Collective Spammer Detection in Evolving Multi-Relational Social Networks

Shobeir Fakhraei*
University of Maryland
College Park, MD, USA
shobeir@cs.umd.edu

James Foulds
University of California
Santa Cruz, CA, USA
jfoulds@ucsc.edu

Madhusudana Shashanka†
if(we) Inc.
San Francisco, CA, USA
shashanka@alum.bu.edu

Lise Getoor
University of California
Santa Cruz, CA, USA
getoor@soe.ucsc.edu

ABSTRACT

Detecting unsolicited content and the spammers who create it is a long-standing challenge that affects all of us on a daily basis. The recent growth of richly-structured social networks has provided new challenges and opportunities in the spam detection landscape. Motivated by the *Tagged.com*[1] social network, we develop methods to identify spammers in evolving multi-relational social networks. We model a social network as a time-stamped multi-relational graph where vertices represent users, and edges represent different activities between them. To identify spammer accounts, our approach makes use of structural features, sequence modelling, and collective reasoning. We leverage relational sequence information using k-gram features and probabilistic modelling with a mixture of Markov models. Furthermore, in order to perform collective reasoning and improve the predictive power of a noisy abuse reporting system, we develop a statistical relational model using hinge-loss Markov random fields (HL-MRFs), a class of probabilistic graphical models which are highly scalable. We use *Graphlab Create*™ and *Probabilistic Soft Logic (PSL)*[2] to prototype and experimentally evaluate our solutions on internet-scale data from *Tagged.com*. Our experiments demonstrate the effectiveness of our approach, and show that models which incorporate the multi-relational nature of the social network significantly gain predictive performance over those that do not.

*Contribution partly performed while under internship at if(we) Inc., formerly Tagged Inc.

†Currently with Niara, Inc., Sunnyvale, CA.

[1] *Tagged.com* was founded in 2004, has over 300 million registered members, and is aimed towards fostering new connections between people.

[2] http://psl.umiacs.umd.edu

KDD'15 August 11-14, 2015, Sydney, NSW, Australia

ACM 978-1-4503-3664-2/15/08 ...$15.00
DOI: http://dx.doi.org/10.1145/2783258.2788606.

Keywords

Social Networks, Spam, Social Spam, Collective Classification, Graph Mining, Multi-relational Networks, Heterogeneous Networks, Sequence Mining, Tree-Augmented Naive Bayes, k-grams, Hinge-loss Markov Random Fields (HL-MRFs), Probabilistic Soft Logic (PSL), Graphlab.

1. INTRODUCTION

Unsolicited or inappropriate messages sent to a large number of recipients, known as "spam", can be used for various malicious purposes, including phishing and virus attacks, marketing of objectionable materials and services, and compromising the reputation of a system. From printed advertisements to unsolicited phone calls, spam has been a perennial problem in modern human communication. With the emergence of the Internet, spammers have found a cost-effective medium to reach a broader audience than was previously possible. Email spam is almost as old as the Internet itself. The first email spam was sent in 1978 to all several hundred users of ARPANET [1].

More recently, social media has given spammers a new and effective medium to spread their content. Using social media platforms, spammers can disguise themselves as legitimate users and engage in realistic looking interactions. They can use these platforms to send messages to users, leave spam comments on popular pages, and reply to legitimate comments using spam content. Such diversity of choice has often increased spammers' ability to conceal their intentions from traditional spam filters. According to a study by Nexgate [2], social spam grew by more than 355% between January to July of 2013, one in 200 social messages contain spam, and 5% of all social apps are spammy.

While content-based approaches have been shown to be effective in stopping spam in email and the web, they can be manipulated by sophisticated spammers via incorporating content randomness. Unlike in email and the web, social media enables spammers to split their content across multiple messages in order to bypass spam filters. Link-based approaches that leverage the connectivity of the entities, have been combined with content-based methods to build more effective methods. While it is easier to pass traditional content-based filters, behavioral patterns and graph properties of the users' interactions are harder to manipu-

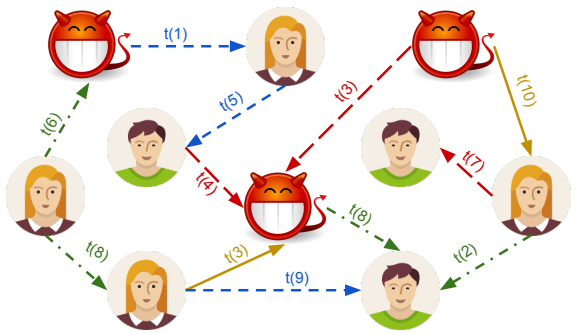


Figure 1: A time-stamped multi-relation social network with legitimate users and spammers. Each link $\langle v_1, v_2 \rangle$ in the network represents an action (e.g. profile view, message, or poke) performed by v_1 towards v_2 at specific time t.

late. Furthermore, many social networks can not monitor all the generated contents due to privacy and resources concerns. Content-independent frameworks, such as the one proposed in this paper, can be applied to systems that provide maximum user privacy with end-to-end encryption.

Perhaps the most important difference between social networks and email or web graphs is that social networks have a multi-relational nature, where users have relationships of different types with other users and entities in the networks. For example, they can send messages to each other, add each other as friends, "like" each other's posts, and send non-verbal signals such as "winks" or "pokes." Figure 1 shows a representation of a social network as a time-stamped multi-relation graph. The multi-relational nature provides more choices for spammers, but it also empowers detection systems to monitor patterns across activity types, and time. In this paper, we propose a content-independent framework which is based on the multi-relational graph structure of different activities between users, and their sequences.

Our proposed framework is motivated by *Tagged.com*, a social network for meeting new people which was founded in 2004 and has over 300 million registered members. More generally, the framework is applicable to any multi-relational social network. Our goal is to identify sophisticated spammers that require manual or semi-automated intervention by the administrative security team. These spammers have already passed initial classifiers and know how to manipulate their accounts and contents to avoid being caught by automatic filters. We show that our framework significantly reduces the need for manual administration to control spam.

Our framework consists of three components. First, we extract graph structure features for each of the relations and show that considering the multi-relational nature of the graphs improves the performance. Second, we consider the activity sequence of each user across these relations and extract k-gram features and employ mixtures of Markov models to label spammers. Third, we propose a statistical relational model based on hinge-loss Markov random fields to perform collective reasoning using signals from an abuse reporting system in the social network.

The following sections formally define the problem and our solution framework along with an experimental validation of our approach on internet-scale data from *Tagged.com*.

