Available at: www.researchpublications.org

A Robust Spam Detection System using a collaborative approach with an E-Mail Abstraction Scheme and Spam Tree Data Structure

Miss.K.S.Sathawane
M.E II Sem.at P.R.M.I.T.R
(PT-CSE)
khushboo.sathawane@gmail.com

Prof.Miss.R.R.Tuteja
Asst. Prof. at P.R.M.I.T.R
(M.E. CSE)
ranu.tuteja@gmail.com

Abstract—E-mail communication has become a necessary part of our day to day life, however the e-mail spam problem is on rise hugely. Unsolicited email is not only a nuisance but can be potentially dangerous. In recent years, so many techniques are developed to detect the spam emails and the idea of collaborative spam filtering with near-duplicate similarity matching scheme has been commonly talked about. This scheme for spam detection maintains a known spam database, formed by user feedback, and then blocks succeeding near-duplicate spams. The prior works is mainly based upon a brief abstraction derived from e-mail content text. However, these abstractions of e-mails cannot fully catch the growing nature of spams, and are thus not successful enough in near-duplicate detection. In this paper, a novel e-mail abstraction scheme is proposed, which considers e-mail layout structure to represent e-mails. Moreover, a Robust and Collaborative Spam Detection System is presented, which possesses an efficient near-duplicate matching scheme and a progressive update scheme.

1. Introduction

E-mail communication is common and necessary nowadays, but the e-mail spam problem continues growing drastically. Unsolicited email is not only a nuisance but can be potentially dangerous. According to a survey, 40 percent of e-mails were considered as spams in 2006. The spam detection problem is growing because the spammers will always find new ways to attack spam filters due to the economic benefits of sending spams.

The primary idea of the similarity matching scheme for spam detection is to maintain a known spam database, formed by user feedback, to block subsequent near-duplicate spams. The reason behind that is to achieve efficient similarity matching and reduced storage utilization. For that purpose prior works mainly represent each e-mail by a brief abstraction derived from e-mail content text. However, these abstractions of e-mails cannot fully catch the growing nature of spams, and are thus not successful enough in near-duplicate detection. Note that existing filters generally perform well when dealing with clumsy spams, which have duplicate content with suspicious keywords or are sent from an identical disreputable server. Therefore, the next stage of spam detection research should focus on dealing with cunning spams which evolve naturally and continuously. In this paper, a novel e-mail abstraction scheme is proposed which considers e-mail layout structure to represent e-mails. A procedure to generate the e-mail abstraction using HTML content in e-mail is presented, which can more effectively capture the near-duplicate phenomenon of spams. Moreover, a complete spam detection system is designed, which possesses an efficient near-duplicate matching scheme and a progressive update scheme. The progressive update scheme enables system to keep the most up-to-date information for near-duplicate detection.

2. Related Works

Various techniques have been discovered to solve this e-mail spam problem. Previous works on spam detection can be generally classified into three categories: 1) content-based methods, 2) non content-based methods, and 3) others. Initially, researchers used to analyze e-mail content text, representatives of this category are Naive Bayes [14], Bayesian [16] and Support Vector Machines (SVMs) [6], [19] methods. Certain specific features, such as URLs [21] and images [22], [23] have also been taken into account for spam detection. While conventional machine learning techniques[17],[18],[20] have reported excellent results with static data sets, one major disadvantage is that it is cost-prohibitive for large-scale applications to constantly retrain these methods with the latest information to adapt to the rapid evolving nature of spams. The spam detection of these methods on the e-mail corpus with various language has been less studied yet.

The other group attempts to exploit noncontent information such as e-mail header, e-mail social network

Available at: www.researchpublications.org

[11], and e-mail traffic [7] to filter spams. Collecting notorious and innocent sender addresses (or IP addresses) from e-mail header to create black list and white list is a commonly applied method initially. Since e-mail header can be altered by spammers to conceal the identity, the main drawback of these methods is the hardness of correctly identifying each user. In addition, how to efficiently update the whole included classifiers is another unsolved issue.

