NetSpam: a Network-based Spam Detection For Reviews in Online Social Media

[1]Ghanta PavanKalyan,[1]Gorre Srikanth,[1]Gandla Manoj Kumar, [1]Kandula Bhargav, [2]M. Jeevan Babu
[1]Student,[2]Assistant Professor
[1]Computer Science and Engineering,
[1]Vasireddy Venkatadri Institute of Technology, Guntur,AP

Abstract :
Nowadays, a big part of people rely on available contentin social media in their decisions (e.g. reviews and feedbackon a topic or product). The possibility that anybody can leave areview provide a golden opportunity for spammers to write spamreviews about products and services for different interests.Identifying these spammers and the spam content is a hot topic ofresearch and although a considerable number of studies havebeen done recently toward this end, but so far the methodologiesput forth still barely detect spam reviews, and none of them showthe importance of each extracted feature type. In this study, wepropose a novel framework, named NetSpam, which utilizes spamfeatures for modeling review datasets as heterogeneousinformation networks to map spam detection procedure into aclassification problem in such networks. Using the importance ofspam features help us to obtain better results in terms of differentmetrics experimented on real-world review datasets from Yelp andAmazon websites. The results show that NetSpam outperformsthe existing methods and among four categories of features;including review-behavioral, user-behavioral, review-linguistic,user-linguistic, the first type of features performs better than theother categories.
Index Terms—Social Media, Social Network, Spammer, SpamReview, Fake Review, Heterogeneous Information Networks. -

I. INTRODUCTION

Online Social Media portals play an influential role ininformation propagation which is considered as an impor-tantsource for producers in their advertising campaigns as well as forcustomers in selecting products and services. In the past years,people rely a lot on the written reviews in their decision-makingprocesses, and positive/negative reviewsencouraging/discouraging them in their selection of products andservices. In addition, written reviews also help service providers toenhance the quality of their products and services. These reviewsthus have become an important factor in success of a businesswhile positive reviews can bring benefits for a company, negativereviews can potentially impact credibility and cause economiclosses. The fact that anyone with any identity can leave commentsas review, provides a tempting opportunity for spammers to writefake reviews designed to mislead users' opinion. Thesemisleading reviews are then multiplied by the sharing function ofsocial media and prop-agation over the web. The reviews writtento change users' perception of how good a product or a serviceare considered as spam [11], and are often written in exchange formoney.S.R. Shehnepoor is with the University of Tehran, Tehran, Iran. M. Salehi(*corresponding author) is with the University of Tehran, Tehran, Iran. R.Farahbakhsh is with the Institut Mines-Telecom, Telecom SudParis, Paris,France. N. Crespi is with the Institut Mines-Telecom, Telecom Sud-Paris,Paris, France. emails: fshehnepoor@ut.ac.ir, mostafasalehi@ut.ac.ir,reza.farahbakhsh@it-sudparis.eu, noel.crespi@institut-telecom.fr.gAs shown in [1], 20% of the reviews in the Yelp websiteare actually spam reviews.On the other hand, a considerable amount of literaturehas been published on the techniques used to identifyspam and spammers as well as different type of analysison this topic [30], [31]. These techniques can be classifiedinto different categories; some using linguistic patterns intext [2], [3], [4], which are mostly based on bigram, andunigram, others are based on behavioral patterns that relyon features extracted from patterns in users' behaviorwhich are mostly metadata-based [34], [6], [7], [8], [9], andeven some techniques using graphs and graph-basedalgorithms and classifiers [10], [11], [12].Despite this great deal of efforts, many aspects have beenmissed or remained unsolved. One of them is a classifier thatcan calculate feature weights that show each feature's level ofimportance in determining spam reviews. The general conceptof our proposed framework is to model a given review datasetas a Heterogeneous Information Network (HIN) [19] and tomap the problem of spam detection into a HIN classificationproblem. In particular, we model review dataset as a HIN inwhich reviews are connected through different node types(such as features and users). A weighting algorithm is thenemployed to calculate each feature's importance (or weight).These weights are utilized to calculate the final labels forreviews using both unsupervised and supervised approaches.To evaluate the proposed solution, we used two sample review datasets from Yelp and Amazon websites. Based onour observations, defining two views for features (review-userand behavioral-linguistic), the classified features as reviewbehavioralhave more weights and yield better performance onspotting spam reviews in both semi-supervised and unsupervisedapproaches. In addition, we demonstrate that usingdifferent supervisions such as 1%, 2.5% and 5% or using anunsupervised approach, make no noticeable variation on theperformance of our approach. We observed that feature weights can be added or removed for labeling and hence timecomplexity can be scaled for a specific level of accuracy. Asthe result of this weighting step, we can use fewer featureswith more weights to obtain better accuracy with less time complexity. In addition, categorizing features in four majorcategories (review-behavioral, user-behavioral, reviewlinguistic,user-linguistic), helps us to understand how mucheach category of features is contributed to spam detection.In summary, our main contributions are as follows:(i) We propose NetSpam framework that is a novel networkbasedapproach which models review networks as heterogeneousinformation networks. The classification step usesdifferent metapath types which are innovative in thespam detection domain.(ii) A new weighting method for spam features is proposedto determine the relative importance of each featureand shows how effective each of features are in identifyingspams from normal reviews. Previous works [12], [20] alsoaimed to address the importance of features mainly in termof obtained accuracy, but not as a build-in function in their framework (i.e., their approach is

dependent to ground truthfor determining each feature importance). As we explain inour unsupervised approach, NetSpam is able to findfeatures importance even without ground truth, and only byrelying on metapath definition and based on valuescalculated for each review.(iii) NetSpam improves the accuracy compared to the stateof-the art in terms of time complexity, which highly depends tothe number of features used to identify a spam review; hence,using features with more weights will resulted in detecting fakereviews easier with less time complexity.

II. PRELIMINARIES

As mentioned earlier, we model the problem as a heterogeneous network where nodes are either real components in a dataset (such as reviews, users and products) or spam features. To better understand the proposed framework we first present an overview of some of the concepts and definitions in heterogeneous information networks [23], [22], [24].

