Spam Detection Using Email Abstraction

Bhawana S. Dakhare
IT department ,Bharati Vidyapeeth College Of Engg.Mumbai University,
Navi Mumbai , Mumbai, INDIA

Prof. U. V. Gaikwad
Terna Engineering College,Mumbai University, Nerul
Navi Mumbai , Mumbai, INDIA

Prof. V. B. Gaikwad
Terna Engineering College,Mumbai University, Nerul
Navi Mumbai , Mumbai, INDIA

Abstract- Communication has been increased enormously now a days. Today's generation consider email as a fastest medium of communication within shorter duration and for longer distance. Spam is junk email or email which users do not want in their inbox. For finding out spam mails several methods exist. These methods are broadly classified as context based, noncontext based. Those methods require regular updating of lists which are used for spam detection. A more effective and robust email abstraction scheme is discussed here. The email abstraction system defines a procedure, structured abstract generation, to abstract features from email using html contents of email. The abstraction is then stored in tree structure so that efficient matching and less time requirement is achieved for matching. The performance of email abstraction system is compared with the web page content based spam detection system method. For comparison the different parameters used are precision, recall, specificity and accuracy. By considering all these parameters we find that the email abstraction system performs well.

Keywords – Email, spam, email abstraction.

I. INTRODUCTION

Email is the most widely used medium for communication worldwide because it's fast, cheap, reliable and easily accessible. Spams are anonymous, unsolicited bulk emails sent to internet users, mainly by unknown persons and organizations. It is sent out in mass quantities by spammers who make money from the small percentage of recipients that actually respond. Email marketing is free of cost so it could results in phishing and to spread malicious code. Spam emails can be recognized either by content or delivery manner. The problems from the spam mails would be listed as, wastage of network resources bandwidth, wastage of time, damage to the PC's & laptops due to viruses. It can lead to the ethical issues such as the spam emails advertising pornographic sites which are harmful to the young generations.

A. Need Of Spam Detection-

Spam email is not useful for the recipients and can cause threat to security. So it is significant security problem for computer everywhere. For ex.[1] it may contain a link to a phony website which want to capture the user's login information or other information like identity theft, phishing or a link to a website that installs malicious software (malware) on the user's computer. Installed malware can be used to capture user information, send spam, host malware on others, host phish. So the prevention of spam transmission would be ideal and detection allows users and email providers to address the problem.

The methods for spam identification are Rule based handmade rules for detection of spam made by experts. These methods need domain experts & constant updating of rules. Receiving spam is a lot more problematic than a physical junk mail. If you publish your email id for business purpose then you need a spam filter because large amount of spam it will arrive in inbox, if there is no filter present. If you receive spam on your business email, you have to spend a lot more time for finding through the garbage, a legitimate message and to delete spam. This result in loss of productivity where you could perform a lot more useful for your company. Business users receive more spam because of their email addresses are often posted on public websites. For example business's websites, directories, networking sites, forums, blogs etc. Users frequently post their email so that their clients and business partners can have an easy way of contacting them. So there is need of filters as spammers often search the web using automated tools to find email addresses. If you posted email on many websites, there are more chances it will get picked up by a spammer and not just one spammer or two. It means dozens of spam messages being sent to you every single day.

