R-SALSA: A Spam Filtering Technique for Social Networking Sites

Mohit Agrawal
Department of Computer Science and Engineering
National Institute of Technology
Tiruchirappalli, India
mohit.agrawal619@gmail.com

R. Leela Velusamy
Department of Computer Science and Engineering
National Institute of Technology
Tiruchirappalli, India
leela@nitt.edu

Abstract— **Now a days, Social media is instrumental for expeditious communication among users across the globe. The escalation in the growth of Social media tools such as LinkedIn, Google+, MySpace, Pinterest, Facebook, Instagram, Twitter, Yammer, Weibo, Hyves, etc., led to rise in the volume of unsolicited messages and spamming activities in past few years. Enormous volume of spamming activities has caused severe problem in essential communication. Spam messages may be generated by automated spam bots or users. In vision of these circumstances, there has been much research effort toward doing spam filtering based on supervised approaches. Motivated by the fact, steady nature of supervised approach requires model retraining to identify new variety of spam messages. An unsupervised approach namely Reliability based Stochastic Approach for Link-Structure Analysis (R-SALSA) algorithm has been proposed in this paper for classifying a message being Spam or benign. The dataset collected from popular Netherland's social media named Hyves is used to test proposed algorithm. It has been evaluated with different performance based metrics namely true positive rate, false positive rate, accuracy, and it is found to be performing better than previously proposed unsupervised author-reporter model. The proposed algorithm achieved 9.17% accuracy in spam identification when compared with Hyper Link Induced Topic Search (HITS) and 2.49% accuracy in spam identification when compared with SALSA based method.**

Keywords— Markov chain, Page Rank, SALSA, Social media, Social spam, SVM, botnet.

I. INTRODUCTION

A social network is a dyadic tie among social actors (such as individuals or an organization) [1]. Information sharing among the user is one of the main objectives of social networking sites. The exponential growth of social media namely Twitter, LinkedIn, Google+, MySpace, Pinterest, Facebook, Instagram, Yammer etc., along with its local immutable such as Weibo, Hyves etc., has been impelled with the proliferation of Web 2.0 and its user friendly characteristics [2]. The popularity of social networks has made them a central platform for many unsolicited activities such as Link farming, Phishing, Sybil attack, malware distribution, Spamming etc.

Social spam is an e-crime on social networking sites with contents such as comments, post, chat, etc. There are many spamming activities going through social media such as malicious links posting, insulting posts, hate speech, fake friends, deceitful reviews, etc. Motivation of these spamming activities can be either private or commercial. Previously, e-mails were the major object of spammers however it is slowly reduced with the advancement of spam filters that can filter almost 95% of spam content mails. On the other hand, growth of social networking sites and its weak security measures attracted many spammers and made it a vigorous field of concern for research community. In past decade, lot of research activity has been performed to control spamming activities using supervised and unsupervised mechanisms. But typical enhancement of technologies made spamming actions difficult to tackle with existing mechanisms. This made the researchers to come up with the advanced mechanism improving the existing one through further modification. Majority of the social media rely on the user community to tackle with spam issues because of its dynamicity. In past few years, enormous supervised approaches had been proposed for spam filtering but immutable nature of these approaches needs the model to be retrained every time to classify a new variety of spam which is inefficient. In this paper, the existent unsupervised model has been improvised with the addition of reliability factor and a Reliability based SALSA (R-SALSA) algorithm has been proposed.

The data analysis technique used to evaluate relationship among different nodes is termed as link analysis. Link based ranking algorithm can be divided into components namely query independent and query dependent algorithms. Recent research has found that, Query dependent algorithms (SALSA) performs better than query independent algorithms such as Page Rank [5]. SALSA is a link based algorithm, which was proposed by R.Lempel and S.Morgan [3]. SALSA inherits the basic feature of HITS and Page rank algorithms and uses random walk across chains of hubs and authorities based on the concept of Markov chain. Previously, SALSA algorithm has been used for trusty account recommendation in Twitter [4]. SALSA can also be used for spam filtering [18].