A. DEFINITIONS

In terms of security, we assume a semi-honest cloud server, which is interested in learning about stored data but will follow our keyword search protocol as described and will not modify or misrepresent any data in order to gain an advantage. Two of the main security issues regarding keyword searches are the privacy of the document sets and the privacy of the queried keywords. Briefly, a secure keyword search protocol should prevent the cloud server from obtaining non-negligible amount of information on the stored documents or the keywords in the query requests. Note that, in our target application, users are employees of the data owner's organization and are authorized to search for any documents in the data set. Should an application requires that users be restricted from accessing certain files, an access control system such as [20] would be required to verify the matched results and returned only those which the user has the required credential to access. Our basic scheme in section 4.2 achieves these goals under the assumption that the cloud has no prior knowl- edge on the stored data. Should the cloud provider has significant statistical knowledge on the stored data, such as the distribution of the keywords, it may be able to infer partial knowledge on its content. Under the security model where the cloud provider has some knowledge over the distribution of keywords or queries on the stored data, we describe modifications to the basic scheme which would offer protection against statistical attacks in section 4.6 and inclusion-relation attacks in section 4.4.

Definition 1 (Heterogeneous Information Network). Suppose we have r(> 1) types of nodes and s(> 1) types of relation links between the nodes, then a heterogeneous information network is defined as a graph G = (V; E) where each node v 2 V and each link e 2 E belongs to one particular node type and link type respectively. If two links belong to the same type, the types of starting node and ending node of those links are the same.

Definition 2 (Network Schema). Given a heterogeneous information network G = (V; E), a network schema T = (A; R) is a metapath with the object type mapping : V ! A and link mapping E ! R, which is a graph defined over object type A, with links as relations from R. The schema describes the metastructure of a given network (i.e., how many node types there are and where the possible links exist).

Definition 3 (Metapath). As mentioned above, there are no edges between two nodes of the same type, but thereare paths. Given a heterogeneous information network G = (V; E), a metapath P is defined by a sequence of relations in the network schema T = (A; R), denoted in the form A1(R1)A2(R2):::(R(l 1))Al, which defines a composite relation P = R1oR2o:::oR(l 1) between two nodes, where o is the composition operator on relations. For convenience, a metapath can be represented by a sequence of node types when there is no ambiguity, i.e., P = A1A2:::Al. The metapath extends the concept of link types to path types and describes the different relations among node types through indirect links, i.e. paths, and also implies diverse semantics.

Definition 4 (Classification problem in heterogeneousinformation networks). Given a heterogeneous informationnetwork G = (V; E), suppose V 0 is a subset of V thatcontains nodes of the target type (i.e., the type of nodes tobe classified). k denotes the number of the class, and foreach class, say C1:::Ck, we have some pre-labeled nodesin V 0 associated with a single user. The classification taskis to predict the labels for all the unlabeled nodes in V 0.

B. Feature Types

In this paper, we use an extended definition of the metapath concept as follows. A metapath is defined as a path between two nodes, which indicates the connection of two nodes through their shared features. When we talk about metadata, we refer to its general definition, which is data about data. In our case, the data is the written review, and by metadata we mean data about the reviews, including user who wrote the review, the business that the review is written for, rating value of the review, date of written review and finally its label as spam or genuine review. In particular, in this work features for users and reviews fall into the categories as follows (shown in Table I): Review-Behavioral (RB) based features. This feature type is based on metadata and not the review text itself. The RB category contains two features; Early time frame (ETF) and Threshold rating deviation of review (DEV) [16]. Review-Linguistic (RL) based features. Features in this category are based on the review itself and extracted directly from text of the review. In this work we use two main features in RL category; the Ratio of 1st Personal Pronouns (PP1) and the Ratio of exclamation sentences containing '!' (RES) [6]. User-Behavioral (UB) based features. These features are specific to each individual user and they are calculated per user, so we can use these features to generalize all of the reviews written by that specific user. This category has two main features; the Burstiness of reviews written by a single user [7], and the average of a users' negative ratio given to different businesses [20]. User-Linguistic (UL) based features. These features are extracted from the users' language and shows how users are describing their feeling or opinion about what they've experienced as a customer of a certain business. We use this type of features to understand how a spammer communicates in terms of wording. There are two features engaged for our framework in this category; Average Content Similarity (ACS) and Maximum Content Similarity (MCS). These two features show how much two reviews written by two different users are similar to each other, as spammers tend to write very similar reviews by using template pre-written text [11].

III. NETSPAM; THE PROPOSED SOLUTION

In this section, we provides details of the proposed solution which is shown in Algorithm III.1.

A. Prior Knowledge

The first step is computing prior knowledge, i.e. the initial probability of review u being spam which denoted as yu. The

TABLE I: Features for users and reviews in four defined categories (the calculated values are based on Table 2 in [12])

Spam Feature	User-based	Review-based
Behavioral-Based Features	Burstiness [20]: Spammers, usually write their spam reviews in short period of time for two reasons: first, because they want to impact readers and other users, and second because they are temporal users, they have to write as much as reviews they can in short time. $x_{BST}(i) = 0 \quad (L_i \quad F_i) 2 = (0;) \quad (1 \quad L_i \quad F_i \quad (L_i \quad F_i) \quad (0;) \quad 2$ (1) where $L_i \quad F_i$ describes days between last and first review for = 28. Users with calculated value greater than 0.5 take value 1 and others take 0. Negative Ratio [20]: Spammers tend to write reviews which defame businesses which are competitor with the ones they have contract with, this can be done with destructive reviews, or with rating those businesses with low scores. Hence, ratio of their scores tends to be low. Users with average rate equal to 2 or 1 take 1 and others take 0.	Early Time Frame [16]: Spammers try to write their reviews asap, in order to keep their review in the top reviews which other users visit them sooner. $x_{ETF}(i) = 0 \quad (T_i \quad F_i) 2 = (0;)$ (2) $(1 \quad T_i \quad F_i \quad (T_i \quad F_i) \quad (0;) \quad 2$ where $L_i \quad F_i$ denotes days specified written review and first written review for a specific business. We have also = 7. Users with calculated value greater than 0.5 takes value 1 and others take 0. Rate Deviation using threshold [16]: Spammers, also tend to promote businesses they have contract with, so they rate these businesses with high scores. In result, there is high diversity in their given scores to different businesses which is the reason they have high variance and deviation. $x_{DEV}(i) = (\quad 0 \quad \text{otherwise} \quad 1 \quad \frac{r_{ij}^{avg} e2E_i^{r(e)}}{4} > _1$ (3) where $_1$ is some threshold determined by recursive minimal entropy partitioning. Reviews are close to each other based on their calculated value, take same values (in [0; 1)).
Linguistic-Based Features	Average Content Similarity [7], Maximum Content Similarity [16]: Spammers, often write their reviews with same template and they prefer not to waste their time to write an original review. In result, they have similar reviews. Users have close calculated values take same values (in [0; 1)).	Number of first Person Pronouns, Ratio of Exclamation Sentences containing '!' [6]: First, studies show that spammers use second personal pronouns much more than first personal pronouns. In addition, spammers put '!' in their sentences as much as they can to increase impression on users and highlight their reviews among other ones. Reviews are close to each other based on their calculated value, take same values (in [0; 1)).

proposed framework works in two versions; semi-supervised learning and unsupervised learning. In the semi-supervised method, yu = 1 if review u is labeled as spam in the pre-labeled reviews, otherwise yu = 0. If the label of this review is unknown due the amount of supervision, we consider yu = 0 (i.e., we assume u as a non-spam review). In the unsupervised method, our prior knowledge is realized by using PL yu = (1=L) l=1 f(xlu) where f(xlu) is the probability of review u being spam according to feature l and L is the number of all the used features (for details, refer to [12]).

