG-Spam. First of all, it infers occasional relations and formulates feature expressions for them. On this foundation, a graph neural network framework is developed to model comprehensive feature spaces of interactions. Thus, a deep graph neural network model can be established for spammer detection. Such a two-stage model is able to produce more fruitful and robust feature spaces by strengthening feature expressions. To evaluate the performance of the proposed DeG-Spam, a set of experiments are carried out on two real-world datasets to compare the DeG-Spam with several baseline methods. Experimental results show that it performs about 5%-10% better than baselines. The proposed method manages to detect organized crime groups and malicious actors from a novel perspective: heterogeneous cyberspace. It certainly has a positive effect on law enforcement and cooperates forensics. Besides, this research work is also instructive to the domain of social network forensics which is a novel concern around the world.

Acknowledgement

This work was supported in part by Chongqing Natural Science Foundation of China under grant cstc2019jcyj-msxmX0747, in part by the State Language Commission Research Program of China under grant YB135-121, in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJQN202000805, in part by Japan Society for the Promotion of Science (JSPS) Grants-in-Aid for Scientific Research (KAKENHI) under Grant JP18K18044, and in part by the Key Research Project of Chongqing Technology and Business University under grant ZDPTTD201917 and 1952027.

References

References

[1] Z. Guo, Y. Shen, A. K. Bashir, M. Imran, N. Kumar, D. Zhang, K. Yu, Robust spammer detection using collaborative neural network in internet of thing applications, IEEE Internet of Things Journal (2020). doi:10.1109/JIOT.2020.3003802.

[2] X. Zhou, Y. Hu, W. Liang, J. Ma, Q. Jin, Variational lstm enhanced anomaly detection for industrial big data, IEEE Transactions on Industrial Informatics (2020). doi:10.1109/TII.2020.3022432.

[3] Z. Guo, H. Wang, A deep graph neural network-based mechanism for social recommendations, IEEE Transactions on Industrial Informatics (2020). doi:10.1109/TII.2020.2986316.

[4] A. Makkar, N. Kumar, Cognitive spammer: A framework for pagerank analysis with split by over-sampling and train by under-fitting, Future Gener. Comput. Syst. 90 (2019) 381–404. doi:10.1016/j.future.2018.07.046.

[5] A. Shalaginov, Advancing neuro-fuzzy algorithm for automated classification in largescale forensic and cybercrime investigations: Adaptive machine learning for big data forensic, Ph.D. thesis, Norwegian University of Science and Technology (2018).

[6] K. N. Tran, M. Alazab, R. Broadhurst, Towards a feature rich model for predicting spam emails containing malicious attachments and urls, in: Australasian Data Mining Conference, Australian Computer Society Inc., 2014.
URL https://openresearch-repository.anu.edu.au/bitstream/1885/28534/2/01_Tran_Towards_a_Feature_Rich_Model_2014.pdf

[7] X. Zhou, W. Liang, K. Wang, H. Wang, L. T. Yang, Q. Jin, Deep learning enhanced human activity recognition for internet of healthcare

things, IEEE Internet of Things Journal 7 (7) (2020) 6429–6438. doi:10.1109/JIOT.2020.3003802.

[8] C. Benzaid, K. Lounis, A. Al-Nemrat, N. Badache, M. Alazab, Fast authentication in wireless sensor networks, Future Gener. Comput. Syst. 55 (2016) 362–375. doi:10.1016/j.future.2014.07.006.

[9] A. Azab, R. Layton, M. Alazab, J. Oliver, Mining malware to detect variants, in: 2014 Fifth Cybercrime and Trustworthy Computing Conference, 2014, pp. 44–53. doi:10.1109/CTC.2014.11.

[10] S. Bhattacharya, S. R. K. S, P. K. R. Maddikunta, R. Kaluri, S. Singh, M. Alazab, U. Tariq, A novel pca-firefly based xgboost classification model for intrusion detection in networks using gpu, Electronics 9 (2020) 219–234. doi:10.3390/electronics9020219.

[11] X. Zhou, Y. Li, W. Liang, Cnn-rnn based intelligent recommendation for online medical pre-diagnosis support, IEEE/ACM Transactions on Computational Biology and Bioinformatics (2018). doi:10.1109/TCBB.2020.2994780.

[12] X. Zhou, W. Liang, K. Wang, R. Huang, Q. Jin, Academic influence aware and multidimensional network analysis for research collaboration navigation based on scholarly big data, IEEE Transactions on Emerging Topics in Computing (2018). doi:10.1109/TETC.2018.2860051.

[13] A. Azab, M. Alazab, M. Aiash, Machine learning based botnet identification traffic, in: 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, IEEE, 2016, pp. 1788–1794. doi:10.1109/TrustCom.2016.0275.