Spam-filtering techniques can be either supervised or unsupervised. Based on a large number of features, a general model of spam filtering is built in supervised spam filtering

978-1-4673-7918-2/16/$31.00 ©2016 IEEE

technique. Building supervised systems that cannot be outwitted by spammers is impractical. Unsupervised approaches track individual operation and may thus aim at narrow class of spam, assuming that once such class becomes noticeable it is likely to continue in similar form for a while. Unfortunately, when relying on complaint volume, by the time the system react to such reports, the spam operation might affect many users. Even though various supervised and semi-supervised spam-filtering approaches exist, goal of designing highly efficient spam filtering technique remains elusive [6]. Therefore SALSA based unsupervised model for spam classification is combined with user reliability and R-SALSA is proposed. This paper focuses on the advantages of unsupervised methods over supervised measures for spam classification. Factors such as true positive rate, false positive rate, and accuracy of proposed algorithm has been compared with existing mechanisms such as HITS and SALSA and the achieved improvement of proposed model is highlighted.

Section II explains the related work on spam filtering techniques. The proposed spam filtering technique using R-SALSA is described in Section III. Section IV compares the proposed algorithm with the existing algorithms. Finally, the research work has been concluded and future research directions is discussed in Section V.

II. Related Work

In a survey of existing machine learning mechanisms for detection of review based spam in social media, Michael Crawford et. al., [7] provided a comparative study of existing research on spam detection using various machine learning techniques. The survey basically splits review centric features into several categories such as combination of bag of words with term frequency feature, linguistic inquiry and word count output (LIWC), frequency of parts of speech (POS), slylometric and syntactic features, and review characteristic features.

The application differences between e-mail, social and web spams, and the anti-spam methodologies based on Rank, Limit, Identification has been highlighted by P. Heymann et. al., [8]. Ismaila Idris et. al., [9] improvised the existing negative selection algorithm (NSA) with the combination of Particle swarm optimization (PSO). The local outlier factor has been used as a fitness function. In this method, for threshold 0.4 NSA-PSO obtained 82.77% negative predictive value outperforming the existing NSA model with 66.24%.

Yudong Zhang et. al., [10] proposed a spam filtering method namely binary PSO with mutation operator (MBPSO) for feature selection. In this model, decision tree is chosen as a classifier model with the training algorithm C4.5. The achieved specificity, sensitivity and accuracy are 97.51%, 91.02%, and 94.27% respectively.

The taxonomy of botnet behavior, its detection and defense has been surveyed by Sheharbano Khattak et. al., [11]. The relevant examples have been augmented to provide a real-world context. Haiying shen et. al., [12] proposed a mechanism namely social network aided personalized and effective spam filter (SOAP). In this mechanism, social network links and overlay links has been used to form a distributed overlay. Each node gathers the information and detects spam using SOAP. The dataset from the popular social networking site namely Facebook is tested with SOAP and found to be better compared to keyword parsing methodologies (Bayesian spam filters).

M. Bosma et. al., [13] introduced the first unsupervised approach for spam identification. The framework is based on the web link analysis algorithm namely HITS. The link between the user and messages are the base of the proposed framework. The messages are classified to be either spam or ham based on its likeliness and trust worthiness.

A near real time detection system for spam detection called as WarningBird has been proposed by Sangho Lee et. al., [14]. The ineffectiveness of conventional feature based detection mechanism of twitter against feature fabrication has been highlighted. In the proposed mechanism, a redirect chain of several tweets is extracted and the correlation Study is performed because of the fact that spammers have limited resources and usually they reuse them. A correlated URL redirect chain using frequently shared URLs is discovered and suspiciousness of URLs was identified.

M. McCord et. al., [15] used the traditional classifier based on user-based content-based features to classify spammers and legitimate users. Random forest classifier has been used for achieving 95.7% precision and 95.7% F-measure.

F. Benevenuto et. al., [16] provided a heuristic for classifying an arbitrary video as legitimate or spam. In the proposed method, the dataset of Youtube users is collected and manually classified to be spam or legitimate. Support Vector Machine (SVM) is used as a data classifier with a 5-fold cross validation. The mechanism achieved a true positive rate of 43.9% and the accuracy of 87% for detecting spam.

A comparative analysis between HITS and SALSA has been carried out by Mark Najork et. al., [17] and it is found that sampling of nodes and edges made SALSA more effective. The existing supervised methods lack detection of new kind of spam because of its static nature. In order to overcome the above drawback, an unsupervised method namely SALSA was proposed. The proposed SALSA based algorithm is further improvised with reliability factor and named as R-SALSA which can predict spam more accurately compared to existing methods.