2. PROBLEM STATEMENT

We represent a social network as a directed time-stamped dynamic multi-relational graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, where $\mathcal{V}$ is the set of vertices of the form $v = \langle f_1, \ldots, f_n \rangle$ representing users and their demographic features f_i, and $\mathcal{E}$ is the set of directed edges of the form $e = \langle v_{src}, v_{dst}, r_i, t_i \rangle$ representing their interactions, relation type r_i, and a discrete time-stamp t_i. The social spam detection problem is to predict whether v_i with an unobserved label is a spammer or not, based on the given network $\mathcal{G}$ and a set of observed labels for already identified spammers. Since the deployed security system could employ different measures based on the classification confidence, we are interested in (un-normalized) probabilities or ranking scores of the likelihood that each user is a spammer. In other words, the problem is assignment of a score (e.g., a probability) to user accounts to rank them from the most to the least probable spammer in the system: $c : v_i \rightarrow [0, 1]$.

3. OUR METHOD

In our framework, we focus on three different mechanisms to identify spammers and malicious activities. We first create networks from the user interactions and compute network structure features from them. As these are evolving networks, each user generates a sequence of actions with the passage of time. Mining these sequences can provide valuable insights into the intentions of the user. We use two methods to study these sequences and extract features from them. We use the output of these methods as features to classify spammers. We then employ a collective model to identify spammer accounts only based on the signals from the abuse reporting system ($\mathcal{G}_{\text{report}}$) as a secondary source to reassure predictions. The following sections discuss our framework and extracted features in more details.

3.1 Graph Structure Features ($\mathbf{X}_{\mathcal{G}}$)

We create a directed graph $\mathcal{G}_r = \langle \mathcal{V}, \mathcal{E}_r \rangle$ for each relation r in the social network, where vertices $\mathcal{V}$ consist of users, and edges $\mathcal{E}_r$ represent interactions of type r between users, e.g. if user_1 sends a message to user_2 then $\mathcal{G}_{\text{message}}$ will contain v_1 and v_2 representing the two users, and $e_{1,2}$ representing the relation between them. We have ten different graphs each containing the same users as vertices but different actions as edges.

We use *Graphlab Create*™[3] to generate features based on each of these graphs for each user. We use six graph analytics methods m_i to compute the features. Using each m_i we create a set of features for each relation graph $\mathcal{G}_r$ as following:

$$\mathbf{X}^{m_i}_{\mathcal{G}_\mathbf{r}} = \left[\begin{array}{ccc} \mathbf{X}^{m_i}_{\mathcal{G}_{r_1}} & \ldots & \mathbf{X}^{m_i}_{\mathcal{G}_{r_n}} \end{array} \right]$$

where m_i is one of the graph analytics methods described below, and r_i is one of the relationships considered in the study.

We then use these features together to get a complete multi-relational graph feature-set, as the following:

$$\mathbf{X}^{\mathbf{m}}_{\mathcal{G}_\mathbf{r}} = \left[\begin{array}{ccc} \mathbf{X}^{m_1}_{\mathcal{G}_\mathbf{r}} & \ldots & \mathbf{X}^{m_k}_{\mathcal{G}_\mathbf{r}} \end{array} \right]$$

The graph analytics methods m_i we use to extract the features from each relation network are described in the fol-

[3] http://dato.com/products/create

lowing section. Each of these algorithms provieds different perspectives on the local connectivity of the graph and neighborhood characteristics of each user. Our goal is to capture the structural differences between spammers' and legitimate users' multi-relational neighborhood graph.

PageRank:

PageRank [3], is a well known ranking algorithm proposed for ranking websites, and computes a score for each node by considering the number and quality of links to a node. The algorithm is based on the underlying assumption that important nodes receive more links from other nodes.

Degree:

We compute the total degree, in-degree, and out-degree of each node for each relation, which correspond to the total number of activities a user has been involved in, the number of communications (or actions) a user received, and the number of actions the user performed.

k-core:

k-core [4] is a centrality measure that is based on the graph decomposition via a recursive pruning of the least connected vertices. The value each vertex receives depends on the step in which the vertex is eliminated from the graph. e.g, vertices removed on the third iteration receive the value three.

Graph Coloring:

Graph coloring [5] is an assignment of colors to elements (here vertices) of a graph, such that no two adjacent vertices share the same color. Using a greedy implementation, we obtain the color identifier of each vertex as a feature.

Connected Components:

A connected component [6] is a group of vertices with a path between each vertex and all other vertices in the component. A weakly connected component is a maximal set of vertices such that there is an undirected path between any two vertices in the set. We compute the weakly connected component on each graph and extract the component identifier and size of the component that the vertex participates in as features.

Triangle Count:

The triangle count [7] of a vertex is the number of triangles (a complete subgraph of three vertices) in the graph the vertex participates in. Such number is an indication of the connectivity of the graph around that vertex.

3.2 Sequence-Based Features ($\mathbf{X}_{\mathcal{S}}$)

Sequence classification is used in many domains, including biology and health-informatics, anomaly detection, and information retrieval [8]. In dynamically evolving multi-relational social networks, each user v_i generates a sequence of edges via their actions as the following:

$$\mathcal{S}^{v_i} = \langle r_p, \ldots, r_q \rangle$$

Spammers typically pursue specific purposes in the network and it is likely that their sequence of actions diverge from the norm. In this section we study these sequences and provide two different solutions for classifying users based on their activity sequences. It is important to note that such an approach would not be possible if the network were not multi-relational.

Sequential k-gram Features

The simplest way to represent a sequence with features is to treat each element in the sequence as a feature independently. However, the order of the sequence cannot be captured with this approach. Furthermore, in our scenario the values of these features will be the same as the out-degree for each vertex, which we previously computed in the graph-based features. To address this, a short sequence segment of k consecutive actions, called a k-gram, can be used to capture the order of events [8]. The sequence can be represented as a vector of the frequencies of the k-grams. To keep the feature space computationally manageable we chose bigram sequence features where $k = 2$. For example, the number of times a user v_i sent a *message* after performing a *profile view*, would be the value for the feature $x^{v_i}_{\text{profileview-message}}$. The bigram feature set for the sequence $\mathcal{S}$ will be the following:

$$\mathbf{X}_{\mathcal{S}_B} = \left[\mathbf{X}_{r_1 r_1} \ \ldots \ \mathbf{X}_{r_p r_q} \ \ldots \ \mathbf{X}_{r_n r_n} \right]$$

where r_i is one of the relationships considered in the study, $\mathbf{X}_{r_p r_q} = \left[x^{v_1}_{r_p r_q} \ldots x^{v_n}_{r_p r_q}\right]^{\intercal}$, and $x^{v_i}_{r_p r_q}$ is the total number of times user v_i performed an action of type r_q consecutively after performing r_p.