[14] M. M. Yamin, A. Shalaginov, B. Katt, Smart policing for a smart world opportunities, challenges and way forward, in: Future of Information and Communication Conference, Springer, Cham, 2020, pp. 532–549.

[15] F. Zhang, X. Hao, J. Chao, S. Yuan, Label propagation-based approach for detecting review spammer groups on e-commerce websites, Knowl. Based Syst. 193 (2020) 105520. doi:10.1016/j.knosys.2020.105520.

[16] L. You, Q. Peng, Z. Xiong, D. He, M. Qiu, X. Zhang, Integrating aspect analysis and local outlier factor for intelligent review spam detection, Future Gener. Comput. Syst. 102 (2020) 163–172. doi:10.1016/j.future.2019.07.044.

[17] M. Bao, J. Li, J. Zhang, H. Peng, X. Liu, Learning semantic coherence for machine generated spam text detection, in: International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019, IEEE, 2019, pp. 1–8. doi:10.1109/IJCNN.2019.8852340.

[18] Y. Liu, B. Pang, X. Wang, Opinion spam detection by incorporating multimodal embedded representation into a probabilistic review graph, Neurocomputing 366 (2019) 276–283. doi:10.1016/j.neucom.2019.08.013.

[19] A. C. Pandey, D. S. Rajpoot, Spam review detection using spiral cuckoo search clustering method, Evol. Intell. 12 (2) (2019) 147–164. doi:10.1007/s12065-019-00204-x.

[20] Z. Wang, S. Gu, X. Xu, GSLDA: lda-based group spamming detection in product reviews, Appl. Intell. 48 (9) (2018) 3094–3107. doi:10.1007/s10489-018-1142-1.

[21] C. Yuan, W. Zhou, Q. Ma, S. L., J. Han, S. Hu, Learning review representations from user and product level information for spam detection, in: 2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019, IEEE, 2019, pp. 1444–1449. doi:10.1109/ICDM.2019.00188.

[22] J. R. Méndez, T. R. Cotos-Yáñez, D. Ruano-Ordás, A new semantic-based feature selection method for spam filtering, Appl. Soft Comput. 76 (2019) 89–104. doi:10.1016/j.asoc.2018.12.008.

[23] J. Su, Q. Bai, S. Sindakis, X. Zhang, T. Yang, Vulnerability of multinational corporation knowledge network facing resource loss, Management Decision (2020). doi:10.1108/MD-02-2019-0227.

[24] N. Hussain, H. T. Mirza, I. Hussain, F. Iqbal, I. Memon, Spam review detection using the linguistic and spammer behavioral methods, IEEE Access 8 (2020) 53801–53816. doi:10.1109/ACCESS.2020.2979226.

[25] E. Elakkiya, S. Selvakumar, GAMEFEST: genetic algorithmic multi evaluation measure based feature selection technique for social network spam detection, Multim. Tools Appl. 79 (11-12) (2020) 7193–7225. doi:10.1007/s11042-019-08334-1.

[26] S. Kennedy, N. Walsh, K. Sloka, A. McCarren, J. Foster, Fact or factitious? contextualized opinion spam detection, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, Florence, Italy, Association for Computational Linguistics, 2019, pp. 344–350. doi:10.18653/v1/p19-2048.

[27] Z. Wu, J. Cao, Y. Wang, Y. Wang, L. Zhang, J. Wu, hpsd: A hybrid pu-learning-based spammer detection model for product reviews, IEEE Trans. Cybern. 50 (4) (2020) 1595–1606. doi:10.1109/TCYB.2018.2877161.

[28] M. Fazil, M. Abulaish, A hybrid approach for detecting automated spammers in twitter, IEEE Trans. Information Forensics and Security 13 (11) (2018) 2707–2719. doi:10.1109/TIFS.2018.2825958.

[29] S. Rathore, V. Loia, J. H. Park, Spamspotter: An efficient spammer detection framework based on intelligent decision support system on facebook, Appl. Soft Comput. 67 (2018) 920–932. doi:10.1016/j.asoc.2017.09.032.

[30] H. Chen, J. Liu, Y. Lv, M. H. Li, M. Liu, Q. Zheng, Semi-supervised clue fusion for spammer detection in *Sina Weibo*, Inf. Fusion 44 (2018) 22–32. doi:10.1016/j.inffus.2017.11.002.

[31] S. Shehnepoor, M. Salehi, R. Farahbakhsh, N. Crespi, Netspam: A network-based spam detection framework for reviews in online social media, IEEE Trans. Information Forensics and Security 12 (7) (2017) 1585–1595. doi:10.1109/TIFS.2017.2675361.

[32] A. Li, Z. Qin, R. Liu, Y. Yang, D. Li, Spam review detection with graph convolutional networks, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, ACM, 2019, pp. 2703–2711. doi:10.1145/3357384.3357820.