III. Proposed Work

This Section explains the proposed model for spam filtering along with its functionality. The proposed filtering model is based on SALSA algorithm combined with the reliability of reporters and renamed as R-SALSA. The SALSA algorithm uses the links between user and the messages to calculate spam score for each message. The proposed model considers the reliability factor of reporters and the spam scores of messages together to detect the spam content messages. Section 3.1 introduces the Spam filtering model based on R-SALSA followed by its Implementation detail in Algorithm 1.

A. Detection Model

In past years, numerous spam filtering techniques based on supervised approach has been proposed. The growth and dynamicity of spam messages demands model retraining to detect a new variety of spam. To avoid this retraining overhead, an unsupervised mechanism of spam filtering has been proposed in this paper. This approach avoids the retraining of model with new features of spam messages rather calculates the authenticity of user reporting a message to be spam or not. This paper improves SALSA by adding a new parameter namely reliability factor denoted as 'α' which can be calculated as the ratio between correct spam report by reporters to total number of reporters reporting content as spam.

In the initial stage, the algorithm creates a bilateral graph of reporters $R=\{r_1, r_2, r_3,\ldots, r_n\}$ and contents $C=\{ c_1, c_2, c_3,\ldots, c_n\}$. Where one disjoint set of the graph represents reporters and another set represents message content connected by directed edges (E) representing the spam reports issued from reporter set to content set as depicted in Fig.1.

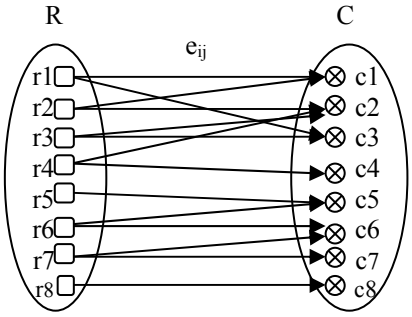

Fig. 1. Sample graph for Proposed Model

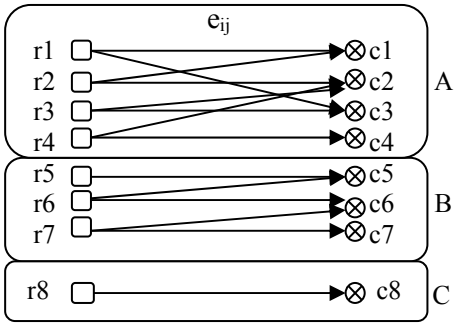

Fig. 2. Finding Sub-component using SALSA.

Fig.1 denotes a bilateral graph with two disjoint sets among which set R represents the set of reporters reporting a message to be spam and set C represents the set of messages. The edge connecting both disjoint set indicates the reported spam with a weight $e_{i,j}$ representing weight of edge from r_i to c_j. The proposed algorithm is inspired by well-known link analysis algorithm namely SALSA.

It executes a random walk across the chain of content represented in the bilateral graph. The procedure combines the frontward and rearward link traversal to calculate the spam scores for each message in set C. Initially, the weight of the edge and reliability factor is initialized with a uniform value. Then a random movement is performed across the chain of messages in set C and it is fragmented into sub-components as shown in Fig. 2. The random walk through a set finds different linked components of set C by means of prevalent messages between reporters as shown in Fig. 2.

After the division of content set into sub-components, the inward directed edges has been used to calculate the in-degree for each message in C and represented as Indegree(c_i). The sum of inward edges of all c_i in a sub-component represents the total in-degree in a component and represented as Compt_indegree. Then the calculated indegree of each c_i is divided with its Compt_indegree to find normalized in-degree. Compt_size represents the size of a component and can be evaluated as the total number of messages in that component. Similarly, Tot_size symbolizes the size of set C, and can be calculated as sum of Compt_size for each component. The relative size of the component can be evaluated as ratio among Compt_size to Tot_size. Initially, SALSA is executed in the training set containing combination

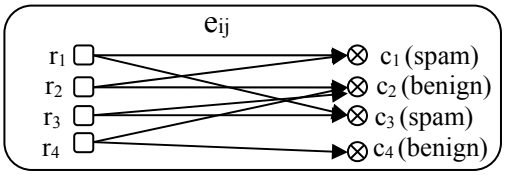

Fig. 3. Sample labeled graph to compare SALSA and R-SALSA.