3. PRELIMINARIES

3.1 What is SPAM?

Definition[15]: *S*pam is a term used to describe Unsolicited Commercial Email (UCE) or Unsolicited Bulk Email (UBE). In general, the predominant subjects of spam email are the following:1) Chain letters.2) Pyramid schemes (including Multilevel Marketing, or MLM).3) Other "Get Rich Quick" or "Make Money Fast" (MMF) schemes.4) Offers of bulk e-mailing services for sending UCE. 5) Ilegally pirated software etc.

3.2 Definition of Near-Duplicate

The fundamental idea of near-duplicate spam detection is to utilize reported known spams to block subsequent ones which have similar content. This paper represents each e-mail using an HTML tag sequence, which depicts the layout structure of e-mail, and look forward to more effectively capturing the near-duplicate phenomenon of spams.

3.3 Definition of (<anchor>) tag.

The tag <anchor> is one type of newly defined tag that records the domain name or the e-mail address in an anchor tag.For example, the anchor tag <a href="http://arbor.ee. ntu.edu.tw/index.htm"> is transformed to *<arbor.ee.ntu. edu.tw>*. The purpose of creating the *<anchor>* tag is to minimize the false positive rate when the number of tags in an e-mail abstraction is short. The less the number of tags in an e-mail abstraction, the more possible that a ham may be matched with known spams and be misclassified as a spam.

3.4 Definition of (<my text=>) tag

<mytext=> is a newly defined tag that represents a paragraph of text without any HTML tag embedded. Since we ignore the semantics of the text, the proposed abstraction scheme is inherently applicable to e-mails in all languages. This significant feature is superior to most existing methods.

3.5 Definition of (Tag Length).

The tag length of an e-mail abstraction is defined as the number of tags in an e-mail abstraction. Note that we strictly define that two e-mail abstractions are near-duplicate only if they are exactly identical to each other.

4. STRUCTURE ABSTRACTION GENERATION (SAG)

The specific procedure SAG is proposed to generate the e-mail abstraction using HTML content in e-mail. The algorithmic form of SAG is outlined in Fig. 4.1. Procedure SAG is composed of three major phases, Tag Extraction Phase, Tag Reordering Phase, and <anchor> Appending Phase. In Tag Extraction Phase, the name of each HTML tag is extracted, and tag attributes and attribute values are eliminated. In addition, each paragraph of text without any tag embedded is transformed to <mytext=>. In lines 4-5, <anchor> tags are then inserted into AnchorSet.Sub- sequently,in line 6 of Fig.4.1,

```
Procedure SAG
Input:  the email with text/html content-type,
        the tag length threshold (Lth_short) of the short email
Output: the email abstraction (EA) of the input email
1   // Tag Extraction Phase
2   Transform each tag to <tag.name>;
3   Transform each paragraph of text to <mytext/>;
4   AnchorSet = the union of all <anchor>;
5   EA = the concatenation of <tag.name>;
6   Preprocess the tag sequence of EA;
7   // Tag Reordering Phase
8   for (each tag of EA) // pn: position number
9       tag.new_pn = ASSIGN_PN (EA.tag_length, tag.pn);
10      Put the tag to the position tag.new_pn;
11  EA = the concatenation of <tag.name> with new_pn;
12  // <anchor> Appending Phase
13  if (EA.tag_length < Lth_short)
14      Append AnchorSet in front of EA;
15  return EA;
End
```

Fig.4.1. Algorithmic form of procedure SAG.

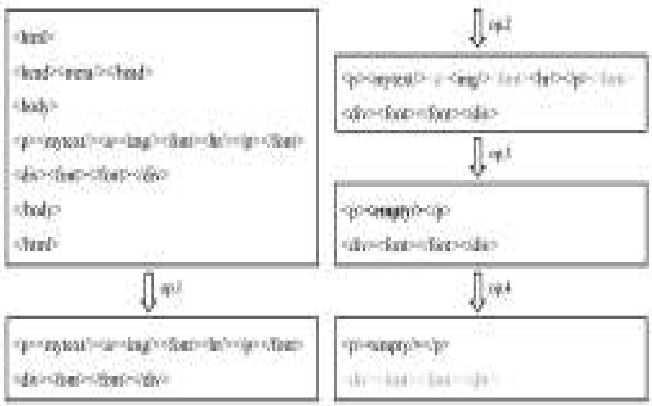
we preprocess the tag sequence of the tentative e-mail abstraction. The following sequence of operations is performed in the preprocessing step Fig4.2.