Mixture of Markov Models

While k-gram features capture some aspects of the order of elements in the sequence, they may miss patterns in longer sequences. Increasing k will rapidly increase the feature space, introducing computational barriers and estimation challenges due to feature sparsity. Instead, to capture the salient information from longer sequence chains, and to study the predictive power of this information, we construct a simple generative model for sequence data. The model is equivalent to the chain-augmented naive Bayes model of [9], a special case of the tree-augmented naive Bayes model [10] which has been shown to be effective in language modelling. The model posits that each user's actions are generated via a mixture of Markov models. In more detail, each class (spammer or not spammer) is associated with a mixture component y. Conditional on the class (mixture component) y for a user, that user's sequence of actions are assumed to be generated from a Markov chain specific to that class. The joint probability for a user's class y and action sequence $x_i, \ldots, x_n$ is given by

$$P(y, x) = P(y)P(x_1|y)\prod_{i=2}^{n} P(x_i|x_{i-1}, y) \ ,$$

which we summarize with a directed graphical model diagram in Figure 2. We place symmetric Dirichlet priors on the parameters of the discrete distributions $P(y)$, $P(x_1|y)$, and $P(x_i|x_{i-1}, y)$, and compute maximum a posteriori (MAP) estimates of them, which are readily obtained as the proportion of each outcome in the training data, with the counts first adjusted by adding the Dirichlet smoothing parameter $\alpha = 1$. Finally, at test time we compute the posterior probability of the user's class label given the observed action sequence x via Bayes rule, $P(y|x) \propto P(x|y)P(y) = P(y, x)$.

There are multiple methods to incorporate the predictions from this model into our framework. We simply use the

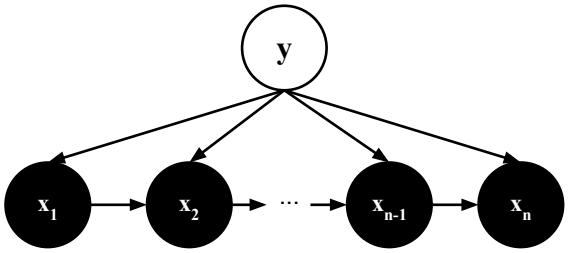


Figure 2: The directed graphical model for the mixture of Markov models / chain-augmented naive Bayes model, for one user. In the diagram, y indicates the label (spammer or not) and x_i represents the ith action performed by the user.

ratio of posterior probabilities and their logarithmic forms as a small feature-set ($\mathbf{X}_{\mathcal{S}_M}$) for our classifier.

3.3 Collective Classification with Reports

Most websites that enable users to publish content also provide an *abuse reporting* mechanism for other users to bring malicious behavior to the system's attention. However, these systems do not necessary offer clean signals. Spammers themselves often randomly report other users (spammers and legitimate users) to increase the noise, legitimate users often have different standards for malicious behaviors, and users may report others for personal gains such as censorship or blocking an opponent in a (social) game from accessing the system. A model that can extract sufficient information from the relational *report* feature, can enhance the administrative team's performance by focusing their attention, and can also provide an additional feature or parallel mechanism for spam classification.

We propose a model based on hinge-loss Markov random fields [11] to collectively classify spammers within the reported users, and assign credibility scores to the users offering feedback via the reporting system. Using this model a better ranking of the reported users based on their probability of being spammers can be provided to the security administration team. The hinge-loss formulation has the advantage of admitting highly scalable inference, regardless of the structure of the network.

Hinge-loss Markov Random Fields

Hinge-loss Markov random fields (HL-MRFs) are a general class of conditional, continuous probabilistic models [11, 12]. HL-MRFs are log-linear models whose features are hinge-loss functions of the variable states. Through constructions based on *soft logic*, hinge-loss potentials can be used to model generalizations of logical conjunction and implication. A hinge-loss Markov random field P over random variables $\mathbf{Y}$ and conditioned on random variables $\mathbf{X}$ defines a conditional probability density function as the following:

$$P\left(\mathbf{Y}|\mathbf{X}\right) = \frac{1}{Z(\lambda)} \exp\left[-\sum_{j=1}^{m} \lambda_j \phi_j(\mathbf{Y},\mathbf{X})\right] ,$$

where Z is the normalization constant of the form

$$Z = \int_{\mathbf{Y}} \exp\left[-\sum_{j=1}^{m} \lambda_j \phi_j(\mathbf{Y},\mathbf{X})\right] .$$

In the above, ϕ is a set of m continuous potential of the form

$$\phi_j(\mathbf{Y},\mathbf{X}) = \left[\max\left\{\ell_j(\mathbf{Y},\mathbf{X}), 0\right\}\right]^{p_j} ,$$

where ℓ is a linear function of $\mathbf{Y}$, and $\mathbf{X}$ and $p_j \in \{1,2\}$.

Probabilistic Soft Logic (PSL) [12] uses a first-order logical syntax as a templating language for HL-MRFs. HL-MRFs have achieved state-of-the-art performance in many domains including knowledge graph identification [13], understanding engagements in MOOCs [14], biomedicine and multi-relational link prediction [15, 16], and modelling social trust [17]. A typical example of a PSL rule is

$$\lambda : P(a,b) \wedge Q(a) \rightarrow R(b),$$

where P, Q, and R are *predicates*, a and b are *variables*, and λ is the weight associated with the rule, indicating its importance. For instance, $P(a,b)$ can represent a relational edge in the graph such as $\textsc{Reported}(a,b)$, and $Q(a)$ could represent a value for a vertex such as $\textsc{Credible}(b)$. Each grounding forms a ground atom, or logical fact, that has a soft-truth value in the range $[0,1]$. The rules can encode domain knowledge about dependencies between these predicates. PSL uses the *Lukasiewicz* norms to provide relaxations of the binary connectives to soft-truth values. A ground instance of a rule r ($r_{\text{body}} \longrightarrow r_{\text{head}}$) is satisfied when the value of r_{body} is not greater than the value of r_{head}. ℓ is defined to capture the distance to satisfaction for rules:

$$\ell = val(r_{\text{body}}) - val(r_{\text{head}}) .$$

HL-MRFs Collective Model for Reports

The goal of this model is to use reports to predict spammers. We study three HL-MRFs models to incorporate the reporting users' credibility into the reporting system and improve the predictability of the reports. We show that collective reasoning over credibility of the reporting user and the probability of the reported user being an spammer, increases the classification performance of the system.

