ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 4, June 2012

Accurate Spam Mail Detection using Bayesian Algorithm

V.Kanaka Durga, M.R. Raja Ramesh

Abstract— **With the increasing popularity of a E-mail users, E-mail spam problem growing proportionally. Spam filtering with near duplicate matching scheme is widely discussed in recent years. It is based on a known spam database formed by user feedback which cannot fully catch the evolving nature of spam and also it requires much storage. In view of above drawbacks, we proposed an effective spam detection scheme based on Bayesian approach. First we use Bayesian filter based on single keyword sets and then we improve the scheme by using multiple keyword sets and satisfactory results obtained.**

Index Terms— **Spam detection, Near Duplicate Matching, SAG.**

I. INTRODUCTION

With the increasing popularity of electronic mail (or e-mail), several people and companies found it an easy way to distribute a massive amount of unsolicited messages to a tremendous number of users at a very low cost. These unwanted bulk messages or junk emails are called spam messages. The majority of spam messages that has been reported recently are unsolicited commercials promoting services and products including sexual enhancers, cheap drugs and herbal supplements, health insurance, travel tickets, hotel reservations, and software products. They can also include offensive content such pornographic images and can be used as well for spreading rumors and other fraudulent advertisements such as make money fast. E-mail spam has become an epidemic problem that can negatively affect the usability of electronic mail as a communication means. Besides wasting users' time and effort to scan and delete the massive amount of junk e-mails received; it consumes network bandwidth and storage space, slows down e-mail servers, and provides a medium to distribute harmful and/or offensive content. Spam emails are getting better in its ability to break anti-spam filters and it would take a great deal to get it fully eradicated. Spammers are also becoming more innovative, so that the anti-spam research is having a great relevance these days.

A. How do spammers get your address?

If you or anyone else ever gives your E-mail address on website, then your E-mail address is out there for the world to use because the companies often sell your address. For instance If you go to a website and send a funny cartoon, or an electronic greeting to a friend you have given out their E-mail address and the rest is history. Nothing is free, and these websites make their money by selling advertising and selling your E-mail address. If your address is posted on a public forum or visible on a website, spammers will find it and start to bombard you. Most spam messages on the internet today are advertisements.

B. Spam Problems

Spammers often says that spam is not a problem." Just hit delete if you don't want to see it". And many spam messages carry the tagline. "If you don't want to receive further mailing reply, we will remove you." But spam is huge problem. It consumes network bandwidth, storage space and slowdown E-mail servers.

C. Flavors of spam:

Unsolicited Commercial E-mail (UCE)-An E-mail messages that you receive without asking for it advertising a product or service. This is also called junk E-mail. Unsolicited Bulk E-mail(UBE)-UBE refers to E-mail messages that are sent in bulk to thousands of recipients. BE may be commercial in nature, in which case it is also UCE. But it may be sent for other purposes as well, such as political lobbying or harassment. Make Money Fast(MMF)-MMF messages in the form of chain letters or multi-level marketing schemes, are messages that suggest you can get rich by sending money to the top name on list, removing that name, adding your name to the bottom of the list ,and forwarding the messages to the other people.

E-mail communication is prevalent and indispensable nowadays. However, the threat of unsolicited junk e-mails, also known as spams, becomes more and more serious. According to a survey by the website Top Ten REVIEWS, 40 percent of e-mails were considered as spams in 2006. The statistics collected by MessageLabs show that recently the spam rate is over 70 percent and persistently remains high. The primary challenge of spam detection problem lies in the fact that spammers will always find new ways to attack spam filters owing to the economic benefits of sending spams. 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 notorious server. Therefore, the next stage of spam detection research should focus on coping with cunning spams which evolve naturally and continuously.