- Front and rear tags (as shown in the gray area of the example e-mail in the top of Fig. 4.3) are excluded.
- Nonempty tags that have no corresponding start tags or end tags are deleted. Besides, mismatched nonempty tags are also deleted.
- All empty tags are regarded as the same and are replaced by the newly created <empty=> tag. Moreover, successive <empty=> tags are pruned and only one <empty=> tag is retained.
- The pairs of nonempty tags enclosing nothing are removed.

On purpose of accelerating the near-duplicate matching process, we reorder the tag sequence of an e-mail abstraction in Tag Reordering Phase. In the worst case, if we consider two e-mail abstractions which have the same tag length and differ only in their last tags, the difference cannot be detected until the last tags are compared. In lines 8-11 of Fig. 4.1, each tag is assigned a new position number by function ASSIGN_PN *(PN* denotes for *Position Number.)* Fig. 4.3 demonstrates the assignment of the first six tags. An example e-mail abstraction produced by procedure SAG is shown in the bottom of Fig.3

5. DESIGN OF SPTABLE AND SPTREES

Available at: www.researchpublications.org



SpTable and SpTrees (sp stands for spam) are proposed to store large amounts of the e-mail abstractions of reported spams. As shown in Fig.5.1 , several SpTrees are the kernel of the database, and the e-mail abstractions of collected spams are maintained in the corresponding SpTrees.

Two e-mail abstractions are possible to be near-duplicate only when the numbers of their tags are identical. Thus, if we distribute e-mail abstractions with different tag lengths into diverse SpTrees, the quantity of spams required to be matched will decrease.

Fig. 4.2. An example of the preprocessing step in Tag Extraction Phase of procedure SAG.

For efficient matching Sp Trees are designed to be binary trees. The branch direction of each SpTree is determined by a binary hash function. If the first tag of a subsequence is a start tag (e.g.,<div>), this[4] subsequence will be placed into the left child node. A subsequence whose first tag is an end tag (e.g.,</div>) will be placed into the right child node. Since most HTML tags are in pairs and the proposed e-mail abstraction is reordered in SAG, subsequences are expected to be uniformly distributed. Moreover,on level i of each SpTree (with the root on level 0), each node stores subsequences whose tag lengths are equal to 2i. For instance, as shown in Fig.5.2, the subsequence <spam:com> is placed into level 0, the subsequence </p><a> (whose tag length is 21) is placed into level 1, and so forth[4].

6. Robustness Issue

The main difficulty of near-duplicate spam detection is to withstand malicious attack by spammers. Prior approaches generate e-mail abstractions based mainly on hash-based content text.

For example, the authors in[8] extract words or terms to generate the e-mail abstraction. Besides, substrings extracted by various techniques are widely employed in [9],[5],[17],[18][20]. However, this type of e-mail representation inherently has following disadvantages. First, the insertion of a randomized and normal paragraph can easily defeat this type of spam filters. Moreover, since the structures and features of different languages are diverse, word and substring extraction may not be applicable to e-mails in all languages. To assess the robustness of the proposed scheme, we model possible spammer attacks and organize these attacks as following three categories.

6.1 Random Paragraph Insertion

This type of spammer attack is commonly used nowadays. As shown in Fig. 6.1, normal contents without any advertisement keywords are inserted to confuse text based spam filtering techniques. It is noted that our scheme transforms each paragraph into a newly created tag <mytext=>, and consecutive empty tags will then be transformed to <empty=>. As such, the representation of each random inserted paragraph is identical, and thus our scheme is resistant to this type of attack.