of labeled and unlabelled data together with the initialized null value of reliability factor α_i for r_i. Then α_i is being updated with the new value representing the ratio among correctly classified spam out of overall spam reports by r_i. The motivation behind the reliability factor is that, some unreliable reporters may report the benign messages as spam for their own benefits which may lead to increase in False Positive Rate of the filtering model. So, the reliability factor of each reporter has been added with the spam score of each messages to calculate the reliable spam score and the obtained results are made known.

$$Spam\_score = \left(\frac{in\deg ree(c_i)}{Comp\_in\deg ree}\right) * \left(\frac{Comp\_size}{Total\_size}\right) \quad (1)$$

$$\text{Re}liable\_Spam\_score(c_i) = \alpha_i + p * q \quad (2)$$

where, $p = \frac{in\deg ree(c_i)}{Comp\_in\deg ree}$, $q = \frac{Comp\_size}{Total\_size}$

$$\alpha_i = \frac{\left(\sum_{j=1}^{n} \frac{correctly\_identified\_spam}{total\_reports}\right)}{num\_of\_reports} \quad (3)$$

where, j represents reliability factor for j^{th} reporter for i^{th} content.

Equation 1 represents the spam score calculation for messages using SALSA [18] and Equation 2 is proposed

method for reliable spam filtering with the introduction of reliability factor. The spam score for each message in the sample graph as shown in Fig. 3 has been calculated using both Equation 1 and Equation 2. The detailed explanation for spam score calculation and working of proposed model is as follows:

1) *Spam Score Calculation Using SALSA:*

Indegree(c_1)=Count of inward edges to c_1=2.
Compt_indegree=Count of inward edges to A=8.
Compt_size=Count of messages in A=4.
Tot_size=Size of content set C=4+3+1=8.
Using Equation 1:
Spam_sco(c_1)= (2/8)(4/8) = 0.125.*

Similarly,
Indegree(c_2)=Count of inward edges to c_2=3.
Compt_indegree=Count of inward edges to A=8.
Compt_size=Count of messages in A=4.
Tot_size=|C|=4+3+1=8.
Using Equation 1:
Spam_sco(c_2)= (3/8)(4/8) = 0.1875.*

2) *Spam Score Calculation Using R-SALSA:*

Indegree(c_1)=Count of inward edges to c_1=2.
Compt_indegree=Count of inward edges to A=8.
Compt_size=Count of messages in A=4.
Tot_size=Size of set C =4+3+1=8.
α_1= Reliabilty factor for c_1=((1)+(1/2))/2= 0.75.
Using Equation 2:
Reliable_Spam_sco(c_1)= 0.75+(2/8)(4/8) = 0.875.*

Similarly,
Indegree(c_2)=Count of inward edges to c_2=3.
Compt_indegree=Count of inward edges to A=8.
Compt_size=Count of messages in A=4.
Tot_size= Size of set C =4+3+1=8.
α_2=Reliability factor for c_2=((1/2)+(1/2)+0)/3= 0.3333.
Using Equation 2:
Reliable_Spam_sco(c_2)=0.3333+(3/8)(4/8)= 0.5208.*

The use of reliability factor in the proposed algorithm is justified with the above calculation of spam scores with both methods. Using SALSA algorithm in Fig. 3, the spam score of spam message c_1 is found to have low compared to benign message c_2. But after inclusion of R-SALSA with a reliability factor the calculated spam score for spam message is high. So, the introduction of reliability factor is shown to be helpful for detecting the spam content messages more accurately by reducing the false positive rate.
In the similar way reliable spam scores for other messages can be measured using the proposed algorithm described in the next Section.

B. Implementation and Observations

The dataset from popular Dutch social network hyves has been collected and tested on the proposed algorithm defined in Algorithm 1. Java language was used for implementation. The behavior of dataset is investigated and it is found to be sparse in nature. Section IV elaborates the detail feature and nature of dataset used.

Algorithm 1. Pseudo-code for R-SALSA to calculate spam score.