B. Various Approaches For Spam Detection-

For detection of spam some tools are available eg. [2] Send safe - released in 2003. It uses domain name system and IP addresses for detection of spam messages. Reactor mailer- released in 2007. It uses domain name system, IP addresses and message type for detection of spam messages. SpamAssaissn –[3] it is distributed under the same terms and conditions as other popular open-source software packages such as the Apache web server. Baku Azerbaijan et. al [4] gives the filtering methods that can be divided based on where the database can be maintained on client machine, on mail server or by agent based. We can find SpamAssassin in use in both email clients and servers, on many different operating systems, filtering incoming as well as outgoing email, and implementing a very broad range of policy actions. For spam detection it uses header test, set of rules for body test, automatic and manual white list or black list of address, domain name block list and Bayesian filtering. In Spam Assassin, each filter assigns a message, the credit; if the message's calculated credit is greater than a threshold for considering spam, Spam Assassin classifies it as a spam. Jangbok Kim et. al[5] gives one of method for filtration based on URL. A spammer might use text, for example, write "work" as "<!—a—>w<!—b—>o<!—ff—>rk." Such a modified word is difficult to detect, email clients simply see the word "work." To make matters worse, many recent spams don't include any text. Instead, they contain only images or URLs or both. Obviously, this makes life hard for keyword-based filters. The method spam filtering with dynamically updated URL Statistic in that user need to update regularly the database for URL. In URL based spam filters uses "white" and "black" lists to classify email. The method URL based spam filter instead analyzes URL statistics to dynamically calculate the probabilities of whether email with specific URL is spam or legitimate, and then classifies them accordingly. In this method URL based spam filter based on observing the statistics of URLs in email. Filter uses the naïve bayesian algorithm to decide whether an email is spam or not. When a new email reaches an email system, filter extracts URL and calculates its probability for weather it is spam or ham.

In content based methods: Rui Zhang [6] and [7] Harris Drucker gives method based on classification method for spam using support vector. For Content based methods which is keyword based the keywords are like congrats, congratulations, won, claims, ticket number, serial number, total sum, money, prize, winner, draw, credited, jackpot, worth, lucky, urgent reply, account , balance, dollars, rupees, lottery, valid , secrete, pay, cash, funds, agents, limited days, offers, claimed, unclaimed etc. In this using words present a feature vector is designed. The support vectors define two hyperplanes, one that goes through the support vectors of one class and one go through the support vectors of the other class. The possible disadvantages of SVM's are that the training time can be very large for dataset if there are large numbers of training examples and execution can be low. This method has disadvantage that it is based on text contents in email message.[8]. Some of method uses images for spam detection. Abdolrahman Attar [9] described the image spam how the image spam are and methods for detection.

In keyword based spam detection method [10] author Marco Tulio Ribeiro et al. gives the method, web page content based spam detection. As the spammers always find way to escape from keyword based spam detectors, in this method urls are used as feature for spam detection. As most of the time processing on these features considered as expensive one. In this method, the urls are extracted from email message. The web pages are downloaded and the keywords are extracted from these pages. Using association rules then web pages are categorised as class spam or ham. In web page content based spam detection method the emails which are not having urls can not be identified. One of the problems faced by this method is time changing urls. Another is spammer might change the content of spam message over time using dynamic pages. When a message is delivered, the server hosting the spam web pages is configured to return a simple ham page. As by crawling a url in spam message sent to user, it may contain a method to provide feedback to spammer. There may possibility that spammer embed some kind of encoding within web page method. Hence it may result in threat to security. While extracting features the bulk of messages having number of urls may overload web crawler. Another drawback of this method is that training dataset used in these methods has to be improved regularly. And the main aim of spammer is to motivate users to click on their links and visit their web sites. Hence the URL links can be important to get spam identified. And mostly spammers think that processing with web pages is expensive. So the email can be identified by other method. Hence this method can be used if message is not giving sufficient information for spam detection so that cost of crawling and analysing web page content is reduced.

Joseph S. Kong et al.[11] describe method based on collaborative detection system. Collaborative spam filters use the collective memory of, and feedback from, users to reliably identify spam. That is, for every new spam, some user must first identify it as spam for example, via locally generated blacklists or human inspection, any subsequent user who receives a suspect email can then query the user community to determine whether the message is already tagged as spam. This method uses not the private network but the email network so that it can work as distributed database system so that larger database can be handled and easily query can be evaluated. But this method has security problems as the mail has to be forwarded as it is to next hosts. Even the all methods discussed are not able to applicable to all types of languages. The Collaborative spam filters must fulfill three challenges as Performance, Scalability and trust.