Our collective HL-MRFs model uses the *report* relation graph ($\mathcal{G}_{\text{report}}$), and is based on the intuition that the credibility of a user's abuse reporting should increase when they report users that are more likely to be spammers. Hence, if a user reports other users whom there are other evidence supporting them being spammers, the credibility of that person should increase. On the other hand, if the user reports another user that is unlikely to be a spammer, the credibility of the reporting user should decrease.

$$\begin{aligned}
\textsc{Credible}(v_1) \wedge \textsc{Reported}(v_1,v_2) &\rightarrow \textsc{Spammer}(v_2)\\
\textsc{Spammer}(v_2) \wedge \textsc{Reported}(v_1,v_2) &\rightarrow \textsc{Credible}(v_1)\\
\neg\textsc{Spammer}(v_2) \wedge \textsc{Reported}(v_1,v_2) &\rightarrow \neg\textsc{Credible}(v_1)\\
\textsc{Prior-Credible}(v) &\rightarrow \textsc{Credible}(v)\\
\neg\textsc{Prior-Credible}(v) &\rightarrow \neg\textsc{Credible}(v)\\
&\neg\textsc{Spammer}(v)
\end{aligned}$$

Figure 3: Collective HL-MRFs model to predict spammers based on the reports from other users.

We propose the model shown in Figure 3 to capture the collective intuition. We incorporate prior credibility of the reporting users based on the past reporting behavior into the model. The negative prior on *SPAMMER* is included

in the model to complement the first rule that increases the score of users being spammers. To study the effect of each part of the model, we experimentally compare the proposed collective model with two simpler HL-MRFs models that do not contain the collective reasoning and credibility priors in section 5.4.

4. DATA

The dataset[4] was collected from the *Tagged.com* social network website, which is a network for meeting new people, and has multiple methods for users to make new connections. *Tagged* has various methods to deal with spam. It uses several registration and activation filters to identify and block spam accounts based on traditional methods such as content and registration information and patterns. *Tagged* also employs a reporting mechanism that users can report spammers to the system. An administrative security team also monitors the network for malicious behaviors and manually blocks spammers. Our goal in this study is to identify sophisticated spammers that require manual intervention by the security team. These spammers have already passed initial classifiers and know how to manipulate their content to avoid being caught by automatic filters.

The purpose of the social network affects its susceptibility to spam. A social network which is designed for connecting the users who already know each other, can control spam by limiting the communications between users who are not already connected in the network. However, a social network that promotes finding new connections may like to impose minimum limitations on how users interact. *Tagged*, which is a social network for meeting new people, has multiple venues for users to communicate without much restriction.

Another challenge with identifying spammers in multi-purpose social networks such as *Tagged* is that users join the network for different reasons. For example, users may come to *Tagged* to play social games such as *Pets* and *MeetMe*, to find romantic relationships, or simply to spend time with virtual connections. Not only they will generate different behavioral patterns, they will use security measures such as *abuse reporting* mechanism differently and introduce noise to it.

In our experiments, all the users who had at least one activity in the sampling time frame were included in the sample dataset. More formally our initial sample included the following elements:

$$\begin{aligned} \mathcal{V} &= \{\, v \mid \exists\, e = \langle v, v_*, r_*, t_k\rangle \in \mathcal{E}_{\text{all}} \ \wedge\ t_{\text{b}} \le t_k \le t_{\text{e}}\} \\ \mathcal{E} &= \{\, e = \langle v_i, v_j, r_*, t_k\rangle \mid \exists\, v_i, v_j \in \mathcal{V} \ \wedge\ t_{\text{s}} \le t_k \le t_{\text{e}}\} \end{aligned}$$

where v_* indicates any user in the network, r_* indicates any type of action in the study, $\mathcal{E}_{\text{all}}$ indicates all the edges in the *Tagged* network, and t_{b} and t_{e} indicate the time of the beginning and the end of the sampling period.

To perform a retrospective study, we chose t_{b} and t_{e} such that enough time had passed since the sampling period by the time we accessed the data (t_{access}), so that most of the spam accounts were identified and labeled. We then removed the users who had deactivated their accounts themselves by t_{access}, because we could not determine their labels. The remaining users were labeled as spam if their accounts has been manually canceled by a security team by t_{access}. Although the security team cancels accounts for multiple reasons, not just spam, most of the canceled accounts are due to malicious activities. For simplicity, we labeled all the canceled accounts as spammers. Ten different activities on the website were selected during the sampling time frame. The activities included in the study are: viewing another user's profile, sending friend requests, sending messages, sending *luv*, sending *winks*, buying or wishing others in the *Pets* game, clicking yes or no in the *MeetMe* game, and reporting other users for abuse.

There are more effective ways to sample the network in order to conserve its characteristics [18, 19, 20]. However, for practical reasons and ease of deployment, we have chosen the simple time-based sampling method. Further performance improvements may be achieved via better sampling employments. The spammer accounts that were selected for this study could initially bypass *Tagged* deployed preventative measures and successfully perform at least one action in the network. Although they could be identified within a short period of time after their activity, their identification required a manual or semi-automated procedure by the members of the security team. Not only are these spammers harder to identify, they are also very rare in the dataset, causing a huge class imbalance.

Table 1 shows some statistics from the sample we used. These numbers do not represent the statistics of the *Tagged* social network, as they have been altered by limiting the number of action types in the study as well as eliminating users with deactivated accounts at t_{access} (which is later than the sample period). Furthermore, only the users who performed an action in the sampling period were included in the dataset.

Table 1: Data Sample Statistics.

Entity	Count
$\lvert\mathcal{V}\rvert$ (total users)	5,607,454
$\lvert\mathcal{E}\rvert$ (total actions)	912,280,409
$\max(\lvert\mathcal{E}_r\rvert)$ (number of actions that are most frequent action type)	350,724,903
$\min(\lvert\mathcal{E}_r\rvert)$ (number of actions that are least frequent action type)	137,550
total users labeled as spammers	(%3.9) 221,305

All of our experiments are based on the relational data in the following form:

$$\langle t_i, v_{\text{src}}, v_{\text{dst}}, r_j\rangle$$

where t_i is the time stamp, v_{src} is the user who initiated the action, v_{dst} is the user the action was towards, and r_i categorizes the type of action.

5. EXPERIMENTAL VALIDATION

We performed four sets of experiments to evaluate the proposed methods. First we study the graph structure properties and compare the multi-relational approach with only considering a single relation. We also study using one graph analytics algorithm as a feature, comparing to having features from multiple methods. We then study the effectiveness of sequence mining features and combine them with graph-based methods to measure the overall performance

[4]An anonymized sample of the multi-relational part of the dataset along with our code for the experiments can be found here: http://github.com/shobeir/fakhraei_kdd2015.

enhancements. We then include only three demographics features for each user to measure their influence on the performance. Finally we perform collective reasoning over *abuse reports* and measure the improvement of the predictions with this method.