In view of above facts, the notion of collaborative spam filtering with near-duplicate similarity matching scheme has recently received much attention. The primary idea of the near-duplicate matching scheme for spam detection is to maintain a known spam database, formed by user feedback, to block subsequent spams with similar

content. Collaborative filtering indicates that user knowledge of what spam may subsequently appear is collected to detect following spams. Overall, there are three key points of this type of spam detection approach we have to be concerned about. First, an effective representation of e-mail (i.e., e-mail abstraction) is essential. Since a large set of reported spams have to be stored in the known spam database, the storage size of e-mail abstraction should be small. Moreover, the e-mail abstraction should capture the near-duplicate phenomenon of spams, and should avoid accidental deletion of non-spam e-mails (also known as hams). Second, every incoming e-mail has to be matched with the large database, meaning that the near-duplicate matching process should be substantially efficient. Finally, the latest spams to be included instantly and successively into the database so as to effectively block subsequent near-duplicate spams.

One subtle difference is that a false positive would be a more serious error than a false negative as a false positive would mean that an important e-mail was identified as spam and rejected. According to a leading body in IT, inaccurate anti-spam solutions may be responsible for wasting more than five million working hours a year on checking that legitimate messages were not mistakenly quarantined. We explore a new approach based on Bayesian approach that can automatically classify e-mail messages as spam or legitimate. We study its performance for various conjunction and disjunction operators for several datasets. The results are promising as compared with a previous classifiers such as genetic algorithm and XCS (extended Classifier System) classifier system. Classification accuracy above 97% and low false positive rates are achieved in many test cases.

II. Related Works

Nowadays E-mail spam problem is increasing drastically. So to relieve this problem various techniques have been explored. Previous works on spam detection can be classified into three categories: a) content-based methods, b) non content-based methods, and c) others . Initially, researchers analyze e-mail content text and model this problem as a binary text classification task. Representatives of this category are Naive Bayes [1], [2] and Support Vector Machines (SVMs) [5], [10], [9], [14] methods. In general, Naive Bayes methods train a probability model using classified e-mails, and each word in e-mails will be given a probability of being a suspicious spam keyword. The other group attempts to exploit non content information such as e-mail header, e-mail social network ,and e -mail traffic [13],[9] 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. On the other hand, collaborative spam filtering with near-duplicate similarity matching scheme has been studied widely in recent years. Regarding collaborative mechanism, P2P-based architecture [12], centralized server-based system [3], [16], [15], and others [6], [8] are generally employed. Note that no matter which mechanism is applied, the most critical factor is how to represent each e-mail for near-duplicate matching. The e-mail abstraction not only should capture the near-duplicate phenomenon of spams, but should avoid accidental deletion of hams.

Preprocessing Steps:

A. E-mail Abstraction Scheme

Here we introduced a novel e-mail abstraction scheme, SAG procedure is presented to depict the generation process of an e-mail abstraction in II-A-1, and the robustness issue is discussed in Section II-A-2.

1) Structure Abstraction Generation: We propose the SAG procedure to generate the e-mail abstraction using HTML content in e-mail. The algorithmic form of SAG is depicted in Fig. 1. Using three major phases Procedure SAG is composed, 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.. One objective of this preprocessing step is to remove tags that are common but not discriminative between e-mails. The other objective is to prevent malicious tag insertion attack, and thus the robust-ness of the proposed abstraction scheme can be further enhanced. The following sequence of operations is performed in the preprocessing step.

1. Front and rear tags are excluded.
2. Nonempty tags 2 that have no corresponding start tags or end tags are deleted. Besides, mismatched nonempty tags are also deleted.
3. All empty tags 2 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.
4. The pairs of nonempty tags enclosing nothing are removed.

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* = concatenation of <tag.name>*;*
6 Preprocess the tag sequence of *EA*;
7 // Tag Reordering Phase:
8 **for**(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 font *EA*;
15 **return** *EA*;
End

Fig. 1 Algorithm form of procedure SAG.