The rest of the paper is organized as follows. Email Abstraction System for spam detection is explained in detail in section II. Experimental results are presented in section III. Concluding remarks are given in section IV.

II. Spam Detection System Using Email Abstraction

The email abstraction system is efficient for spam detection and progressive update system. Each time for new spam mail database is updated.The proposed system consists of :

- The email abstraction system defines a procedure structured abstraction generation to abstract features from email using html contents. The domain name is also retrieved and stored in anchor set so that it can be used for matching purpose.
- For storing email abstraction we design a tree structure of reported spam. Spam trees, the tree structure provide easier method for matching purpose. Matching process consists of two phases approximate and exact matching. The two email messages are same if its subsequence and its hash values from tree are same.
- The system is complete, which consists of regular update for newly detected spam mail. The system also provides action against misclassified hams.

The algorithm for main module is given as:

```
Input : Tm : the maximum time spam for reported spam being retained in system.
        Td : the time spam for triggering deletion handler.
        S : the score threshold for determining spam.
Step 1  : Read email.
Step 2  : if(EA.S>S initial)
            Then the receiving email is spam mail.
            (where EA is tag length  and S initial = 5)
Step 3 : If the read email is spam activate insertion handler.
         The reported spam mail is stored in database.
Step 4 : If the received email is misclassified as spam
         then activate error report handler.
       In this the S initial value is made zero so that same mail is not identified as spam.
Step 5 : On trigger Td , deletion handler is activated.
           Here trigger is some time period (Tm). For making availability for new spam mails previous spam mails
          are deleted.
```

The system consists of main following modules:

1. Abstraction Generation
2. Spam Detection
3. Database Maintenance

Abstraction Generation Module:

In this module we generate an email abstraction. Here we use structure abstraction generation procedure to generate the email abstraction. First read html content type based input mail.

1. Structured Abstraction Generation:

In this module email abstraction is generated using HTML content in email. It 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. <anchor> tags are then inserted into Anchor Set, and tags are concatenated to form the tentative email abstraction.

The following sequence of operations is performed in the pre-processing step.

a. Front and rear tags are excluded.
b. Nonempty tags that have no corresponding start tags or end tags are deleted. Besides, mismatched nonempty tags are also deleted.
c. All empty tags are regarded as the same and are replaced by the newly created <empty> tag.
d. The pairs of nonempty tags enclosing nothing are removed.

For the purpose of accelerating the email matching process, the tag sequence of an email abstraction is reordered in Tag Reordering Phase. The main objective of appending <anchor> tags is to reduce the probability that a ham is successfully matched with reported spam when the tag length of an email abstraction is short.

1.1.1 Tag Extraction phase:

In this phase we read input mail. Find the all HTML tags present in the email. After reading all html tags then we find matching pairs of html tags as html tags comes with pairs ie with opening and closing tags. So after this if there is any only opening tag and closing tag is not there then that tag is deleted. And only tags with pairs are selected. Find the anchor tag having domain name. Add all anchor tags and remaining tags. Then preprocess the tag sequence.

1.1.2 *Tag Reordering Phase:*

In this phase we reorder each tag for accelerating tag matching phase. Since the arrangement of html tags are in pairs various sequential patterns of tags are contained in email. In worst case if we consider two email having same length and only one tag is different. And difference is detected at last tag when each tag is compared. So the tag's regular sequence is changed by reordering the tag sequence. This helps in lowering the number of comparisons. Assign new position number to each tag using given formula. Add all the tags with position numbers (EA).

Using the following formula given:-

b (1)

(2)

(3)

(4)

Where L is length of abstraction, PNorigin is original position of tag, b is number of buckets, r is which bucket should be placed, q is number of shift counts from end of the bucket.

1.1.3 Appending Phase:

Append the anchor set <anchor> in front of EA. Use of this to reduce probability that ham is matched with reported spam.

