NetSpam: a Network-based Spam Uncovering Context for Analyses in Operational Social Media

[1]Prof. Pravin M. Dhanrao, [2]Prof. Harshal S. Sangale, [3]Prof. Kiran B. Mate, [4]Prof. Amit V. Shejwal

[1]Lecturer, [2](MTech. App.), Lecturer, [3](MSc Maths) Lecturer, (MSc Maths) Lecturer

[1]Department of Computer Technology, [2]Department of Information Technology, [3]Department of Civil Engineering, [4]Department of Electrical Engineering

[1,2,3,4]Sanjivani K. B. P. Polytechnic Kopargaon, Maharashtra, India.

[1]pravindhanrao@gmail.com, [2]hssangaleit@sanjivani.org.in, [3]kbmnighoj@gmail.com, [4]amitshejwal1211@gmail.com

Abstract **Nowadays, a big part of people rely on available content in social media in their decisions (e.g. reviews and feedback on a topic or product). The possibility that anybody can leave a review provide a golden opportunity for spammers to write spam reviews about products and services for different interests. Identifying these spammers and the spam content is a hot topic of research and although a considerable number of studies have been done recently toward this end, but so far the methodologies put forth still barely detect spam reviews, and none of them show the importance of each extracted feature type. In this study, we propose a novel framework, named *NetSpam*, which utilizes spam features for modeling review datasets as heterogeneous information networks to map spam detection procedure into a classification problem in such networks. Using the importance of spam features help us to obtain better results in terms of different metrics experimented on real-world review datasets from Yelp and Amazon websites. The results show that *NetSpam* outperforms the existing methods and among four categories of features; including review-behavioral, user-behavioral, reviewlinguistic, user-linguistic, the first type of features performs better than the other categories.**

Keywords — Social Media, Social Network, Spammer, Spam Review, Fake Review, Heterogeneous Information Networks.

I. INTRODUCTION

Online Social Media portals play an influential role in information propagation which is considered as an important source for producers in their advertising campaigns as well as for customers in selecting products and services. In the past years, people rely a lot on the written reviews in their decision-making processes, and positive/negative reviews encouraging/discouraging them in their selection of products and services. In addition, written reviews also help service providers to enhance the quality of their products and services. These reviews thus have become an important factor in success of a business while positive reviews can bring benefits for a company, negative reviews can potentially impact credibility and cause economic losses. The fact that anyone with any identity can leave comments as review, provides a tempting opportunity for spammers to write fake reviews designed to mislead users' opinion. These misleading reviews are then multiplied by the sharing function of social media and propagation over the web. The reviews written to change users' perception of how good a product or a service are considered as spam [11], and are often written in exchange for money.

As shown in [1], 20% of the reviews in the Yelp website are actually spam reviews.

On the other hand, a considerable amount of literature has been published on the techniques used to identify spam and spammers as well as different type of analysis on this topic [30], [31]. These techniques can be classified into different categories; some using linguistic patterns in text [2], [3], [4], which are mostly based on bigram, and unigram, others are based on behavioral patterns that rely on features extracted from patterns in users' behavior which are mostly metadatabased [34], [6], [7], [8], [9], and even some techniques using graphs and graph-based algorithms and classifiers [10], [11], [12].

Despite this great deal of efforts, many aspects have been missed or remained unsolved. One of them is a classifier that can calculate feature weights that show each feature's level of importance in determining spam reviews. The

 DOI : 10.18231/2454-9150.2019.0303

general concept of our proposed framework is to model a given review dataset as a Heterogeneous Information Network (HIN) [19] and to map the problem of spam detection into a HIN classification problem. In particular, we model review dataset as a HIN in which reviews are connected through different node types (such as features and users). A weighting algorithm is then employed to calculate each feature's importance (or weight). These weights are utilized to calculate the final labels for reviews using both unsupervised and supervised approaches.

To evaluate the proposed solution, we used two sample review datasets from Yelp and Amazon websites. Based on our observations, defining two views for features (review-user and behavioral-linguistic), the classified features as reviewbehavioral have more weights and yield better performance on spotting spam reviews in both semi-supervised and unsupervised approaches.

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

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 $v \in V$ and each link $e \in E$ 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 $\tau : V \rightarrow A$ and link mapping $\varphi : E \rightarrow R$, 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 there are 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

$A_1(R_1)A_2(R_2)...(R_{(l-1)})A_l$, which defines a composite relation $P = R_1 o R_2 o ... o R_{(l-1)}$ 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., $P = A_1A_2...A_l$. 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 heterogeneous information networks). Given a heterogeneous information network $G = (V,E)$, suppose V^0 is a subset of V that contains nodes of the target type (i.e., the type of nodes to be classified). k denotes the number of the class, and for each class, say $C_1...C_k$, we have some pre-labeled nodes in V^0 associated with a single user. The classification task is 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.