For our experiments we used *Graphlab Create*^{TM} and the Java-based open-source *Probabilistic Soft Logic (PSL)*,[5] on a single Ubuntu machine with 32GB RAM and 3.2GHz CPU (4 cores). For classification, we used *Gradient-Boosted Decision Trees* which is a collection of decision trees combined through a technique called gradient boosting [21].

The deployment options of the framework and what actions are planned to be taken on the identified spammer accounts determine which performance metrics are more appropriate for this task. High precision lets the spam accounts be blocked without manual intervention, and without concerns of the system harming legitimate users. High recall allows the system to identify the legitimate users with more confidence and clear the environment via deploying measures such as CAPTCHA and additional account verifications for the users with suspicious status. Hence, the appropriate metric to measure the performance of this system is the *Precision-Recall* curve. The *ROC* curve could also be useful, however, due to the high class-imbalance, it would not provide much insight, and unless properly adjusted, it would result in over-optimistic estimates. We report the area under *Precision-Recall* curve (AUPR) and the area under the *ROC* curve (AUROC) for the experiments. We used 10-fold cross-validation to estimate the performance of each method and feature-set. Unless stated otherwise, the reported numbers represent *mean* and *standard deviation* over 10-fold cross-validation.

5.1 Graph Structure Features

Table 2 shows the average results of classification via graph-based features. The first row indicated the best results from using a single relation with features from all the graph-based algorithms. The second row shows the best graph-based feature with all the relations. Comparing the results from the two rows suggests that combining different relations is more effective than combining features from different algorithms on a single relation. Using all algorithms to compute features on all relation graphs results in the best performance for graph-based methods.

Table 2: Classification with graph-based features.

Experiment		AUPR	AUROC
$\left[\mathbf{X}^{\mathbf{m}}_{\mathcal{G}_{r_i}}\right]$	1 Relation, k Methods	0.187±0.004	0.803±0.001
$\left[\mathbf{X}^{m_i}_{\mathcal{G}_{\mathbf{r}}}\right]$	n Relations, 1 Method	0.285±0.002	0.809±0.001
$\left[\mathbf{X}^{\mathbf{m}}_{\mathcal{G}_{\mathbf{r}}}\right]$	n Relations, k Methods	0.328±0.003	0.817±0.001

5.2 Sequence-based Features

Next, we experimentally evaluated the sequence-based features. First, we study their effectiveness independently, and then we measure their performance in combination with the graph-based features. To compute the bigram features, we first sorted all of the activities in our dataset based on user IDs and timestamps via the standard external sort function in Linux. We did a single pass on the sorted file to compute the bigram features.

[5] http://psl.umiacs.umd.edu

Table 3: Classification with k-gram features.

Experiment	AUPR	AUROC
$\left[\mathbf{X}_{\mathcal{S}_B}\right]$ k-gram features	0.471±0.004	0.859±0.001
$\left[\mathbf{X}_{\mathcal{S}_B}\ \mathbf{X}^{\mathbf{m}}_{\mathcal{G}_{\mathbf{r}}}\right]$ k-gram & graph features	0.543±0.005	0.914±0.001

Table 3 shows the results of classification using the bigram features. The second row suggests that a model that uses both graph-based and k-gram features outperforms the ones that use them independently. *Precision-Recall* and *ROC* curves from graph-based and k-gram features are shown in Figure 4.

We further study the sequence-based classification with the Mixture of Markov Models (MMM) approach. We did a single pass on the sorted file we already generated for the bigram features to compute the probabilities for this model. We then used the probabilities generated from this model in logarithmic and ratio forms as features for classification.

Table 4: Classification with mixture of Markov models.

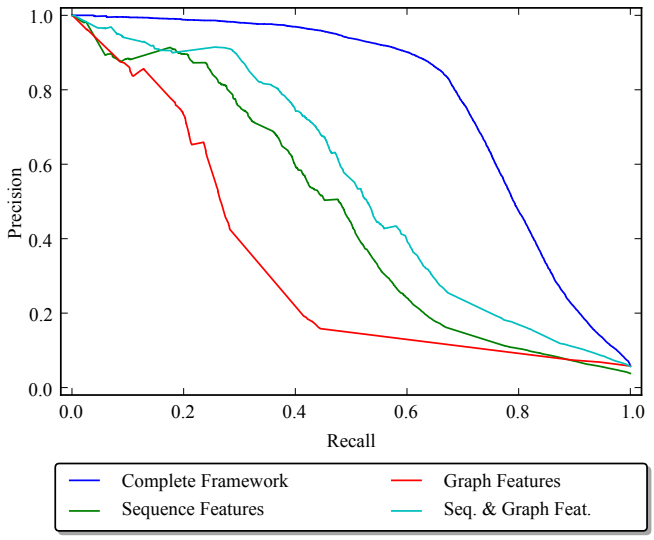
Experiment	AUPR	AUROC
$\left[\mathbf{X}_{\mathcal{S}_M}\right]$ MMM	0.246±0.009	0.821±0.003
$\left[\mathbf{X}_{\mathcal{S}_M}\ \mathbf{X}_{\mathcal{S}_B}\right]$ MMM & k-gram	0.468±0.012	0.860±0.002
$\left[\mathbf{X}_{\mathcal{S}_M}\ \mathbf{X}_{\mathcal{S}_B}\ \mathbf{X}^{\mathbf{m}}_{\mathcal{G}_{\mathbf{r}}}\right]$ MMM & k-gram & graph	0.550±0.005	0.914±0.002

The results from Table 4 shows the classification performance with these features, which suggests minimal improvement employing longer sequence models. This may suggest that the bigram features can incorporate enough signal to capture spam activity in a multi-relational network. However, computing the Mixture of Markov Models does not impose much overhead when extracting bigram features, and can be done within the same process.