1.2 Design Of Spam Tree:-

For facilitating email matching, data structure Spam tree is designed. Spam tree is used to store large amount of email abstraction of reported spam. The two email abstraction is same only when the number of their tags is identical. And if the email abstractions are distributed with different tag lengths into diverse Spam Tree the quantity of spams required to be matched will decrease. An email abstraction is segmented into several subsequence and these subsequence are put into corresponding leafs from low levels to high level. As such email abstraction is stored in one path from root node to leaf node of spam tree and hence matching between a testing email and spam is processed from root to leaf. It reduces the number of tags to be required to be matched.

If the tags are stored in tree as they are extracted ie each is on each level of tree then the height of tree will larger as that of length of tags increases. And this kind of matching will be infeasible. And tree will be unbalanced. So to avoid this balanced tree structure is used.

Spam trees are designed to be binary trees. The values for leafs of binary tree are determined by hash function. Tags are stored after reordering phase. As mostly the html tags are in pairs the subsequence are uniformly distributed. In spam tree on level of i each node stores tag length equal to 2^i. On root node anchor tag is stored and as per tags are reordered are located ranging from 2^i to 2^{i+1}-1the last subsequence of each email abstraction in spam tree will be stored in the leaf nodes on same levels. Also the tag lengths of subsequence stored in leaf nodes of leaf nodes of level j range from 1 to 2^j .

1.3 Hash Function:-

For further accelerating matching process Hash function is used to map each subsequence to an integer. The aim of hash value is that subsequence should be exactly matched. The matching process is easier as the subsequences are mapped to an integer value. The hash function used is additive function.

Hash function is:-

$$hash(sequence) = f(seq\ [0] * 2^{m-1}) + f(seq\,[1] * 2^{m-2}) + \ \ldots .+f(seq[m \cdot \qquad (5)$$

where m = number of tags in the subsequence

seq [n] = tag type of n_{th} tag.

Because of this hash function most subsequence matching is transform to integer matching and hence complexity of matching is reduced.

The advantageous features of this method are as follows:

- The height of a Spam Tree is equal to logL, where L is the tag length of the longest email abstraction in this SpamTree.
- As per this design parent nodes store less number of subsequence than children nodes and longer subsequence are put into higher levels (the tag length of the subsequence on level i is 2^i). Thus, the number of tags matched from root to leaf is markedly decreased. Moreover, with the hash function, the matching efficiency is substantially increased.
- The numbers of tags stored in the nodes of SpamTree are expected to be similar, and hence SpamTree are balanced binary trees.

2. Spam Detection Module:

In this module matching handler is present. It detects weather received mail is spam or ham. From the email abstraction the tags are abstracted. Next find corresponding spam tree and spam table where the tags are stored. Using the tree data structure, tag matching can be performed.

2.1 Matching Handler

Matching Handler is the most important module in proposed system to achieve efficient matching between every testing email and the known spam database. There are two major phases in the matching process: Approximate Matching Phase and Exact Matching Phase.

2.1.1 Approximate Matching Phase:-

In this phase without doing exact tag comparisons, spam tree is directly traversed to targeted leaf node on tag length based on type of tags present on 2^i level.

There may be possibility that testing email may merely be same with spams which have the same tag length and are in the same path. Therefore, information of spam is recorded which appear in the targeted leaf node and have the same tag length, into a candidate set candSet.

The main objectives of the approximate matching are:

1) to reduce unnecessary tag comparisons of emails with different tag lengths, and

2) to exclude emails this can be determined without the exact tag matching.

2.1.2 Exact Matching Phase:-

In this phase, in spam tree, hash values of each subsequence are calculated and matched first. Exact matching is performed if hash values are exactly equal. Then unmatched candSet are deleted. The exact matching process is performed on f level, only first $2^{f\ level+1}$ -1 tags are exactly matched. For the purpose of matching S is considered as some threshold value which gives number of subsequences are matched.

Finally, if the sum of S of all candidate spams in candSet exceeds S threshold value, the testing email will be classified as a spam.