B. Network Schema Definition

The next step is defining network schema based on a given list of spam features which determines the features engaged in spam detection. This Schema are general definitions of meta-paths and show in general how different network components are connected. For example, if the list of features includes NR, ACS, PP1 and ETF, the output schema is as presented in Fig. 1.

C. Metapath Definition and Creation

As mentioned in Section II-A, a metapath is defined by a sequence of relations in the network schema. Table II shows all the metapaths used in the proposed framework. As shown, the length of user-based metapaths is 4 and the length of review-based metapaths is 2. For metapath creation, we define an extended version of the metapath concept considering different levels of spam

certainty. In particular, two reviews are connected to each other if they share same value. Hassanzadeh et al. [25] propose a fuzzy-based framework and indicate for spam detection, it is better to use fuzzy logic for determining a review's label as a

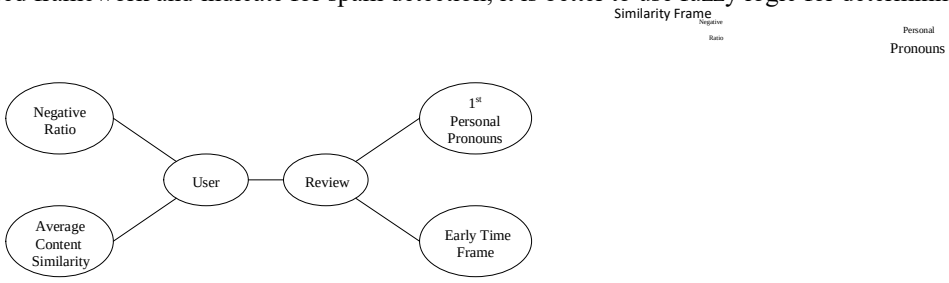


Fig. 1: An example for a network schema generated based on a given spam features list; NR, ACS, PP1 and ETF.

spam or non-spam. Indeed, there are different levels of spam certainty. We use a step function to determine these levels. In particular, given a review u, the levels of spam certainty for
metapathpl (i.e., feature l) is calculated as mp l = bs f(xlu)c , u s
where s denotes the number of levels. After computing mpul for all reviews and metapaths, two reviews u and v with the same metapath values (i.e., mpul = mpvl) for metapathpl are connected to each other through that metapath and create one link of review network. The metapath value between them denoted as mpu;vl = mpul .
Using s with a higher value will increase the number of each feature's metapaths and hence fewer reviews would be connected to each other through these features. Conversely, using lower value for s leads us to have bipolar values (which means reviews take value 0 or 1). Since we need enough spam and non-spam reviews for each step, with fewer number of reviews connected to each other for every step, the spam probability of reviews take uniform distribution, but with lower value of s we have enough reviews to calculate final spamicity for each review. Therefore, accuracy for lower levels of s decreases because of the bipolar problem, and it decades for higher values of s, because they take uniform distribution. In the proposed framework, we considered s = 20, i.e.

ALGORITHM III.1: NETSPAM()

Input : review dataset; spam f eature list; pre labeled reviews
Output : f eatures importance(W); spamicity probability(P r)
% u; v: review, yu: spamicity probability of review u
% f(xlu): initial probability of review u being spam
% pl: metapath based on feature l, L: features number
% n: number of reviews connected to a review
% mpul : the level of spam certainty
% mpu;vl: the metapath value
%Prior Knowledge
if semi-supervised mode
if u = pre labeled reviews
yu = label(u)
Else
yu = 0
else % unsupervised mode
yu = 1 L PL l=1 f(xlu)
%Network Schema Definition
schema = defining schema based on spam-feature-list
% Metapath Definition and Creation
forpl∈ schema
for u, v ∈ review − dataset
do
mpl u = bs×f(xlu)c/s
mpl v = bs×f(xlv)c/ s
ifmpl u = mpl
vmpplu,v = mpl u
else mpplu,v = 0
% Classification - Weight Calculation
forpl∈ schemes
doWpl = Pn r=1 Pn s=1 mppl P r,s×yr×ys/ n r=1 Pn s=1 mpplr,s
% Classification – Labeling

for u, v ∈ review − dataset
do P ru,v = 1 − ΠL pl=11 − mpplu,v × Wpl
P ru = avg(P ru,1, P ru,2, ..., P ru,n)
return (W, Pr).

D. Classification

The classification part of NetSpam includes two steps; (i) weight calculation which determines the importance of each spam feature in spotting spam reviews, (ii) Labeling which calculates the final probability of each review being spam. Next we describe them in detail.

1) **Weight Calculation:** This step computes the weight of each metapath. We assume that nodes' classification is done based on their relations to other nodes in the review network; linked nodes may have a high probability of taking the same labels. The relations in a heterogeneous information network not only include the direct link but also the path that can be measured by using the metapath concept. Therefore, we need to utilize the metapaths defined in the previous step, which represent heterogeneous relations among nodes. Moreover, this step will be able to compute the weight of each relation path (i.e., the importance of the metapath), which will be used in the next step (Labeling) to estimate the label of each unlabeled review. The weights of the metapaths will answer an important question; which metapath (i.e., spam feature) is better at ranking spam reviews? Moreover, the weights help us to understand the formation mechanism of a spam review. In addition, since some of these spam features may incur considerable computational costs (for example, computing linguistic-based features through NLP methods in a large review dataset), choosing the more valuable features in the spam detection procedure leads to better performance whenever the computation cost is an issue. To compute the weight of metapath pi , for i = 1, ..., L where L is the number of metapaths, we propose following equation:

Wpi = Pn r=1 Pn s=1 mppir,s × yr × ys/ Pn r=1 Pn s=1 mp pi r,s

Where n denotes the number of reviews and mppir,s is a metapath value between reviews r and s if there is a path between them through metapath pi , otherwise mppir,s = 0. Moreover, yr(ys) is 1 if review r(s) is labeled as spam in the pre-labeled reviews, otherwise 0. 2) Labeling: Let P ru,v be the probability of unlabeled review u being spam by considering its relationship with spam review v. To estimate P ru, the probability of unlabeled review u being spam, we propose the following equations:

P ru,v = 1 − Π L i=11 − mppiu,v × Wpi
P ru = avg(P ru,1, P ru,2, ..., P ru,n)

where n denotes number of reviews connected to review u. Fig. 2 shows an example of a review network and different steps of proposed framework. It is worth to note that in creating the HIN, as much as the number of links between a review and other reviews increase, its probability to have a label similar to them increase too, because it assumes that a node relation to other nodes show their similarity. In particular, more links between a node and other non-spam reviews, more probability for a review to be non-spam and vice versa. In other words, if a review has lots of links with non-spam reviews, it means that it shares features with other reviews with low spamicity and hence its probability to be a non-spam review increases.