Input: Bilateral graph of reporter and messages with spam reports.
Output: Reliable_spam _score for messages in content set.

```
R-SALSA_spam_score()
begin
  call Calc_size();   // To calculate total size of set C.
       for each compt do
       Compt_indegree:=0;

          for each  node n in compt c do
              for each edge m in inward_edge(ni) do
                        Compt_indegree:=Comp_indegree+Edge_wt(mi);
                        //where mi represents edge to message i
              end
            end
            for each message n in compt c do
              Norm_indegree(ni):=indegree(ni)/Compt_indegree;
                    //where  n1,n2,n3,n4 represents messages of set C
              Relat_size:=Compt_size/Tot_size;
              Rel_factor:= Calc_reliability(); //calculating α
              Norm:=Norm_indegree(ni);
               Reliable_Spam_sco(ni):=Rel_factor+(Norm*Relat_size);
            end
            return Reliable_Spam_sco(ni);
       end
end

Calc_size()
begin
     Tot_size:=0;
     for each compt do
        Tot_size:=Tot_size+Compt_size;
     end
     return Tot_size;
end

Calc_reliability()
begin
     Rel_factor:=0;
     Num_reporters:=0;
     for each reporter r in compt c do
        if(ri report spam) then
          //where r1,r2,r3 represents reporter 1,2,3 respectively
           Rel_factor:=Rel_factor+ identified_spam/Tot_reports;
                //identified_spam and Tot_reports in Training set
             Num_reporters++;
        end
      end
      return (Rel_factor/Num_reporters);
end
```

After collection of dataset the analysis has been done and it is found that 90% of data is having one spam report. Nearly

5% of data consists of two spam reports. Rest of the information consists of three to four spam reports.

The collected dataset is transformed into directed bilateral graph as shown in Fig. 1. Using the proposed algorithm namely R-SALSA, the content set is further divided into sub-components and spam score for each content message is calculated using Equation 2. The algorithm has been tested using various performance parameters and the obtained TPR, FPR, ACC are 88.40%, 8.72%, 89.25% respectively.

IV. Experiment and Resuls

A. Data Collection

The dataset used for the testing purpose has been crawled for the period of Jan 2010 to Jan 2011 from the popular dutch social networking site named Hyves. The dataset contains 9,491 reporters, 13,188 messages and 28,998 spam reports. It consists of two sub-divisions namely Testing set and Training set. Training set is the collection of old messages that were used for unsupervised learning, whereas Testing set consists of the recent messages for testing the model with machine learning approaches. The dataset was present in java script object notation (JSON) format. After dataset collection, it is further divided into variable size dataset as shown in Table III. to test its working with volatile size of data. The obtained result is highlighted in Section IV.C.

Table II. Confusion Matrix.

		ACTUAL	
		Spam	**Not Spam**
O/P	**Spam**	TP	FP
	Not Spam	FN	TN

Table III. Fragment Details.

Dataset	Content percentage (%)	Detail
1	25	Testing set
2	50	
3	75	
4	100	
5	25	Training set
6	50	
7	75	
8	100	

B. Performance Metrics

Confusion matrix illustrated in Table II has been used for evaluation purpose. True Positive (TP) represents the number of correctly classified spam among all spam reports and False Positive (FP) denotes the false spam reports among all reports. Similarly, True Negative (TN) denotes the correctly classified benign messages and False Negative denotes the false benign reports. Various performance metrics has been considered for the comparison of proposed model with existing approaches.

The ratio between correctly classified spam to that of total reporting is expressed as true positive rate and it can be mathematically represented as shown in Equation 4. Similarly, false positive rate can be expressed as the ratio between false spam reports to total non spam reports and it can be expressed mathematically as shown in Equation 5.The ratio among correctly classified messages to total classification messages is termed as Accuracy and it can be expressed mathematically using Equation 6.

$$\text{True Positive Rate, } TPR = \frac{TP}{TP + FN} \tag{4}$$

$$\text{False Positive Rate, } FPR = \frac{FP}{FP + TN} \tag{5}$$

$$\text{Accuracy, } ACC = \frac{TP + TN}{TP + FN + FP + TN} \tag{6}$$

C. Result and Comparison

Different performance parameters from confusion matrix has been calculated and tabulated in Table I. Using performance metrics described in Section IV.B, True positive rate (TPR), false positive rate (FPR), and accuracy (ACC) is calculated and a comparison between proposed approach and existing approaches has been prepared. Table IV represents the statistical comparison among competing algorithms. The observed value of Table I is used to calculate true positive rate for the competing algorithm, the comparison has been made with varying dataset and plotted in Fig.4.