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 y_u. The

proposed framework works in two versions; semi-supervised learning and unsupervised learning. In the semi-supervised method, $y_u = 1$ if review u is labeled as spam in the pre-labeled reviews, otherwise $y_u = 0$. If the label of this review is unknown due the amount of supervision, we consider $y_u = 0$ (i.e., we assume u as a non-spam review). In the unsupervised method, our prior knowledge is realized by using $y_u = (1/L)\sum_{l=1}^{L} f(x_{lu})$ where $f(x_{lu})$ is the probability of

 DOI : 10.18231/2454-9150.2019.0303

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 metapaths 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 reviewbased 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

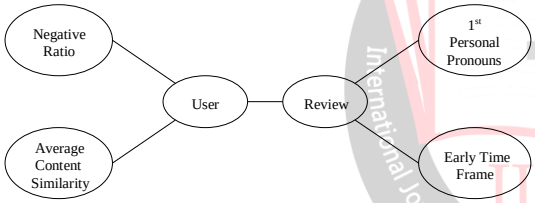

Fig. 1: An example for a network schema generated based on a given spam features list; NR, ACS, PP1 and ETF.

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 metapath p_l (i.e., feature *l*) is calculated as $m_u^{p_l} = \frac{\lfloor s \times f(x_{lu}) \rfloor}{s}$, where *s* denotes the number of levels. After computing $m^{p}{}_{u}^{l}$ for all reviews and metapaths, two reviews *u* and *v* with the same metapath values (i.e., $m_u^{p_l} = m_v^{p_l}$) for metapath p_l are connected to each other through that metapath and create one link of review network. The metapath value between them denoted as $m_{u,v}^{p_l} = m_u^{p_l}$.

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.

$m^{p}{}_{u}^{l} \in \{0, 0.05, 0.10, \ldots, 0.85, 0.90, 0.95\}$.

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.

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.

 DOI : 10.18231/2454-9150.2019.0303

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 i^{th} index in A. If the number of spam (non-spam) reviews before review in the j^{th} index is equal to n_j and the total number of spam (non-spam) reviews is equal to f, then TPR (FPR) for the j^{th} is computed as $\frac{n_j}{f}$. To calculate the AUC, we set TPR values as the *x-axis* and FPR 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:

$$AUC = \sum_{i=2}^{n} (FPR(i) - FPR(i-1)) * (TPR(i)) \quad (7)$$

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:

$$AP = \sum_{i=1}^{n} \frac{i}{I(i)} \quad (8)$$

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.

C. 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 SPeaglePlus specially 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 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

 DOI : 10.18231/2454-9150.2019.0303

finally Item-based dataset has the minimum weights (for at least the four features with most weights).

AUC also increase respectively and therefore our framework can be helpful in detecting spam reviews based on features

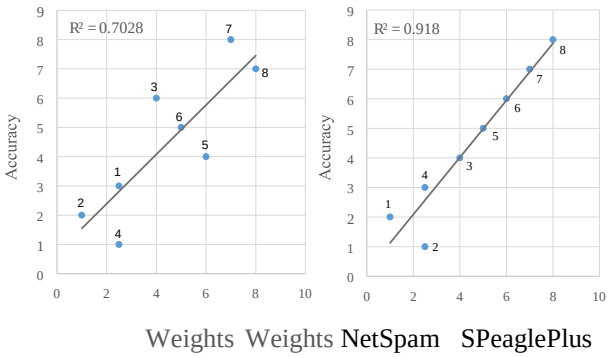

Fig. 6: Regression graph of features vs. accuracy (with 5% data as train set) for *Main* dataset. (see Table II for numbers)importance.The observations indicate larger datasets yield better correlation between features weights and also its accuracy in term of AP. Since we need to know each feature rank 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. 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.

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*, *ETF* and *BST*.

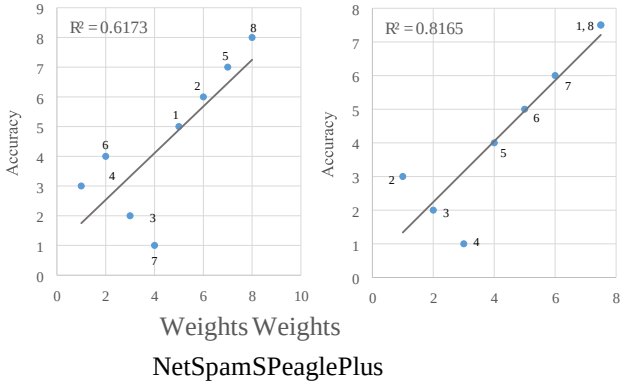

Fig. 8: Regression graph of features vs. accuracy

(unsupervised) for *Main* dataset. (see Table II for numbers)

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 $O(em)$.

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]

 DOI : 10.18231/2454-9150.2019.0303

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 classification approach need different number of features to attain desired performance and propose approaches which use fewer features to attain that performance and hence recommend to improve 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.

VI. CONCLUSION

This study introduces a novel spam detection framework namely *NetSpam* based on a metapath concept as well as a 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.

REFERENCES

[1] J. Donfro, A whopping20% 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.

 DOI : 10.18231/2454-9150.2019.0303

[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.

 DOI : 10.18231/2454-9150.2019.0303