IV. EXPERIMENTAL EVALUATION

This section presents the experimental evaluation part of this study including the datasets and the defined metrics as well as the obtained results.

TABLE II: Metapaths used in the NetSpam framework

Row	Notation	Type	MetaPath	Semantic
1	R-DEV-R	RB	Review-Threshold Rate Deviation-Review	Reviews with same Rate Deviation from average Item rate (based on recursive minimal entropy partitioning)
2	R-U-NR-U-R	UB	Review-User-Negative Ratio-User-Review	Reviews written by different Users with same Negative Ratio
3	R-ETF-R	RB	Review-Early Time Frame-Review	Reviews with same released date related to Item
4	R-U-BST-U-R	UB	Review-User-Burstiness-User-Review	Reviews written by different users in same Burst
5	R-RES-R	RL	Review-Ratio of Exclamation Sentences containing '!'-Review	Reviews with same number of Exclamation Sentences containing '!'
6	R-PP1-R	RL	Review-first Person Pronouns-Review	Reviews with same number of first Person Pronouns
7	R-U-ACS-U-R	UL	Review-User-Average Content Similarity-User-Review	Reviews written by different Users with same Average Content Similarity using cosine similarity score
8	R-U-MCS-U-R	UL	Review-User-Maximum Content Similarity-User-Review	Reviews written by different Users with same Maximum Content Similarity using cosine similarity score

A. Datasets

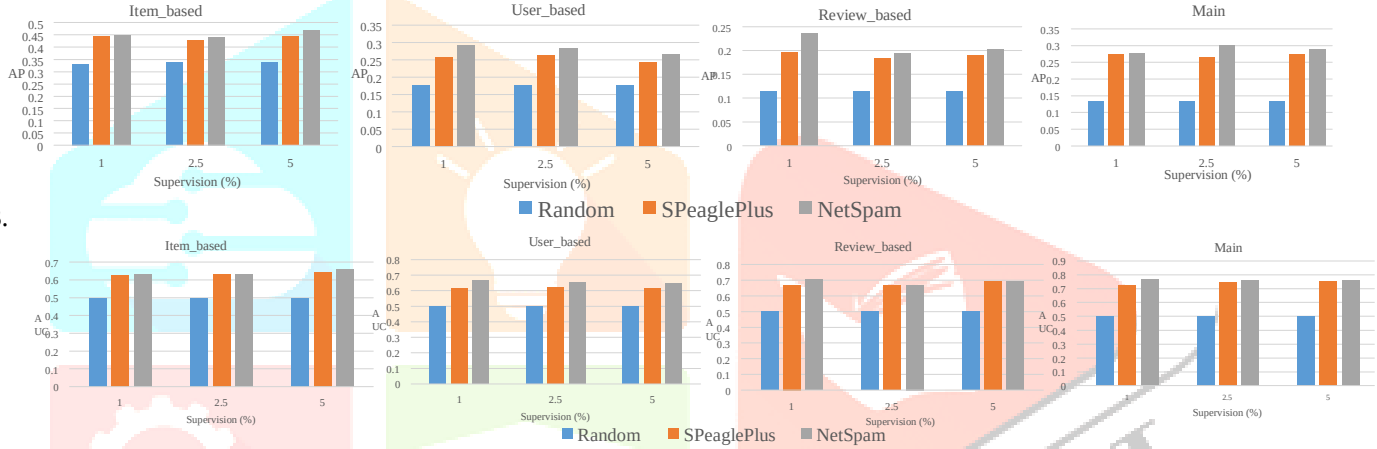
Table III includes a summary of the datasets and their characteristics. We used a dataset from Yelp, introduced in [12], which includes almost 608,598 reviews written by customers of restaurants and hotels in NYC. The dataset includes the reviewers' impressions and comments about the quality, and other aspects related to a restaurants (or hotels). The dataset also contains labeled reviews as ground truth (so-called near ground-truth [12]), which indicates whether a review is spam or not. Yelp dataset was labeled using filtering algorithm engaged by the Yelp recommender, and although none of recommenders are perfect, but according to [36] it produces trustable results. It explains hiring someone to write different fake reviews on different social media sites, it is the yelp algorithm that can spot spam reviews and rank one specific spammer at the top of spammers. Other attributes in the dataset are rate of reviewers, the date of the written review, and date of actual visit, as well as the user's and the restaurant's id (name).

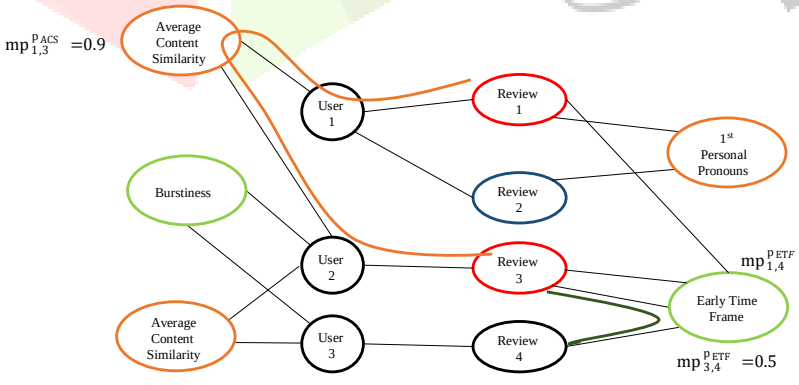
We created three other datasets from this main dataset as follow: -

Review-based dataset, includes 10% of the reviews from the Main dataset, randomly selected using uniform distribution. - Item-based dataset, composes of 10% of the randomly selected reviews of each item, also based on uniform distribution (as with Review-based dataset).

- User-based dataset, includes randomly selected reviews using uniform distribution in which one review is selected from every 10 reviews of single user and if number of reviews was less than 10, uniform distribution has been changed in order to at least one review from every user get selected.