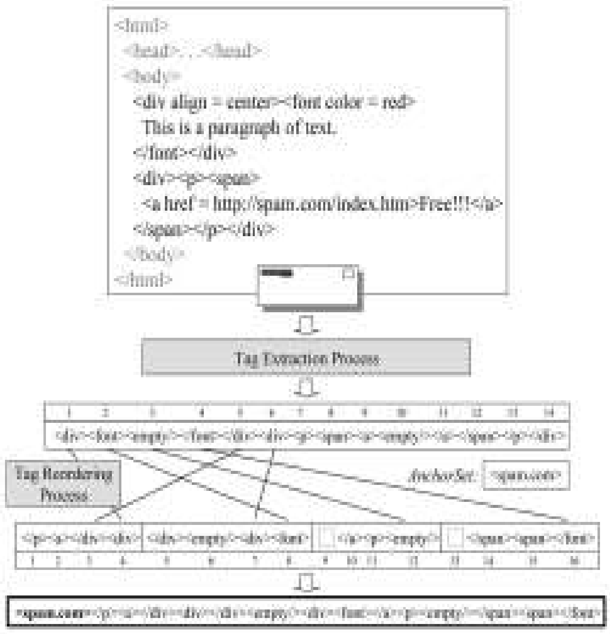


Fig. 4.3. An example procedure flow of SAG.

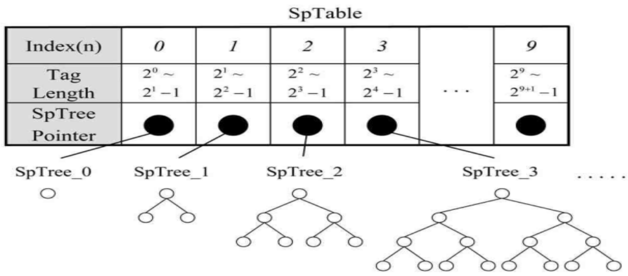


Fig. 4. The data structures of SpTable and SpTrees.

Available at: www.researchpublications.org

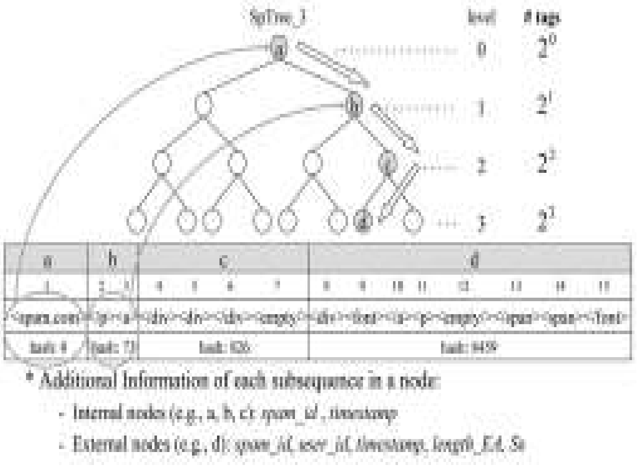


Fig.5.2 The SP-Tree Data Structure

6.2 Random HTML Tag Insertion

If spammers know that the proposed scheme is based on HTML tag sequences, random HTML tags will be inserted rather than random paragraphs.On the one hand, arbitrary tag insertion will cause syntax errors due to tag mismatch-ing. This may lead to abnormal display of spam content that spammers do not wish this to happen. On the other hand, procedure SAG also adopts some heuristics to deal with the random insertion of empty tags and the tag mismatching of nonempty tags. Fig. 6.1 shows two example outputs.

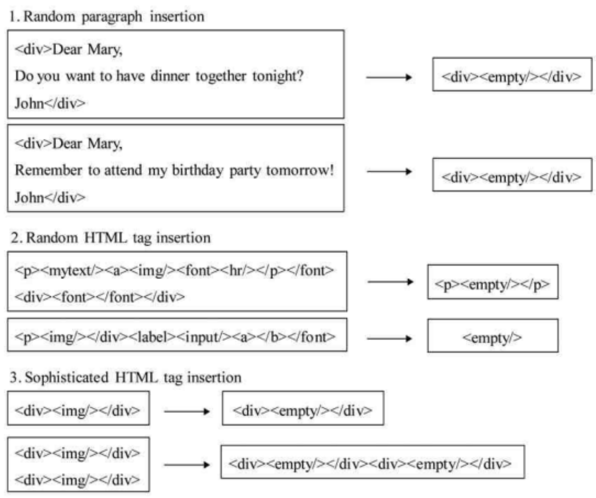


Fig. 6.1. Examples of possible spammer attacks.