2) Robustness Issue: The near-duplicate spam detection's main difficulty is to withstand malicious attack by spammers. E-mail abstractions are generated based mainly on hash-based content text in prior approaches. For example, the authors in [3], [16], [15] extract words or terms to generate the e-mail abstraction. Besides, substrings extracted by various techniques are widely employed in [7], [12], [11], [6], [8], [4]. This type of e-mail representation has following disadvantages. Firstly, 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. Concretely speaking, for instance, trigrams of substrings used in [7], [8], [6] are not suitable for non alphabetic languages, such as Chinese. In this paper, we devise a novel e-mail abstraction scheme that considers e-mail layout structure to represent e-mails. To assess the robustness of the proposed scheme, we model possible spammer attacks and organize these attacks as following three categories. Examples and the outputs of preprocessing of procedure SAG are shown in Fig. 1.

A) Random Paragraph Insertion

This type of spammer attack is commonly used now-a-days. As shown in Fig. 2, 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.

B) 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 mismatching. 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 (as depicted in Section III-C to deal with the random insertion of empty tags and the tag mismatching of nonempty tags. Fig. 2 shows two example outputs and the details of each step can be found in Fig. 2. With the proposed method, most random inserted tags will be removed, and thus the effectiveness of the attack of random tag insertion is limited.

C) Sophisticated HTML Tag Insertion:

Suppose that spammers are more sophisticated, they may insert legal HTML tag patterns. As shown in Fig. 1, if tag patterns that do conform to syntax rules are inserted, they will not be eliminated.

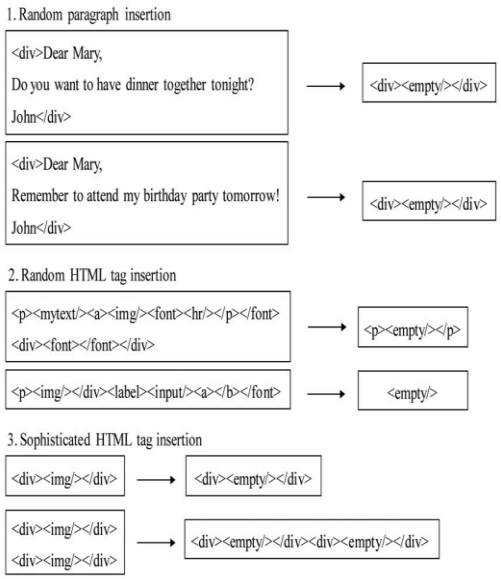

Fig. 2 Examples of possible Spammer Attacks

Note that due to space limitation, we are not able to discuss all possible situations. Nevertheless, representing e-mails 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. On the other hand, our approach extracts only HTML tag sequences and transforms each paragraph with no tag embedded to<mytext/>, meaning that the proposed abstraction scheme can be applied to e-mails in all languages without modify in any components.

III. BAYESIAN ALGORITHM

One of the most effective and intelligent solutions to combat spam email nowadays is Bayesian filtering.

How does Bayesian filtering work?

Naive Bayes classifier is used to identify spam e-mail. It is based on the principle that most events are dependent and that the probability of an event occurring in the future can be inferred from the previous occurrences of that event. The same technique can be used to classify spam. If some piece of text occurs often in spam but not in legitimate mail, then it would be reasonable to assume that this email is probably spam.

Bayesian filters must be 'trained' to work effectively. Particular words have certain probabilities (also known as likelihood functions) of occurring in spam email but not in legitimate email. For instance, most email users will frequently encounter the word Viagra in spam email, but will seldom see it in other email. Before mail can be filtered using this method, the user needs to generate a database with words and tokens (such as the $ sign, IP addresses and domains, and so on), collected from a sample of spam mail and valid mail (referred to as 'ham'). For all words in each training email, the filter will adjust the probabilities that each word will appear in spam or legitimate email in its database.

A. Mathematical Foundation

Bayes theorem is used several times in the context of spam:

1) First time, to compute the probability that the message is spam, knowing that a given word appears in this message.

2) Second time, to compute the probability that the message is spam, taking into consideration all of its words (or a relevant subset of them);

3) Sometimes third time, to deal with rare words.