In addition to the presented dataset, we also used another real-world set of data from Amazon [34] to evaluate our work on unsupervised mode. There is no credible label in the Amazon dataset (as mentioned in [35]), but we used this dataset to show how much our idea is viable on other datasets beyond Yelp and results for this dataset is presented on Sec. IV-C3.





B. Evaluation Metrics

We have used Average Precision (AP) and Area Under the Curve (AUC) as two metrics in our evaluation. AUC measures accuracy of our ranking based on False Positive Ratio (FPR

TABLE III: Review datasets used in this work.

Dataset	Reviews (spam%)	Users	Business (Resto. & hotels)
Main	608,598 (13%)	260,277	5,044
Review-based	62,990 (13%)	48,121	3,278

Item-based	66,841 (34%)	52,453	4,588
User-based	183,963 (19%)	150,278	4,568
Amazon	8,160 (-)	7685	243

as y-axis) against True Positive Ratio (TPR as x-axis) and integrate values based on these two measured values. The value of this metric increases as the proposed method performs well in ranking, and vise-versa. Let A be the list of sorted spam reviews so that A(i) denotes a review sorted on the ith index in A. If the number of spam (non-spam) reviews before review in the j th index is equal to nj and the total number of spam (non-spam) reviews is equal to f, then TPR (FPR) for the j th is computed as nj f . To calculate the AUC, we set T P R values as the x-axis and F P R values on the y-axis and then integrate the area under the curve for the curve that uses their values. We obtain a value for the AUC using:

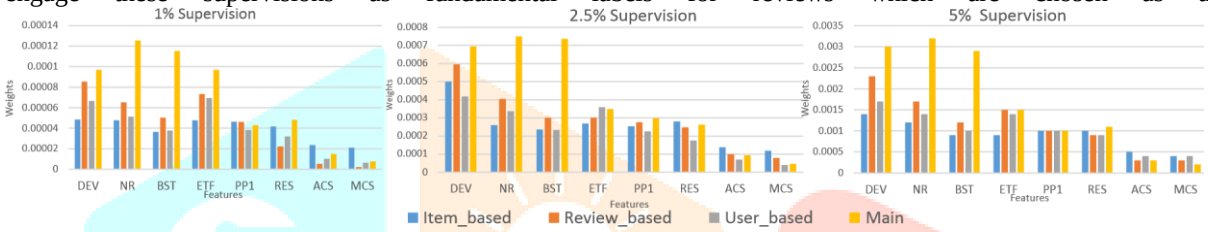
$$AUC = Xni=2 (F P R(i) - F P R(i - 1)) * (T P R(i))$$

where n denotes number of reviews. For AP we first need to calculate index of top sorted reviews with spam labels. Let indexes of sorted spam reviews in list A with spam labels in ground truth be like list I, then for AP we have:

$$AP = Xni=1 iI(i)$$

As the first step, two metrics are rank-based which means we can rank the final probabilities. Next we calculate the AP and AUC values based on the reviews' ranking in the final list. In the most optimum situation, all of the spam reviews are ranked on top of sorted list;

In other words, when we sort spam probabilities for reviews, all of the reviews with spam labels are located on top of the list and ranked as the first reviews. With this assumption we can calculate the AP and AUC values. They are both highly dependent on the number of features. For the learning process, we use different supervisions and we train a set for weight calculation. We also engage these supervisions as fundamental labels for reviews which are chosen as a training set.



B. Main Results

In this section, we evaluate NetSpam from different perspective and compare it with two other approaches, Random approach and SPeaglePlus [12]. To compare with the first one, we have developed a network in which reviews are connected to each other randomly. Second approach use a wellknown graph-based algorithm called as "LBP" to calculate final labels. Our observations show NetSpam, outperforms these existing methods. Then analysis on our observation is performed and finally we will examine our framework in unsupervised mode. Lastly, we investigate time complexity of the proposed framework and the impact of camouflage strategy on its performance.

1) **Accuracy**: Figures 3 and 4 present the performance in terms of the AP and AUC. As it's shown in all of the four datasets NetSpam outperforms SPeaglePlusspecially when number of features increase. In addition different supervisions have no considerable effect on the metric values neither on NetSpam nor SPeaglePlus. Results also show the datasets with higher percentage of spam reviews have better performance because when fraction of spam reviews in a certain dataset increases, probability for a review to be a spam review increases and as a result more spam reviews will be labeled as spam reviews and in the result of AP measure which is highly dependent on spam percentage in a dataset. On the other hand, AUC measure does not fluctuate too much, because this metric is not dependent on spam reviews percentage in dataset, but on the final sorted list which is calculated based on the final spam probability.

2) **Feature Weights Analysis:** Next we discuss about features weights and their involvement to determine spamicity. First we inspect how much AP and AUC are dependent on variable number of features. Then we show these metrics

TABLE IV: Weights of all features (with 5% data as train set); features are ranked based on their overall average weights.

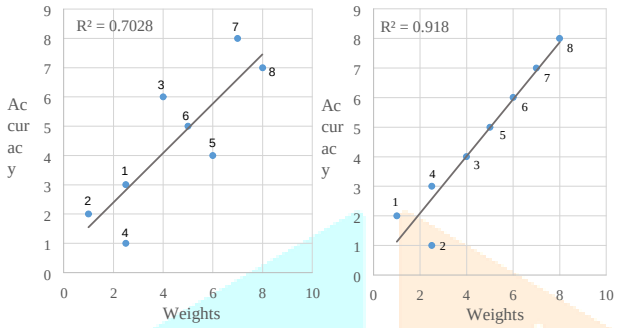
Dataset - Weights	DEV	NR	ETF	BST	RES	PP1	ACS	MCS
Main	0.0029	0.0032	0.0015	0.0029	0.0010	0.0011	0.0003	0.0002
Review-based	0.0023	0.0017	0.0017	0.0015	0.0010	0.0009	0.0004	0.0003
Item-based	0.0010	0.0012	0.0009	0.0009	0.0010	0.0010	0.0004	0.0003
User-based	0.0017	0.0014	0.0014	0.0010	0.0010	0.0009	0.0005	0.0004

are different for the four feature types explained before (RB, UB, RL and UL). To show how much our work on weights calculation is effective, first we have simulated framework on several run with whole features and used most weighted features to find out best combination which gives us the best results. Finally, we found which category is most effective category among those listed in Table I.