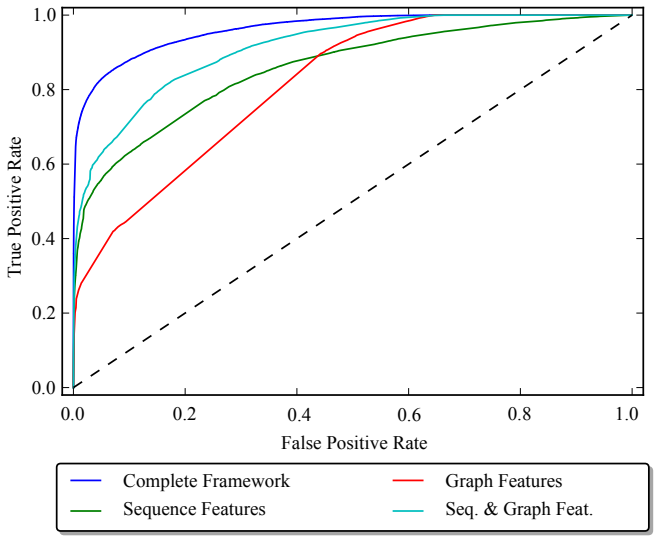
5.3 Demographic Information

Many people use *Tagged* to find new relationships. We anticipate that in such environment users behave differently based on their demographics. To capture this point, we added three features ($\mathbf{X}_{\mathcal{D}}$) to our model: age, gender, and time since registration. Age and gender highly improved the classification results as they tend to be most discriminative of behavioral patterns. Another feature that we included in our model is the time past since registration. As mentioned earlier we labeled all the cancelled accounts for malicious activities as spammers. However, these users have different behavioral patterns, where spammers who mainly mass advertise, may use much newer accounts, in contrast to users who have been blocked due to misbehaviors, and have been active in the system much longer.

Table 5 shows the significant improvements of the results when including these features in different models. Figure 4



(a) Precision-Recall Curve



(b) ROC Curve

Figure 4: *Precision-Recall* and *ROC* curves for models with k-gram and graph-based features. Using both features with user demographics significantly improve the results.

shows the *Precision-Recall* and *ROC* curves of the complete framework.

Table 5: Classification when including user demographics information.

Experiment	AUPR	AUROC
$\left[\mathbf{X}_{\mathcal{D}}\ \mathbf{X}_{\mathcal{S}_B}\right]$ Demo. & k-gram	0.689±0.006	0.935±0.001
$\left[\mathbf{X}_{\mathcal{D}}\ \mathbf{X}^{\mathbf{m}}_{\mathcal{G}_\mathbf{r}}\right]$ Demo. & graph	0.701±0.002	0.950±0.001
$\left[\mathbf{X}_{\mathcal{D}}\ \mathbf{X}_{\mathcal{S}_B}\ \mathbf{X}^{\mathbf{m}}_{\mathcal{G}_\mathbf{r}}\right]$ Demo. & k-gram & graph	0.778±0.001	0.963±0.001
$\left[\mathbf{X}_{\mathcal{D}}\ \mathbf{X}_{\mathcal{S}_M}\ \mathbf{X}_{\mathcal{S}_B}\ \mathbf{X}^{\mathbf{m}}_{\mathcal{G}_\mathbf{r}}\right]$ Demo. & MMM & k-gram & graph	0.779±0.002	0.963±0.001

5.4 Collective Classification with Reports

The reporting system can have useful information to detect spammers. We studied the effectiveness of our proposed collective model (in Figure 3) to extract useful signals from this relation. We first designed a baseline model shown in Figure 5a to only use the reports to detect spammers. This model gives similar results to assigning total count of the reports for each user as their score of being a spammer. We then designed the model shown in Figure 5b to use the reports and prior credibility of the reporting user to detect spammers. This model gives similar results to assigning total *weighted* count of the reports for each user as their score of being a spammer, where reports are weighted by the credibility of the reporting users.

To perform the experiments we have only used $\mathcal{G}_{report}$, which is a sparse graph. Our collective model is aimed to propagate information between the reported users' likelihood of being spammer, through the credibility of the reporting users. In order for information to propagate in the model, each reporting user should at least have reported two other users. Hence, we removed the vertices with out-degree less than two. We then performed 10-fold cross validation to compare the three models and study the effectiveness of the collective model. We used the ratio of the correctly reported spammers from the training data as a simple prior on credibility for each user. Potentially more effective priors could incorporate the the count and the frequency of the reports as well.

$$\textsc{Reported}(v_1, v_2) \rightarrow \textsc{Spammer}(v_2)$$
$$\neg\textsc{Spammer}(v)$$

(a) HL-MRFs model that only uses the reports to detect spammers. This model would give similar results to assigning total count of the reports for each user as their score of being a spammer.

$$\textsc{Credible}(v_1) \wedge \textsc{Reported}(v_1, v_2) \rightarrow \textsc{Spammer}(v_2)$$
$$\textsc{Prior-Credible}(v) \rightarrow \textsc{Credible}(v)$$
$$\neg\textsc{Prior-Credible}(v) \rightarrow \neg\textsc{Credible}(v)$$
$$\neg\textsc{Spammer}(v)$$

(b) HL-MRFs model that uses the reports and prior credibility of the reporting user to detect spammers. This model would give similar results to assigning total *weighted* counts of the reports for each user as their score of being a spammer.

Figure 5: Simple HL-MRFs models to compare with the collective model shown in Figure 3.

Table 6 shows the results from our three experiments. Using the collective model significantly increases the performance of the reports in detecting spammers. These predictions can be added to the overall classification framework

as a feature. However, since the report graph was sparse relative to the other relation graphs in our dataset, many users could not be classified with this model. Hence, we did not include these predictions as a feature in our framework. This model can be deployed independently to improve the signal from the reports.

Table 6: Classification with collective HL-MRFs model.

Experiment	AUPR	AUROC
Reports (Figure 5a)	0.674±0.008	0.611±0.007
Reports & Credibility (Figure 5b)	0.869±0.006	0.862±0.004
Reports & Credibility & Collective Reasoning (Figure 3)	0.884±0.005	0.873±0.004

6. RELATED WORK

Spam detection in email [22] and the web [23] have been extensively studied, and various methods and features have been proposed for them. Network-based approaches are more closely related to our proposed framework. These methods can be generally categorized based on feature construction and label propagation. Shrivastava et al. [24] generalized the network-based spam detection to random link attacks and showed that the problem is NP-complete. Tseng and Chen [25] used network features to identify email spammers, and incrementally updated the SVM classifier to capture the changes in spam patterns. Oscar and Roychowdbury [26] used a network representation of the emails where nodes were email addresses and links between them indicated a sender-receiver relationship. They used clustering properties of the network to build white and black lists of email addresses and identify spammers. Becchetti et al. [27] proposed a link-based classification for web spam detection, and later combined it with content-based features and used graph topology to improve performance [28]. Since spammers tend to form clusters on the web (unlike in social networks), the authors leveraged clustering and label propagation, to further improve their predictions.

A group of methods are based on label propagation and influenced by PageRank. TrustRank [29] for example, used reputable sites as seeds and propagated reputations through the network. There are multiple variations which propagate dis-trust. Similar to this work, Chirita et al. [30] proposed MailRank which ranked the trustworthiness of a sender based on the network representation of the mail environment. Abernethy et al. [31] proposed a method based on graph regularization and used regularizers that is based on the intuition that linked pages are somewhat similar.