Table I. Values of TP, FP, FN, and TN for competing algorithms.

Dataset	TP			FP			FN			TN		
	HITS	SALSA	R-SALSA	HITS	SALSA	R-SALSA	HITS	SALSA	R-SALSA	HITS	SALSA	R-SALSA
1	894	1004	1021	50	32	22	320	210	193	257	275	285
2	1355	1501	1516	66	42	28	430	284	269	456	480	494
3	1847	1991	2033	70	67	36	481	337	295	661	664	695
4	2179	2334	2420	73	76	58	563	408	322	913	910	928
5	609	625	646	82	77	62	103	84	63	368	376	391
6	964	1035	1061	104	95	81	172	99	73	627	638	652
7	1233	1337	1397	121	111	89	313	206	146	689	702	724
8	1439	1551	1629	160	135	109	366	254	176	852	877	903

Table IV. Statistical Performance Comparison.

Algorithm Used	Average TPR Percentage	Average FPR Percentage	Average ACC Percentage
HITS	79.71	13.54	81.75
SALSA	86.38	11.55	87.08
R-SALSA	88.40	8.72	89.25

The achieved improvement in the filtering rate can be noticed clearly because of the effectiveness of R-SALSA which combines the benefits of SALSA in scoring messages highlighted by Najork et. al.,[17] along with the reliability factor. Correspondingly Fig. 5 shows the false positive rate of competing methodologies with variable size datasets. Proposed algorithm is observed to be performing well compared to existing ones because of the introduction of reliability factor that improved the spam scoring as explained in Section 3.1. Similarly Fig. 6 illustrates the accuracy achieved using the proposed algorithm. By analyzing the plotted graph, we can observe R-SALSA based model outperforms the existing unsupervised mechanisms. Unlike HITS and SALSA, the proposed model combines concept of relative scoring and reliability factor which makes it efficient and perform better.

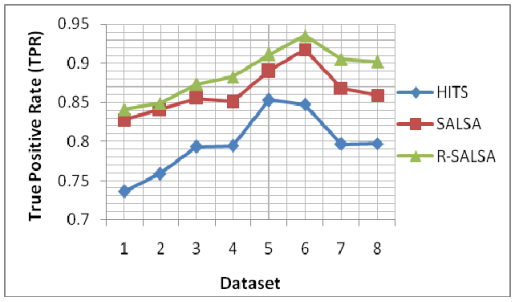

Fig. 4. TPR with variable size dataset

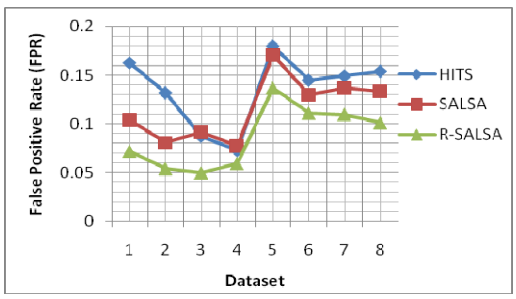

Fig. 5. FPR with variable size dataset

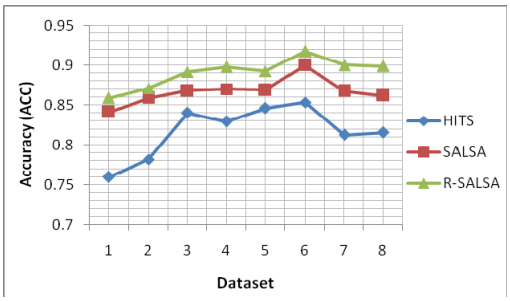

Fig. 6. ACC with variable size dataset

V. Conclusion and Future Work

The observed value of Table I is used to calculate true positive rate for the competing algorithm, the comparison has been made with varying dataset and plotted in Fig. 4. The achieved improvement in the detection rate can be noticed clearly because of the effectiveness of R-SALSA which combines the benefits of SALSA with reliability factor for scoring contents highlighted by Najork et. al.,[17] along with the reliability factor. Correspondingly Fig. 5 shows the false positive rate of competing algorithms with variable size datasets. Proposed algorithm is observed to be performing well compared to existing ones because of the introduction of reliability factor that improved the spam scoring as explained in Section III.A. Similarly Fig. 6 illustrates the accuracy achieved using the proposed algorithm. By analyzing the plotted graph, we can observe R-SALSA based model outperforms the existing unsupervised mechanisms. Unlike HITS and SALSA, the proposed model combines concept of relative scoring and reliability factor which makes it efficient and perform better.