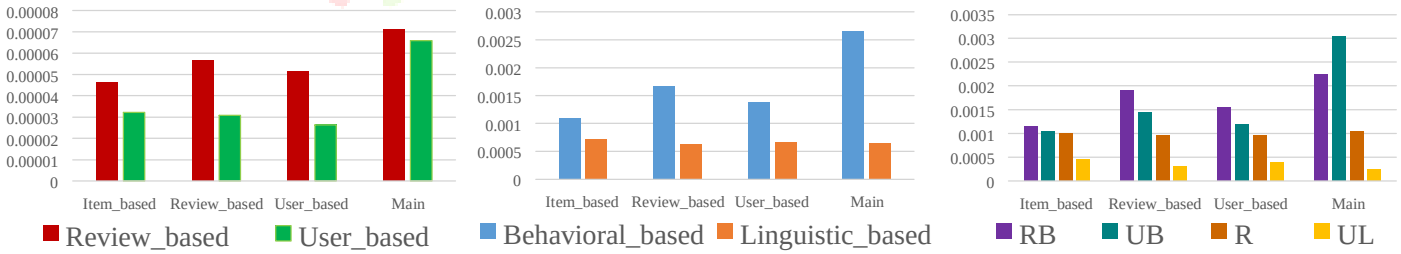
Dataset Impression on Spam Detection: As we explained previously, different datasets yield different results based on their contents. For all datasets and most weighted features, there is a certain sequence for features weights. As is shown in Fig. 5 for four datasets, in almost all of them, features for the Main dataset have more weights and features for Review-based dataset stand in the second position. Third position belongs to User-based dataset and finally Item-based dataset has the minimum weights (for at least the four features with most weights).

Features Weights Importance: As shown in Table IV, there are couple of features which are more weighted than others. Combination of these features can be a good hint for obtaining better performance. The results of the Main dataset show all the

four behavioral features are ranked as first features in the final overall weights. In addition, as shown in the Reviewbased as well as other two datasets, DEV is the most weighted feature. This is also same for our second most weighted feature, NR. From the third feature to the last feature there are different order for the mentioned features. The third feature for both datasets User-based and Review-based is same, ET F, while for the other dataset, Item-based, P P1 is at rank 3. Going further, we see in the Review-based dataset all four most weighted features are behavioral-based features which shows how much this type of features are important in detecting spams as acknowledged by other works as well [12], [20]. As we can see in Fig. 6, there is a strong correlation between features weights and the accuracy. For the Main dataset we can see this correlation is much more obvious and also applicable. Calculating weights using NetSpam help us to understand how much a feature is effective in detecting spam reviews; since as much as their weights increase two metrics including AP and AUC also increase respectively and therefore our framework can be helpful in detecting spam reviews based on features importance.



The observations indicate larger datasets yield better correlation between features weights and also its accuracy in term of AP. Since we need to know each feature rank and importance we use Spearman's rank correlation for our work. In this experience our main dataset has correlation value equal to 0.838 (p-value=0.009), while this value for our next dataset, User-based one, is equal to 0.715 (p-value = 0.046). As much as the size of dataset gets smaller in the experiment, this value drops. This problem is more obvious in Item and Review-based datasets. For Item-based dataset, correlation value is 0.458 which is low, because sampling Item-based dataset needs Item-based features. The features are identical to each item and are similar to user-based features. Finally the obtained results for our smallest dataset is satisfying, because final results considering AP show a correlation near to 0.683 between weights and accuracy (similar results for SPeaglePlus as well). Weights and accuracy (in terms of AP) are completely correlated. We observed values 0.958 (pvalue=0.0001), 0.764 (p=0.0274), 0.711 (p=0.0481) and 0.874 (p=0.0045) for the Main, User-based, Item-based and Reviewbased datasets, respectively. This result shows using weight calculation method and considering metapath concept can be effective in determining the importance of features. Similar result for SPeaglePlus also shows our weights calculation method can be generalized to other frameworks and can beused as a main component for finding each feature weight.

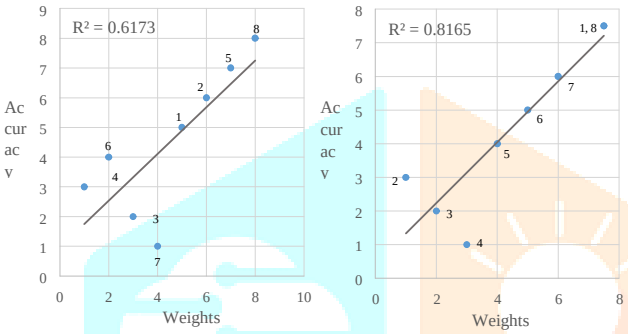


Our results also indicate feature weights are completely dependent on datasets, considering this fact two most important features in all datasets are same features. This means except the first two features, other features weights are highly variable regrading to dataset used for extracting weights of features.

Features Category Analysis: As shown in Fig. 7 there are four categories with different weights average which is very important, specially in determining which feature is more appropriate for spotting spam reviews (refer to Sec. IV-C2). Since results for different supervision are similar we have just presented the results for 5% supervision. We have analyzed features based on their categories and obtained results in all datasets show that Behavioral-based features have better weights than linguistic ones which is confirmed by [16] and [12]. Analysis on separate views shows that reviewbased features have higher weights which leads to better

performance. It is worth to mention that none of previous works have investigated this before. Same analysis on the Main dataset shows equal importance of both category in finding spams. On the Other hand, in the first three dataset from Table I, RB has better weights (a bit difference in comparison with RU), which means this category yields better performance than other categories for spotting spam reviews. Differently, for Main dataset UB categories has better weights and has better performance than RU category and also other categories, in all datasets behavioral-based features yield better performance with any supervision

3) Unsupervised Method: One of the achievement in this study is that even without using a train set, we can still find the best set of features which yield to the best performance. As it is explained in Sec. III-A, in unsupervised approach special formulation is used to calculate fundamental labels and next these labels are used to calculate the features' weight and finally review labels. As shown in Fig. 8, our observations show there is a good correlation in the Main dataset in which for NetSpam it is equal to 0.78 (p-value=0.0208) and for SPeaglePlus this value reach 0.90 (p=0.0021). As another example for user-based dataset there is a correlation equal to 0.93 (p=0.0006) for NetSpam, while for SPeagle this value is equal to 0.89 (p=0.0024). This observation indicates NetSpam can prioritize features for both frameworks. Table V demonstrates that there is certain sequence in feature weights and it means in spam detection problems, spammers and spam reviews have common behaviors, no matter what social network they are writing the review for: Amazon or Yelp. For all of them, DEV is most weighted features, followed by NR, ET F and BST.