Algorithm for matching handler:-

```
Input: EA : the email abstraction of testing email,
S : the score threshold for determining spam
Here we consider threshold value=5
Output: the detection result
Level on which exact matching is performed Var f_level;
From subsequences , Var candSet for exact matching.;
Step 1 : Approximate Matching Phase
i)      Find the corresponding Spam Tree with tag length;
ii)     In the designed spam tree , traverse directly to the targeted leaf node
        based on the types of tags at positions $2^i$;
For (each subsequence in the leaf node)
If(EA.tag_length==subsequence.tag_length)
candSet.insert (subsequence.info);
Step 2 : Exact Matching Phase
i)      From tree data structure consider nowNode for each internal node.
ii)     nowNode = Spamtree.root;
iii)    For (i=0 to f_level)
iv)     For (each subsequence.in candSet)
(the subsequence which was found in approximate matching phase)
if (subsequence.hash_value) == (EA.current.subsequence.hash_value)
        if (subsequence! = EA.currentsubsequence)
```

```
                    candSet.delete(subsequence.info);
                  else candSet. Delete(sunsequence.info);
                 nowNode= the corresponding child node;
v)          sum = the sum of S of all candidate spam in candSet;
vi)         if (sum> S) return spam;
vii)        else    return ham;
Step 4 :  End
```

3. Database Maintenance:

In database maintenance module three handlers are present deletion handler, insertion handler and error report handler. When mail is received email abstraction is performed. Then find Spam tree. Insert all subsequent tags using insertion handler. When any mail is misclassified as spam error report handler is executed.

3.1 Insertion Handler

In the insertion handler the corresponding spam tree is found in Spam tree according to the tag length of the inserted spam. Insert the subsequences of the email abstraction along the path from root to leaf. The subsequence with 2^i tags is inserted into level i of spam tree. Meanwhile, the hash value of this subsequence is computed. The subsequence with remaining tags is stored in database.

3.2 Error Report Handler

When receiving a misclassified ham, Error Report Handler first finds the corresponding Spam tree and does the matching process as the same in Matching Handler. For the spam matched with the reported misclassified ham, the value of S (used for finding spam mail) of these spams is reset as 0 to avoid subsequent misclassification incurred by the identical group of spam. In addition, the reputation scores of reporters who cause the false positive error are halved to prevent continuous attacks by specific users.

The method is compared with web page content based spam detection. In this method first find out patterns hidden in spam messages. And then it associates the discovered patterns with the corresponding class ie spam or legitimate message. These associations are presented in the form of rules X→c, where X is a pattern and c is either ham or spam.

3.3 Deletion Handler

After some time period(Tm), outdated spams are deleted for this deletion handler is used. This Tm is used as trigger for activating deletion handler. For evolving nature of spam it is inappropriate to utilize old spams to filter current ones.

III. Experiment and Result

A. Performance Evaluation Methods-

Finding out accuracy of the spam detection system is to find out how accurately it detects the spam from test dataset. The data set used for testing is spam assassin [3] which is publically available. The dataset contains spam and legitimate emails. The total numbers of emails tested are 500. Performance is measured using parameters as precision, recall, accuracy. Precision used for finding how accurately spam mails are identified as spam mails. Sensitivity and specificity used for how non spam mails are accurately identified as non spam mails. True Positive (TP), states the number of spam mails correctly classified as spam. True Negative (TN) states the number of non spam mails correctly classified as non spam. False Positive (FP) states the number spam mails classified as non spam. False Negative (FN) states the number of non spam mails classified as spam. Accuracy gives performance of system. The tables show that the email abstraction system detects more spam that web page content based method. And increase in dataset the accuracy of system is improved.