References

[1] S. Wasserman and K. Faust, "*Social Network Analysis, Methods and Applications,*" Cambridge Univ. Press, 1994, pp. 505-555.

[2] S. Murugesan, "*Understanding Web 2.0,*" IT Prof., vol. 9, no. 4, 2007, pp. 34–41.

[3] R. Lempel and S. Moran, "*SALSA: the stochastic approach for link-structure analysis*," ACM Trans. Inf. Syst. TOIS, vol. 19, no. 2, pp. 131–160, 2001.

[4] P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang, and R. Zadeh: "*Wtf: The who to follow service at twitter,*" in Proceedings of the 22nd international conference on World Wide Web, pp. 505–514, 2013.

[5] M. Najork, S. Gollapudi, and R. Panigrahy, "*Less is more: sampling the neighborhood graph makes salsa better and faster,*" in Proceedings of the Second ACM International Conference on Web Search and Data Mining, pp. 242–251, 2009.

[6] T. Fawcett, "In vivo" spam filtering, "*A chalenge problem for data mining,*" KDD Explorations 5, vol. 2, pp. 203-231, 2003.

[7] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. Al Najada, "*Survey of review spam detection using machine learning techniques*," Journal of Big Data 2.1, pp. 1-24, 2015.

[8] P. Heymann, G. Koutrika, and H. Garcia-Molina, "*Fighting spam on social web sites: A survey of approaches and future challenges,*" Internet Comput. IEEE, vol. 11, no. 6, pp. 36–45, 2007.

[9] I. Idris, A. Selamat, N. T. Nguyen, S. Omatu, O. Krejcar, K. Kuca, and M. Penhaker, "*A combined negative selection algorithm-paticle swarm optimization for an email detection system,"* Engineering Application of Artificial Intelligence, 39, pp. 33-44, 2015.

[10] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "*Binary PSO with mutation operator for feature selection using decision tree applied to spam detection,*" Knowledge-Based Systems, 64, pp. 22-31, 2014.

[11] S. Khattak, N. R. Ramay, K. R. Khan,Affan A. Syed, and S.A. Khayam, "*A Taxonomy of Botnet Behavior, Detection, and Defence,*" IEEE Communications Surveys & Tutorials, Vol. 16, No. 2, 2014.

[12] Haiying Shen and Ze Li, "*Leveraging Social Networks for Effective Spam Filtering,*" IEEE Transactions on Computers, Vol. 63, No. 11, Nov. 2014.

[13] M. Bosma, E. Meij, and W. Weerkamp: "*A framework for unsupervised spam detection in social networking sites,*" in Advances in Information Retrieval, Springer, pp. 364–375, 2012.

[14] Sangho Lee and Jong Kim, "*WarningBird: A Near Real-Time Detection System for Suspecious URLs in Twitter Stream,*" IEEE Transactions on Dependable and Secure Computing, Vol. 10, No. 3, May/June 2013.

[15] M. Mccord and M. Chuah, "*Spam detection on twitter using traditional classifiers,*" Autonomic and trusted computing, Springer Berlin Heidelberg, pp. 175-186, 2011.

[16] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, C. Zhang, and K. Ross, "*Identifying video spammers in online social networks,*" Proceedings of the 4th international workshop on Adversarial information retreival on the web,ACM, pp. 45-52, 2008.

[17] M. Najork, S. Gollapudi, and R. Panigrahy, "*Less is more: sampling the neighborhood graph makes salsa better and faster,*" in Proceedings of the Second ACM International Conference on Web Search and Data Mining, pp. 242–251, 2009.

[18] M. Agrawal and R. L. Velusamy, "*Unsupervised Spam Detection in Hyves Using SALSA,*" 4th International Conference on Frontier in Intelligent Computing: Theory and Applications, 2015.