[33] Z. Wang, S. Gu, X. Zhao, X. Xu, Graph-based review spammer group detection, Knowl. Inf. Syst. 55 (3) (2018) 571–597. doi:10.1007/s10115-017-1068-7.

[34] Y. Liu, B. Pang, A unified framework for detecting author spamicity by modeling review deviation, Expert Syst. Appl. 112 (2018) 148–155. doi:10.1016/j.eswa.2018.06.028.

[35] H. Fu, X. Xie, Y. Rui, N. Z. Gong, G. Sun, E. Chen, Robust spammer detection in microblogs: Leveraging user carefulness, ACM Trans. Intell. Syst. Technol. 8 (6) (2017) 83:1–83:31. doi:10.1145/3086637.

[36] D. Yu, N. Chen, F. Jiang, B. Fu, A. Qin, Constrained nmf-based semi-supervised learning for social media spammer detection, Knowl. Based Syst. 125 (2017) 64–73. doi:10.1016/j.knosys.2017.03.025.

[37] C. Li, S. Wang, L. He, P. S. Yu, Y. Liang, Z. Li, SSDMV: semi-supervised deep social spammer detection by multi-view data fusion, in: Proceedings of the 2018 IEEE International Conference on Data Mining, Singapore, IEEE Computer Society, 2018, pp. 247–256. doi:10.1109/ICDM.2018.00040.

[38] F. Wu, C. Wu, J. Liu, Semi-supervised collaborative learning for social spammer and spam message detection in microblogging, in: Proceedings of the 27th ACM International Conference on Information and Knowledge

Management, CIKM 2018, Torino, Italy, ACM, 2018, pp. 1791–1794. doi:10.1145/3269206.3269324.

[39] V. Ranjbar, M. Salehi, P. Jandaghi, M. Jalili, Qanet: Tensor decomposition approach for query-based anomaly detection in heterogeneous information networks, IEEE Trans. Knowl. Data Eng. 31 (11) (2019) 2178–2189. doi:10.1109/TKDE.2018.2873391.

[40] F. Xia, J. Liu, H. Nie, Y. Fu, L. Wan, X. Kong, Random walks: A review of algorithms and applications, IEEE Trans. Emerging Topics in Comput. Intellig. 4 (2) (2020) 95–107. doi:10.1109/TETCI.2019.2952908.

[41] S. Bock, M. Weiß, Non-convergence and limit cycles in the adam optimizer, in: Artificial Neural Networks and Machine Learning - ICANN 2019: Deep Learning - 28th International Conference on Artificial Neural Networks, Munich, Germany, Vol. 11728 of Lecture Notes in Computer Science, Springer, 2019, pp. 232–243. doi:10.1007/978-3-030-30484-3_20.

[42] C. Yang, R. Harkreader, G. Gu, Empirical evaluation and new design for fighting evolving twitter spammers, IEEE Transactions on Information Forensics and Security 8 (8) (2013) 1280–1293. doi:10.1109/TIFS.2013.2267732.

[43] M. C. Yang, H. C. Rim, Identifying interesting twitter contents using topical analysis, Expert Syst. Appl. 41 (9) (2014) 4330–4336. doi:10.1016/j.eswa.2013.12.051.

[44] H. Park, T. Park, Y. S. Lee, Partially collapsed gibbs sampling for latent dirichlet allocation, Expert Syst. Appl. 131 (2019) 208–218. doi:10.1016/j.eswa.2019.04.028.

[45] C. Y. Lin, A reversible privacy-preserving clustering technique based on k-means algorithm, Appl. Soft Comput. 87 (2020) 105995. doi:10.1016/j.asoc.2019.105995.

[46] W. J. Baddar, Y. M. Ro, Encoding features robust to unseen modes of variation with attentive long short-term memory, Pattern Recognit. 100 (2020) 107159. doi:10.1016/j.patcog.2019.107159.

[47] R. Wang, N. Xiu, C. Zhang, Greedy projected gradient-newton method for sparse logistic regression, IEEE Trans. Neural Networks Learn. Syst. 31 (2) (2020) 527–538. doi:10.1109/TNNLS.2019.2905261.

[48] X. Tao, Q. Li, C. Ren, W. Guo, Q. He, R. Liu, J. Zou, Affinity and class probability-based fuzzy support vector machine for imbalanced data sets, Neural Networks 122 (2020) 289–307. doi:10.1016/j.neunet.2019.10.016.

[49] A. Kuri-Morales, Closed determination of the number of neurons in the hidden layer of a multi-layered perceptron network, Soft Comput. 21 (3) (2017) 597–609. doi:10.1007/s00500-016-2416-3.

[50] B. Guan, G. Zhang, J. Yao, X. Wang, M. Wang, Arm fracture detection in x-rays based on improved deep convolutional neural network, Comput. Electr. Eng. 81 (2020) 106530. doi:10.1016/j.compeleceng.2019.106530.