The research focus on spam detection in social networks is relatively more recent. Heymann et al. [32] surveyed different countermeasures to address the spam issue in social networks, and categorized them into methods based on detection, demotion, and prevention. Hu et al. [33] combined information from email, text messages (SMS), and web with Twitter content to detect spammers, and showed improvements in results. Tan et al. [34] proposed an unsupervised spam detection method that focused on identifying a white list of non-spammers from the social network. They argued that legitimate users show more stable patterns in social blogs.

Stein et al. [35] described the spam filtering system in Facebook. They highlighted that attacks on social media use multiple channels, and an effective systems must share feedback and feature data across channels. Gao et al. [36] studied messages between users in Facebook, and used clustering to detect spam campaigns. They identify multiple clusters associated with several campaigns.

Markines et al. [37] studied multiple features and classifiers to detect spam in social tagging systems. Benevenuto et al. [38] used content such as presence or absence of a URL in the post, and user social behaviors such as number of posts to detect spam on Twitter. Lee et al. [39] used honeypots in Twitter and MySpace to harvest deceptive spam profiles. They then used content, posting rate, number of friends, and user demographics such as age and gender as features in their classifier.

Zhu et al. [40] reported that unlike email and web, in social networks, spammers do not form clusters with other spammers, and their neighbors are mostly non-spammers. They use matrix factorization on user activity matrix of data extracted from Renren[6] and use the latent factors as features for classification.

Evolving social networks are of high interest to researchers and have been studied for different purposes [41]. Jin et al. [42] modeled a social network as a *time-stamped heterogeneous network* and used a clustering method to identify spammers. They also used active learning to refine their model. Zhang et al. [43] identified spam campaigns on Twitter by linking accounts with similar malicious URLs in their posts.

Laorden et al. [44] used collective classification to filter spam messages based on their text, to reduce the number of necessary labeled messages. They used implementations in WEKA for collective classification in their evaluation. Geng et al. [45] used a semi-supervised learning algorithm to reduce the labeled training data requirement for web spam detection. Torkamani and Lowd [46] proposed a method to robustly perform collective classification against malicious adversaries that change their behavior in the system.

7. DISCUSSION AND DEPLOYMENT

We have studied the characteristics of time-stamped multi-relational social networks that can be leveraged to detect spammers. We showed that by considering action or relation types and incorporating graph-based features from different relations, one can improve the spammer classification performance. We then showed two sequence mining techniques and their effectiveness to model sequences extracted from time-stamped multi-relational network for spam detection. We also proposed a collective model to refine and improve the signals from the abuse report graph.

Depending on the precision of the results from the model, the security system could either automatically flag a user as spammer and deactivate the account or block its activities in the system, or ask for more verification. Our experimental results show that our model can detect over 65% of the manually detected spammers with higher than 85% precision. These sophisticated spammers had passed the already deployed security measures and performed some activity in the network. Inspecting some of the false positives with the highest spammer probability, we found unlabeled and abandoned spammer accounts, which suggests the real precision

[6]A social network in China: http://renren.com

of the proposed framework might actually be higher than reported. These results can significantly reduce the manual overhead of the administrative security team. Furthermore, our results show that the precision at 80% recall, is above 50%, suggesting this portion of users can be asked for additional verification (e.g., CAPTCHA) without affecting many legitimate users.

This model can be deployed as an iterative batch module to complement real-time filters. Except for some parameters such as the user *credibility prior* for the report system that should be set and adjusted globally for the user, the model has to be re-trained from a fresh sample of the network to adjust to the adversarial changes in patterns. Computing the parameters of the models on a relatively low-powered single machine for our experiments suggests that the framework could be run on very short intervals depending on the training size and computational power. The features can also be computed in parallel. Using *Graphlab Create*^{TM}, computing the features is highly efficient. To provide an example, for a graph with 5.6 million vertices and 350 million edges computing PageRank on our experiment machine took approximately 6.25 minutes, triangle counting 17.98 minutes, k-core 14.3 minutes, and graph coloring 143 minutes.

To optimize the model for production, it is possible to perform feature selection and reduce the necessary features. Feature selection [47] may also improve the performance of the model. Our method for collectively refining the signals from the report graph can be used independently or as a feature in the framework. Improved precision of the predictions via reports enables the system to take actions with more confidence, and reduces the manual overhead.

Our method should be retrained with every new sample. An online learning method that can incorporate the changes in the dynamic network can effectively improve the usability of our framework. Another approach that could improve the prediction results significantly could be incorporating this framework with content-based models. Furthermore, spam accounts often do not act independently and are part of spam campaigns. Their targets may often not be at random as well. They may use a white list of legitimate users to target. Our initial observations show that spammers make relations with legitimate users disproportionally to the overall population ratios. A multi-relational model that can classify spammer accounts based on their target accounts, and identify campaigns based on their relational information could potentially improve the results.

Acknowledgements

Part of this work was performed during the first author's internship at *if(we)* formerly *Tagged Inc.* We are highly grateful to Johann Schleier-Smith and Karl Dawson for their extensive help and support. We also thank Dai Li, Stuart Robinson, Vinit Garg, and Simon Hill for constructive discussions, and Jay Pujara and Arti Ramesh for their helpful suggestions and feedback. We also like to thank the *Dato (Graphlab)* team for their insightful guidance and help with using *Graphlab Create*$^{\text{TM}}$for this project, especially Danny Bickson, Brian Kent, Srikrishna Sridhar, Rajat Arya, Shawn Scully, and Alice Zheng. This work is partially supported by the National Science Foundation (NSF) under contract number IIS0746930. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the supporting institutions.

References

[1] Wikipedia. History of email spam — Wikipedia, the free encyclopedia, 2014. URL http://en.wikipedia.org/wiki/History_of_email_spam.

[2] Harold Nguyen. 2013 state of social media spam. Technical report, Nexgate. URL http://go.nexgate.com/nexgate-social-media-spam-research-report.

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

[4] J Ignacio Alvarez-Hamelin, Luca Dall'Asta, Alain Barrat, and Alessandro Vespignani. Large scale networks fingerprinting and visualization using the k-core decomposition. In *Advances in neural information processing systems (NIPS)*, 2005.

[5] Tommy R Jensen and Bjarne Toft. *Graph coloring problems*. John Wiley & Sons, 2011.

[6] S Pemmaraju and S Skiena. Implementing discrete mathematics: Combinatorics and graph theory with mathematica, 2003.