6.3 Sophisticated HTML Tag Insertion

Suppose that spammers are more sophisticated, they may insert legal HTML tag patterns. As shown in Fig. 6.1 if tag

patterns that do conform to syntax rules are inserted,they will not be eliminated. However, it is not intuitive for spammers to generate a large number of spams with completely distinct e-mail layout structure.

Hence representing emails with layout structure is more robust to most existing attacks than text-based approaches. Even though new attack has been designed, we can react against it by adjusting the preprocessing step of procedure SAG. The proposed abstraction scheme can be applied to e-mails in all languages without modifying any components. This feature also enables system Cosdes to perform more robustly.

7. COLLABORATIVE SPAM DETECTION SYSTEM- COSDES

COSDES is a complete spam detection system. Collaborative Spam Detection System which possesses an efficient near-duplicate matching scheme and a progressive update scheme. The progressive update scheme not only adds in new reported spams, but also removes obsolete ones in the database. In addition, to withstand intentional attacks, a reputation mechanism is also provided in Cosdes to ensure the truthfulness of user feedback.

7.1 The System Model- Cosdes

The system model of Cosdes is illustrated in Fig. 7.1, and the algorithmic form is outlined in Fig. 7.2. Before starting to do the spam detection, Cosdes collects feedback spams for time T_m in advance to construct an initial database. Three major modules, Abstraction Generation Module, Database Maintenance Module, and Spam Detection Module, are included in Cosdes. With regard to Abstraction Generation Module, each e-mail is converted to an e-mail abstraction by Structure Abstraction Generator with procedure SAG. Three types of action handlers, Deletion Handler, Insertion Handler, and Error Report Handler, are involved in Database Maintenance Module. In addition, Matching Handler in Spam Detection Module takes charge of determining results.

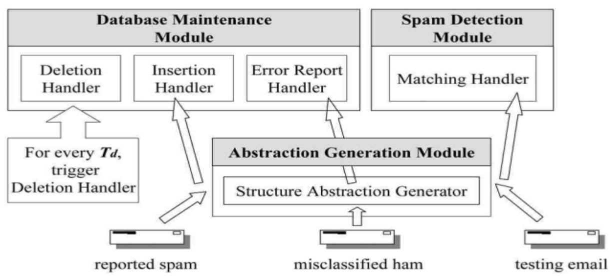


Fig. 7.1. System model of Cosdes.

There are three types of e-mails, reported spam, testing e-mail, and misclassified ham, required to be dealt with by Cosdes. When receiving a reported spam, Insertion Handler adds the e-mail abstraction of this spam into the database except that the reputation score of this reporter istoo low. Whenever a new testing e-mail arrives, Matching Handler performs the near-duplicate detection with collected spams to do the judgment. Meanwhile, if a testing email is classified as a spam, this e-mail will be viewed as a reported spam and be added into the database. Moreover, Error Report Handler copes with feedback misclassified hams and adjusts Cosdes by degrading the reputation of related reporters to prevent malicious attacks. For every T_d, Deletion Handler is triggered to delete obsolete spams which exist over time T_m. Overall, Cosdes is self-adjusting and retains the most up-to-date spams for near-duplicate detection.[12]

Available at: www.researchpublications.org

7.2 Reputation Mechanism

The principal concept of collaborative spam detection is to collect human judgment to block subsequent near-duplicate spams. To ensure the truthfulness of spam reports and to prevent malicious attacks, we propose the reputation mechanism to evaluate the credit of each reporter. The fundamental idea of the reputation mechanism is to utilize a reputation table to maintain a reputation score S_R of each reporter according to the previous reliability record. In such a context, when doing near-duplicate detection, if the sum of suspicion scores of matched spams exceeds a predefined threshold, the testing e-mail will be classified as a spam. The reputation mechanism is described in detail as follows:

1. Each reporter is assigned an initial score $S_{initial}$ when he submits a reported spam at the first time.

2. If a reporter submits any feedback spam once more, the reputation score will be incremented by a smaller incremental score S_{incre}.