B. Computing the probability that a message containing a given word is spam

Let's suppose the suspected message contains the word "replica". Most people who are used to receiving e-mail know that this message is likely to be spam, more precisely a proposal to sell counterfeit copies of well-known brands of watches. The spam detection software, however, does not "know" such facts, all it can do is compute probabilities.

The formula used by the software to determine that is derived from Bayes theorem

where:

- Pr(S|W) is the probability that a message is a spam, knowing that the word "replica" is in it;
- Pr(S) is the overall probability that any given message is spam;
- Pr(S|W) is the probability that the word "replica" appears in spam messages;
- Pr(H) is the overall probability that any given message is not spam (is "ham");
- Pr(W|H) is the probability that the word "replica" appears in ham messages.

C. The Spamicity of a word

Recent statistics show that the current probability of any message being spam is 80%, at the very least:

Pr(S)=0.8;Pr(H)=0.2

However, most bayesian spam detection software makes the assumption that there is no *a priori* reason for any incoming message to be spam rather than ham, and considers both cases to have equal probabilities of 50%:

Pr(S)=0.8;Pr(H)=0.2

The filters that use this hypothesis are said to be "not biased", meaning that they have no prejudice regarding the incoming email. This assumption permits simplifying the general formula to:

$$\Pr(S|W) = \frac{\Pr(W|S)}{\Pr(W|S) + \Pr(W|S)}$$

This quantity is called "spamicity" (or "spaminess") of the word "replica", and can be computed. The number Pr(W|S) used in this formula is approximated to the frequency of messages containing "replica" in the messages identified as spam during the learning phase. Similarly, Pr(W|H) is approximated to the frequency of messages containing "replica" in the messages identified as ham during the learning phase. For these approximations to make sense, the set of learned messages needs to be big and representative enough.[9] It is also advisable that the learned set of messages conforms to the 50% hypothesis about repartition between spam and ham, i.e. that the datasets of spam and ham are of same size.

Of course, determining whether a message is spam or ham based only on the presence of the word "replica" is error-prone, which is why Bayesian spam software tries to consider several words and combine their spamicities to determine a message's overall probability of being spam.

IV. Spammer Attack

To further verify the robustness of Cosdes, in this section, we simulate the spammer attack of random HTML tag insertion. We consider the situation that a spammer sends n identical e-mails at a time, where n is varied from 1,000 to 100,000. It is assumed that a sequence of random HTML tags is inserted into the beginning of each e-mail. The number of tags in a sequence is a random number between 1 and 50.5

Regarding the type of tag, we randomly choose them from HTML tag list. around 90 percent of e-mails are matched with other e-mails, meaning that only 10 percent of spams have completely distinct HTML tag sequences as random HTML tag insertion is applied. This is because the sequence preprocessing step of procedure SAG will delete nonempty tags that have no corresponding start tags or end tags. Random HTML tag insertion cannot generate legal tag sequences and thus most tags will be eliminated. Concerning the efficiency analysis, it can be observed in that the sequence preprocessing step incurs very little overhead. This also indicates that if new spammer attack occurs, Cosdes still has capacity to react against it by applying a more sophisticated counter measure.