[7] Thomas Schank. Algorithmic aspects of triangle-based network analysis. *Phd in computer science, University Karlsruhe*, 2007.

[8] Zhengzheng Xing, Jian Pei, and Eamonn Keogh. A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter*, 2010.

[9] Fuchun Peng, Dale Schuurmans, and Shaojun Wang. Augmenting naive Bayes classifiers with statistical language models. *Information Retrieval*, 2004.

[10] Fei Zheng and Geoffrey I Webb. Tree augmented naive Bayes. In *Encyclopedia of Machine Learning*, pages 990–991. Springer, 2010.

[11] Stephen H. Bach, Bert Huang, Ben London, and Lise Getoor. Hinge-loss Markov random fields: Convex inference for structured prediction. In *Uncertainty in Artificial Intelligence (UAI)*, 2013.

[12] S. H. Bach, M. Broecheler, B. Huang, and L. Getoor. Hinge-loss Markov random fields and probabilistic soft logic. arXiv:1505.04406 [cs.LG], 2015.

[13] Jay Pujara, Hui Miao, Lise Getoor, and William Cohen. Knowledge graph identification. In *International Semantic Web Conference (ISWC)*, 2013.

[14] Arti Ramesh, Dan Goldwasser, Bert Huang, Hal Daume III, and Lise Getoor. Learning latent engagement patterns of students in online courses. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI Press, 2014.

[15] Shobeir Fakhraei, Bert Huang, Louiqa Raschid, and Lise Getoor. Network-based drug-target interaction prediction with probabilistic soft logic. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2014.

[16] Shobeir Fakhraei, Louiqa Raschid, and Lise Getoor. Drug-target interaction prediction for drug repurposing with probabilistic similarity logic. In *ACM SIGKDD 12th International Workshop on Data Mining in Bioinformatics (BIOKDD)*. ACM, 2013.

[17] Bert Huang, Angelika Kimmig, Lise Getoor, and Jennifer Golbeck. A flexible framework for probabilistic models of social trust. In *International Conference on Social Computing, Behavioral-Cultural Modeling, & Prediction (SBP)*, 2013.

[18] Nesreen K. Ahmed, Jennifer Neville, and Ramana Kompella. Network sampling: From static to streaming graphs. *ACM Trans. Knowl. Discov. Data*, 2013.

[19] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006.

[20] Mohammad Al Hasan and Mohammed J. Zaki. Output space sampling for graph patterns. *PVLDB*, 2009.

[21] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[22] Enrico Blanzieri and Anton Bryl. A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review*, 2008.

[23] Nikita Spirin and Jiawei Han. Survey on web spam detection: principles and algorithms. *ACM SIGKDD Explorations Newsletter*, 13(2):50–64, 2012.

[24] Nisheeth Shrivastava, Anirban Majumder, and Rajeev Rastogi. Mining (social) network graphs to detect random link attacks. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International conference on*, pages 486–495. IEEE, 2008.

[25] Chi-Yao Tseng and Ming-Syan Chen. Incremental SVM model for spam detection on dynamic email social networks. In *Computational Science and Engineering, 2009. CSE'09. International conference on*, volume 4, pages 128–135. IEEE, 2009.

[26] P Oscar and VP Roychowdbury. Leveraging social networks to fight spam. *IEEE Computer*, 38(4):61–68, 2005.

[27] Luca Becchetti, Carlos Castillo, Debora Donato, Ricardo Baeza-Yates, and Stefano Leonardi. Link analysis for web spam detection. *ACM Transactions on the Web (TWEB)*, 2008.

[28] Carlos Castillo, Debora Donato, Aristides Gionis, Vanessa Murdock, and Fabrizio Silvestri. Know your neighbors: Web spam detection using the web topology. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007.

[29] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with trustrank. In *Proceedings of the thirtieth international conference on very large data bases*, pages 576–587. VLDB Endowment, 2004.

[30] Paul-Alexandru Chirita, Jörg Diederich, and Wolfgang Nejdl. Mailrank: using ranking for spam detection. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 373–380. ACM, 2005.

[31] Jacob Abernethy, Olivier Chapelle, and Carlos Castillo. Graph regularization methods for web spam detection. *Machine Learning*, 81(2):207–225, 2010.

[32] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Fighting spam on social web sites: A survey of approaches and future challenges. *Internet Computing, IEEE*, 11(6):36–45, 2007.

[33] Xia Hu, Jiliang Tang, and Huan Liu. Leveraging knowledge across media for spammer detection in microblogging. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 2014.

[34] Enhua Tan, Lei Guo, Songqing Chen, Xiaodong Zhang, and Yihong Zhao. Unik: unsupervised social network spam detection. In *Proceedings of the 22nd ACM international conference on information & knowledge management*. ACM, 2013.

[35] Tao Stein, Erdong Chen, and Karan Mangla. Facebook immune system. In *Proceedings of the 4th Workshop on Social Network Systems*. ACM, 2011.

[36] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 35–47. ACM, 2010.

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

[38] Fabricio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgilio Almeida. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, volume 6, page 12, 2010.

[39] Kyumin Lee, James Caverlee, and Steve Webb. Uncovering social spammers: social honeypots+ machine learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 435–442. ACM, 2010.

[40] Yin Zhu, Xiao Wang, Erheng Zhong, Nathan N Liu, He Li, and Qiang Yang. Discovering spammers in social networks. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[41] Gueorgi Kossinets and Duncan J Watts. Empirical analysis of an evolving social network. *Science*, 2006.

[42] Xin Jin, Cindy Xide Lin, Jiebo Luo, and Jiawei Han. Socialspamguard: A data mining-based spam detection system for social media networks. *PVLDB*, 2011.

[43] Xianchao Zhang, Shaoping Zhu, and Wenxin Liang. Detecting spam and promoting campaigns in the twitter social network. In *ICDM*, pages 1194–1199, 2012.

[44] Carlos Laorden, Borja Sanz, Igor Santos, Patxi Galán-García, and Pablo G Bringas. Collective classification for spam filtering. *Logic Journal of IGPL*, 2012.

[45] Guang-Gang Geng, Qiudan Li, and Xinchang Zhang. Link based small sample learning for web spam detection. In *Proceedings of the 18th international conference on World wide web*, pages 1185–1186. ACM, 2009.

[46] Mohamadali Torkamani and Daniel Lowd. Convex adversarial collective classification. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 642–650, 2013.

[47] Shobeir Fakhraei, Hamid Soltanian-Zadeh, and Farshad Fotouhi. Bias and stability of single variable classifiers for feature ranking and selection. *Expert Systems with Applications*, 41(15):6945 – 6958, 2014.