4)Time Complexity: If we consider the Main dataset as input to our framework, time complexity with these circumstances is equal to O(e 2m) where e is number of edges in created network or reviews number. It means we need to check if there is a metapath between a certain node (review) with other nodes which is O(e 2) and this checking must be repeated for very feature. So, our time complexity for offline mode in which we give the Main dataset to framework and calculate spamicity of whole reviews, is O(e 2m) where m is number of features. In online mode, a review is given to NetSpam to see whether it is spam or not, we need to check if there is a metapath between given review with other reviews, which is in O(e), and like offline mode it has to be repeated for every feature and every value. Therefore the complexity is O(em).

5) The Impact of Camouflage Strategy: One of the challenges that spam detection approaches face is that spammers often write non-spam reviews to hide their true identity known as camouflage. For example they write positive reviews for good restaurant or negative reviews for low-quality ones; hence every spam detector system fails to identify this kind of spammers or at least has some trouble to spot them. In the previous studies, there are different approaches for handling this problem. For example, in [12], the authors assumes there is always a little probability that a good review written by a spammer and put this assumption in its compatibility matrix. In this study, we tried to handle this problem by using weighted metapaths. In particular, we assume that even if a review has a very little value for a certain feature, it is considered in feature weights calculation. Therefore, instead

TABLE V: Weights of all features (using unsupervised approach); features are ranked based on their overall average weights.

Dataset - Weights	DEV	NR	ETF	BST	RES	PP1	ACS	MCS
Main	0.0029	0.0550	0.0484	0.0445	0.0379	0.0329	0.0321	0.0314
Review-based	0.0626	0.0510	0.0477	0.0376	0.0355	0.0346	0.0349	0.0340
Item-based	0.0638	0.0510	0.0501	0.0395	0.0388	0.0383	0.0374	0.0366
User-based	0.0630	0.0514	0.0494	0.0380	0.0373	0.0377	0.0367	0.0367
Amazon	0.1102	0.0897	0.0746	0.0689	0.0675	0.0624	0.0342	0.0297

of considering metapaths as binary concepts, we take 20 values which denoted as s. Indeed, if there is a camouflage its affection will be reduced. As we explained in Section III-C in such problems it is better to propose a fuzzy framework, rather than using a bipolar values (0, 1).

V. RELATED WORKS

In the last decade, a great number of research studies focus on the problem of spotting spammers and spam reviews. However, since the problem is non-trivial and challenging, it remains far from fully solved. We can summarize our discussion about previous studies in three following categories.

A. Linguistic-based Methods

This approach extract linguistic-based features to find spam reviews. Feng et al. [13] use unigram, bigram and their composition. Other studies [4], [6], [15] use other features like pairwise features (features between two reviews; e.g. content similarity), percentage of CAPITAL words in a reviews for finding spam reviews. Lai et al. in [33] use a probabilistic language modeling to spot spam. This study demonstrates that 2% of reviews written on business websites are actually spam.

B. Behavior-based Methods

Approaches in this group almost use reviews metadata to extract features; those which are normal pattern of a reviewer behaviors. Feng et al. in [21] focus on distribution of spammers rating on different products and traces them. In [34], Jindal et. al extract 36 behavioral features and use a supervised method to find spammers on Amazon and [14] indicates behavioral features show spammers' identity better than linguistic ones. Xue et al. in [32] use rate deviation of a specific user and use a trust-aware model to find the relationship between users for calculating final spamicity score. Minnich et al. in [8] use temporal and location features of users to find unusual behavior of spammers. Li et al. in [10] use some basic features (e.g polarity of reviews) and then run a HNC (Heterogeneous Network Classifier) to find final labels on Dianpings dataset. Mukherjee et al. in [16] almost engage behavioral features like rate deviation, extremity and etc. Xie et al. in [17] also use a temporal pattern (time window) to find singleton reviews (reviews written just once) on Amazon. Luca et al. in [26] use behavioral features to show increasing competition between companies leads to very large expansion of spam reviews on products.

Crawford et al. in [28] indicates using different classifi- cation approach need different number of features to attain desired performance and propose approaches which use fewer features to attain that performance and hence recommend toimprove their performance while they use fewer features which leads them to have better complexity. With this perspective our framework is arguable. This study shows using different approaches in classification yield different performance in terms of different metrics.

C. Graph-based Methods

Studies in this group aim to make a graph between users, reviews and items and use connections in the graph and also some network-based algorithms to rank or label reviews (as spam or genuine) and users (as spammer or honest). Akoglu et al. in [11] use a network-based algorithm known as LBP (Loopy Belief Propagation) in linearly scalable iterations related to number of edges to find final probabilities for different components in network. Fei et al. in [7] also use same algorithm (LBP), and utilize burstiness of each review to find spammers and spam reviews on Amazon. Li et al. in [10] build a graph of users, reviews, users IP and indicates users with same IP have same labels, for example if a user with multiple different account and same IP writes some reviews, they are supposed to have same label. Wang et al. in [18] also create a network of users, reviews and items and use basic assumptions (for example a reviewer is more trustworthy if he/she writes more honest reviews) and label reviews. Wahyuni in [27] proposes a hybrid method for spam detection using an algorithm called ICF++ which is an extension to ICF of [18] in which just review rating are used to find spam detection. This work use also sentiment analysis to achieve better accuracy in particular.

Deeper analysis on literature show that behavioral features work better than linguistic ones in term of accuracy they yield. There is a good explanation for that; in general, spammers tend to hide their identity for security reasons. Therefore they are hardly recognized by reviews they write about products, but their behavior is still unusual, no matter what language they are writing. In result, researchers combined both feature types to increase accuracy of spam detection. The fact that adding each feature is a time consuming process, this is where feature importance is useful. Based on our knowledge, there is no previous method which engage importance of features (known as weights in our proposed framework; NetSpam) in the classification step. By using these weights, on one hand we involve features importance in calculating final labels and hence accuracy of NetSpam increase, gradually. On the other hand we can determine which feature can provide better performance in term of their involvement in connecting spam reviews (in proposed network).

VI. CONCLUSION

This study introduces a novel spam detection framework namely NetSpam based on a metapath concept as well asa new graph-based method to label reviews relying on a rank-based labeling approach. The performance of the proposed framework is evaluated by using two real-world labeled datasets of Yelp and Amazon websites. Our observations show that calculated weights by using this metapath concept can be very effective in identifying spam reviews and leads to a better performance. In addition, we found that even without a train set, NetSpam can calculate the importance of each feature and it yields better performance in the features' addition process, and performs better than previous works, with only a small number of features. Moreover, after defining four main categories for features our observations show that the reviewsbehavioral category performs better than other categories, in terms of AP, AUC as well as in the calculated weights. The results also confirm that using different supervisions, similar to the semi-supervised method, have no noticeable effect on determining most of the weighted features, just as in different datasets.