A. Experimental results:

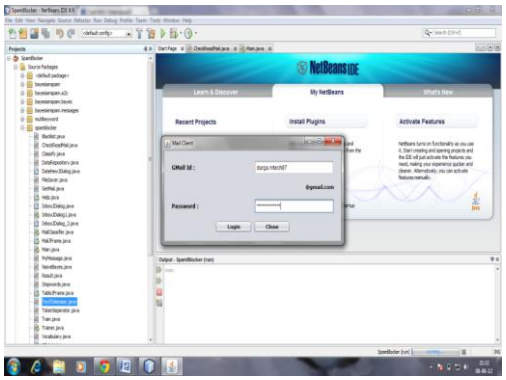

Fig. 3 Seeking username and password

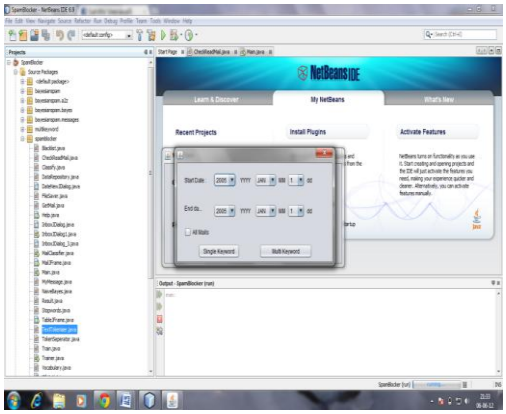

Fig. 4 Specifying spam detection period

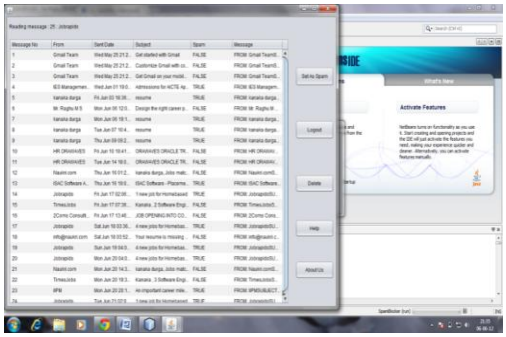

Fig. 5 Showing result.

V. CONCLUSION

In this paper we proposed more sophisticated and robust e-mail abstraction scheme based on Bayesian with new scheme.we can efficiently capture the near duplicate of the spams.Our scheme achieved efficient similarity matching and educed data storage.So we can conclude that our scheme is apt for spam detection.

REFERENCES

[1] J. Hovold, "Naive Bayes Spam Filtering Using Word-Position-Based Attributes," Proc. Second Conf. Email and Anti-Spam (CEAS),2005.

[2] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam Filtering with Naive Bayes—Which Naive Bayes?" Proc. Third Conf. Email and Anti-Spam (CEAS), 2006

[3]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.

[4]S. Sarafijanovic, S. Perez, and J.-Y.L. Boudec, "Resolving FP-TP Conflict in Digest-Based Selection Algorithm," Proc. Fifth Conf. Email and Anti-Spam (CEAS), 2008.

[5] 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.

[6] J.S.Kong,P.O.Boykin,B.A.Rezaei,N.Sarshar,and V.P.Roychowdhury, " Scalable and Rel iable Col laborative Spam Filters: Harnessing the Global Social Email Networks," Proc. Second Conf. Email and Anti-Spam (CEAS), 2005.

[7] E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati,"An Open Digest-Based Technique for Spam Detection," Proc. Int'lWorkshop Security in Parallel and Distributed Systems, pp. 559-564,2004.

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

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

[10] H. Drucker, D. Wu, and V.N. Vapnik, "Support Vector Machines for Spam Categorization," Proc. IEEE Trans. Neural Networks, pp. 1048-1054, 1999.

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

[12] S. Sarafijanovic, S. Perez, and J.-Y.L. Boudec, "Improving Digest-Based Collaborative Spam Detection," Proc. MIT Spam Conf., 2008.

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

[14]D. Sculley and G.M. Wachman, "Relaxed Online SVMs for Spam Filtering," Proc. 30th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), pp. 415-422, 2007.

[15] I. Rigoutsos and T. Huynh, "Chung-Kwei: A Pattern-Discovery-Based System for the Automatic Identification of Unsolicited E-Mail Messages (SPAM)," Proc. First Conf. Email and Anti-Spam (CEAS), 2004.

[16] 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.

ABOUT THE AUTHORS

Mrs. V. KANAKA DURGA is pursuing M.Tech. at Sri Vasavi Engineering college, Tadepalligudem. Her area of interest includes Computer Networks and Network Security.

Mr. M.R. RAJA RAMESH, M.Tech., is working as an Associate Professor in the Dept. of Computer Science & Engineering in Sri Vasavi Engineering College, Tadepalligudem. Presently he is pursuing Ph.D. at Andhra University.