3. If a reporter is charged that his previous feedback spam is mistaken, the reputation score will be halved.

8. Features Of Cosdes

Research in considering e-mail layout structure to represent e-mails in the field of near-duplicate spam detection is a unique way of spam detection. In summary, the properties of Cosdes are as follows:

1.The specific procedure SAG is proposed to generate the e-mail abstraction using HTML content in e-mail, and this newly devised abstraction can more effectively capture the near-duplicate phenomenon of spams.

2.An innovative tree structure is devised, SpTrees, to store large amounts of the e-mail abstractions of reported spams. SpTrees contribute to the accomplishment of the efficient near-duplicate matching with a more sophisticated e-mail abstraction.

```
System Cosdes
Input: Tm: the maximum time span for reported spams being retained in
           the system,
       Td: the time span for triggering Deletion Handler,
       Sth: the score threshold for determining spams
switch (circumstance)
case: when receiving a reported spam
   if (EA.reporter.SR > Sinitial);
      Trigger Insertion Handler(EA);
      Increase SR of the reporter in RepTable; // Rep: Reputation
   break;
case: when receiving a testing email
   Trigger Matching Handler(EA, Sth);
   if (the testing email is classified as a spam);
      Trigger Insertion Handler(EA);
   break;
case: when receiving a misclassified ham
   Trigger Error Report Handler(EA);
   break;
case: for every Td
   Trigger Deletion Handler(Tm);
   break;
End
```

Fig . 7.2. Algorithmic form of Collabrative Spam detection system

3.A complete spam detection system Cosdes is designed with an efficient near-duplicate matching scheme and a progressive update scheme. The progressive update scheme enables system Cosdes to keep the most up-to-date information for near-duplicate detection.

4.The reputation mechanism is proposed to evaluate the credit of each reporter.

5.Since we are comparing only e-mail layout there is a reduction in time and cost factor of comparing the whole text content.

6.Representing emails with layout structure is more robust to most existing attacks than text-based approaches.

9. Challenges To Detect Spam E-Mails

Spammers are finding ways to trick people into thinking their unsolicited junk messages are worth the time you spend reading them. A list of the top five ways to tell if an email is spam is as follows[4]. These rules can help you when spam slips through the protection of your Spam filter.

- *If it ends up in Spam Folder*
- *Look at the Email Address*
- *Look at the Content*
- *If it asks for personnel Information*
- *Look at the Greeting*

10. Conclusion

A superior e-mail abstraction scheme is required to more certainly catch the evolving nature of spams in the field of collaborative spam filtering by near-duplicate detection. Compared to the existing methods, in this paper, a more sophisticated and robust e-mail abstraction scheme is explored, which considers e-mail layout structure to represent e-mails. The specific procedure SAG is proposed to generate the e-mail abstraction using HTML content in e-mail, and this newly-devised abstraction can more effectively capture the near-duplicate phenomenon of spams. Moreover, a complete spam detection system Cosdes has been designed to efficiently process the near-duplicate matching and to progressively update the known spam database.

11. References

[1] M.-T. Chang, W.-T. Yih, and C. Meek, “Partitioned Logistic Regression for Spam Filtering,” Proc. 14th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data mining (KDD), pp. 97-105, 2008.
[2] A.C. Cosoi, “A False Positive Safe Neural Network; The Followers of the Anatrim Waves,” Proc. MIT Spam Conf., 2008.
[3] B. Mehta, S. Nangia, M. Gupta, and W. Nejdl, “Detecting Image Spam Using Visual Features and Near Duplicate Detection,” Proc. 17th Int’l Conf. World Wide Web (WWW), pp. 497-506, 2008.
[4] M. Siva kumar reddy & Krishna Sagar. “Improved Near duplicate matching scheme for e-mail spam Detection” International Journal of Internet Computing ISSN No: 2231 – 6965, VOL- 1, ISS- 4 2012 29

[5] S. Sarafijanovic, S. Perez, and J.-Y.L. Boudec, “Resolving FP-TP Conflict in Digest-Based Collaborative Spam Detection by Use of Negative Selection Algorithm,” Proc. Fifth Conf. Email and Anti-Spam (CEAS), 2008.