For future work, metapath concept can be applied to other problems in this field. For example, similar framework can be used to find spammer communities. For finding community, reviews can be connected through group spammer features (such as the proposed feature in [29]) and reviews with highest similarity based on metapth concept are known as communities. In addition, utilizing the product features is an interesting future work on this study as we used features more related to spotting spammers and spam reviews. Moreover, while single networks has received considerable attention from various disciplines for over a decade, information diffusion and content sharing in multilayer networks is still a young research [37]. Addressing the problem of spam detection in such networks can be considered as a new research line in this field.

VII. ACKNOWLEDGMENT

This work is partially supported by Iran National Science Foundation (INSF) (Grant No. 94017889).

REFERENCES

[1] J. Donfro, A whopping 20 % of yelp reviews are fake. http://www.businessinsider.com/20-percent-of-yelp-reviews-fake-2013-9. Accessed: 2015-07-30.

[2] M. Ott, C. Cardie, and J. T. Hancock. Estimating the prevalence of deception in online review communities. In ACM WWW, 2012.

[3] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock. Finding deceptive opinion spam by any stretch of the imagination.In ACL, 2011.

[4] Ch. Xu and J. Zhang. Combating product review spam campaigns via multiple heterogeneous pairwise features. In SIAM International Conference on Data Mining, 2014.

[5] N. Jindal and B. Liu. Opinion spam and analysis. In WSDM, 2008.

[6] F. Li, M. Huang, Y. Yang, and X. Zhu. Learning to identify review spam. Proceedings of the 22nd International Joint Conference on Artificial Intelligence; IJCAI, 2011.

[7] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh. Exploiting burstiness in reviews for review spammer detection. In ICWSM, 2013.

[8] A. j. Minnich, N. Chavoshi, A. Mueen, S. Luan, and M. Faloutsos. Trueview: Harnessing the power of multiple review sites. In ACM WWW, 2015.

[9] B. Viswanath, M. Ahmad Bashir, M. Crovella, S. Guah, K. P. Gummadi, B. Krishnamurthy, and A. Mislove. Towards detecting anomalous user behavior in online social networks. In USENIX, 2014.

[10] H. Li, Z. Chen, B. Liu, X. Wei, and J. Shao. Spotting fake reviews via collective PU learning. In ICDM, 2014

[11] L. Akoglu, R. Chandy, and C. Faloutsos. Opinion fraud detection in online reviews bynetwork effects. In ICWSM, 2013.

[12] R. Shebuti and L. Akoglu. Collective opinion spam detection: bridging review networksand metadata. In ACM KDD, 2015.

[13] S. Feng, R. Banerjee and Y. Choi. Syntactic stylometry for deception detection. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers; ACL, 2012

. [14] N. Jindal, B. Liu, and E.-P. Lim. Finding unusual review patterns using unexpected rules. In ACM CIKM, 2012.

[15] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw. Detecting product review spammers using rating behaviors. In ACM CIKM, 2010.

[16] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh. Spotting opinion spammers using behavioral footprints. In ACM KDD, 2013.

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

[18] G. Wang, S. Xie, B. Liu, and P. S. Yu. Review graph based online store review spammer detection. IEEE ICDM, 2011.

[19] Y. Sun and J. Han. Mining Heterogeneous Information Networks; Principles and Methodologies, In ICCCE, 2012.

[20] A. Mukerjee, V. Venkataraman, B. Liu, and N. Glance. What Yelp Fake Review Filter Might Be Doing?, In ICWSM, 2013.

[21] S. Feng, L. Xing, A. Gogar, and Y. Choi. Distributional footprints of deceptive product reviews. In ICWSM, 2012.

[22] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In VLDB, 2011

. [23] Y. Sun and J. Han. Rankclus: integrating clustering with ranking for heterogeneous information network analysis. In Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, 2009.

[24] C. Luo, R. Guan, Z. Wang, and C. Lin. HetPathMine: A Novel Transductive Classification Algorithm on Heterogeneous Information Networks. In ECIR, 2014

[25] R. Hassanzadeh. Anomaly Detection in Online Social Networks: Using Datamining Techniques and Fuzzy Logic. Queensland University of Technology, Nov. 2014.

[26] M. Luca and G. Zervas. Fake It Till You Make It: Reputation, Competition, and Yelp Review Fraud., SSRN Electronic Journal, 2016.

[27] E. D. Wahyuni and A. Djunaidy. Fake Review Detection From a Product Review Using Modified Method of Iterative Computation Framework. In Proceeding MATEC Web of Conferences. 2016.

[28] M. Crawford, T. M. Khoshgoftaar, and J. D. Prusa. Reducing Feature set Explosion to Faciliate Real-World Review Sapm Detection. In Proceeding of 29th International Florida Artificial Intelligence Research Society Conference. 2016.

[29] A. Mukherjee, B. Liu, and N. Glance. Spotting Fake Reviewer Groups in Consumer Reviews. In ACM WWW, 2012.
[30] A. Heydari, M. A. Tavakoli, N. Salim, and Z. Heydari. Detection of review spam: A survey. Expert Systems with Applicants, Elsevier, 2014.
[31] M. Crawford, T. D. Khoshgoftar, J. N. Prusa, A. Al. Ritcher, and H. Najada. Survey of Review Spam Detection Using Machine Learning Techniques. Journal of Big Data. 2015.
[32] H. Xue, F. Li, H. Seo, and R. Pluretti. Trust-Aware Review Spam Detection. IEEE Trustcom/ISPA . 2015.
[33] C. L. Lai, K. Q. Xu, R. Lau, Y. Li, and L. Jing. Toward a Language Modeling Approach for Consumer Review Spam Detection. In Proceedings of the 7th international conference on e-Business Engineering. 2011.
[34] N. Jindal and B. Liu. Opinion Spam and Analysis. In WSDM, 2008.
[35] S. Mukherjee, S. Dutta, and G. Weikum. Credible Review Detection with Limited Information using Consistency Features, In book: Machine Learning and Knowledge Discovery in Databases, 2016.
[36] K. Weise. A Lie Detector Test for Online Reviewers. http://bloom.bg/1KAxzhK. Accessed: 2016-12-16.
[37] M. Salehi, R. Sharma, M. Marzolla, M. Magnani, P. Siyari, and D. Montesi. Spreading processes in multilayer networks. In IEEE Transactions on Network Science and Engineering. 2(2):65–83, 2015