[6]E. Blanzieri and A. Bryl, “Evaluation of the Highest Probability SVM Nearest Neighbor Classifier with Variable Relative Error Cost,” Proc.

Fourth Conf. Email and Anti-Spam (CEAS), 2007.
[7] R. Clayton, "Email Traffic: A Quantitative Snapshot," Proc. of the Fourth Conf. Email and Anti-Spam (CEAS), 2007.

[8] M.S. Pera and Y.-K. Ng, "Using Word Similarity to Eradicate Junk Emails," Proc. 16th ACM Int'l Conf. Information and Knowledge Management (CIKM), pp. 943-946, 2007.

[9] S. Sarafijanovic and J.-Y.L. Boudec, "Artificial Immune System for Collaborative Spam Filtering," Proc. Second Workshop Nature Inspired Cooperative Strategies for Optimization (NICSO), 2007.

[10] Revathy.K, Revathi.K.K, A.Boopalan "Cosdes:A Collabraive Spam Detection System Based on Bayesian Approach" OSIET journal of Computer Science Engineering, May 2012.

[11] C.-Y. Tseng, J.-W. Huang, and M.-S. Chen, "Promail: Using Progressive Email Social Network for Spam Detection," Proc. 10th Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD), pp. 833-840, 2007.

[12] G.Surekha1, Y.Sowjanya Kumari , Dr.P.Harini "Cosdes:A Collabraive Spam Detection System with novel e-mail abstraction" IOSR Journal of Engineering (IOSRJEN) e-ISSN: 2250-3021, p-ISSN: 2278-8719, www.iosrjen.org Volume 2, Issue 9 (September 2012)

[13] T.R. Lynam and G.V. Cormack, "On-Line Spam Filter Fusion," Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), pp. 123-130, 2006.

[14] J. Hovold, "Naive Bayes Spam Filtering Using Word-Position-Based Attributes," Proc. Second Conf. Email and Anti-Spam (CEAS)

[15] Steve Davis & Gloria Craney- e-book " How Do I Stop Spam?

[16] Joseph S. Kong, Behnam A. Rezaei, Nima Sarshar, and Vwani P. Roychowdhury , P. Oscar Boykin "**Collaborative Spam Filtering Using E-Mail Networks"** IEEE Compu t e r So c i e t y **0018-9162/06/$20.00 © 2006 IEEE** (August 2006)

[17] E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati, "P2P-Based Collaborative Spam Detection and Filtering," Proc. Fourth IEEE Int'l Conf. Peer-to-Peer Computing, pp. 176-183, 2004.

[18] A. Gray and M. Haahr, "Personalised, Collaborative Spam Filtering," Proc. First Conf. Email and Anti-Spam (CEAS), 2004.

[19] A. Kolcz and J. Alspector, "SVM-Based Filtering of Email Spam with Content-Specific Misclassification Costs," Proc. ICDM Workshop Text Mining, 2001.

[20] A. Kolcz, A. Chowdhury, and J. Alspector, "The Impact of Feature Selection on Signature-Driven Spam Detection," Proc. First Conf. Email and Anti-Spam (CEAS), 2004.

[21] K.M. Schneider, "Brightmail URL Filtering," Proc. MIT Spam Conf.,2004.

[22] B. Mehta, S. Nangia, M. Gupta, and W. Nejdl, "Detecting Image Spam Using Visual Features and Near Duplicate Detection," Proc. 17th Int'l Conf. World Wide Web (WWW), pp. 497-506, 2008.

[23] Z. Wang, W. Josephson, Q. Lv, and K.L.M. Charikar, "Filtering Image Spam with Near-Duplicate Detection," Proc. Fourth Conf. Email and Anti-Spam (CEAS), 2007.