TABLE I - For data set of 100 email messages

Measure	**Web page content based spam detection in %**	**Spam detection using email abstraction in %**
$\text{precision} = \frac{TP}{TP + FP}$	**15.78**	**62.0**
$\text{Recall} = \frac{TP}{TP + FN}$	**32.14**	**80.51**
$\text{specificity} = \frac{TN}{TN + FP}$	**11.11**	**69.10**

	18.29	**73.5**

TABLE II - For data set of 500 email messages

Measure	**Web page content based spam detection in %**	**Spam detection using email abstraction in %**
$\text{precision} = \frac{TP}{TP + FP}$	**21.34**	**72.39**
$\text{Recall} = \frac{TP}{TP + FN}$	**45.23**	**85.78**
$\text{specificity} = \frac{TN}{TN + FP}$	**11.39**	**76.12**
	23.14	**80.2**

IV.CONCLUSION

The main aim of our extensive study of automated spam filtering is to develop a devoted filter. Which provide efficient method for spam problem in order to improve the blocking rate of spam emails. The email abstraction system provides certainty to detect the evolving nature of spam. The email abstraction system gives a complete spam detection system which can efficiently process the matching. The email abstraction system is compared with web page content based spam detection method. Web page content based is text based method which cannot work if the words are misspelled and the time required for this method is more than other methods. From the performance measurement it is observed that precision value is improved using email abstraction method ie. from the dataset the spam mails that are positively identified as spam mails. And the rate of identification of ham mails as spam is lesser. The results show that with the increase in dataset the performance is improved. The limitations of web page content based are overcome in email abstraction as it does not work on url. Hence spam detection using email abstraction gives better performance so that it can be utilized for real world applications.

REFERENCES

[1] M. Basavaraju Research Scholar, Dept. of CSE, CIT, Anna University, Coimbatore, Tamilnadu., INDIA Professor & Head, Dept. of CSE, Atria Institute of Technology, Bengaluru, Karnataka., INDIA Dr. R. Prabhakar Professor-Emeritus Dept. of CSE, Coimbatore Institute of Tech., Coimbatore, Tamilnadu, " A Novel Method of Spam Mail Detection using Text Based Clustering Approach" INDIA International Journal of Computer Applications (0975 – 8887) Volume 5– No.4, August 2010.

[2] Elm Ave. San Bruno, "A Survey of Modern Spam Tools" Henry Stern Cisco IronPort Systems 950 CA, 94066.

[3] http://wiki.apache.org/spamassassin/SpamAssassin

[4] Baku Azerbaijan ,Saadat Nazirova Institute of Infotech Technology of Azerbaijan National Academy of Science, published Online August 2011 (http:// www. SciRP. org/ journal/cn) accepted May 15,.Communication and Network 2011,3,153-160, 2011.

[5] Jangbok Kim,Kim, Kihyun Chung, and Kyunghee,Ajou University: "Spam Filtering With Dynamically Updated URL Statistics", Published By The IEEE Computer Society 1540-7993/07/$25.00 © 2007 IEEE, IEEE Security and policy 2007 .

[6] Rui Zhang,Wenjian Wang,Yichen Ma,Changqian Men, 2009, "Least Square Transduction Support Vector Machine", Published online Springer: 28 February 2009 © Springer Science Business Media, LLC. 2009.

[7] Harris Drucker, 1999, "Support Vector Machines for Spam Categorization", Senior Member, IEEE, Donghui Wu, Student Member, IEEE, and Vladimir N. Vapnik, , IEEE Transcation On Neural Networks, Vol. 10, No. 5, SEPTEMBER 1999.

[8] Thiago S. Guzella , Walmir M. Caminhas, " A review of machine learning approaches to Spam filtering" Expert Systems with Applications 36 (2009) 10206,10222 0957-4174/$ see front matter 2009 Elsevier Ltd. All rights reserved.]2009.

[9] Abdolrahman Attar ,Reza Moradi Rad ,Reza Ebrahimi Atani, "A survey of image spamming and filtering techniques", © Springer Science+Business Media B.V. 2011.

[10] Marco Túlio Ribeiro, Pedro H. Calais Guerra, Leonardo Vilela, Adriano Veloso, Dorgival Guedes∗, Wagner Meira Jr. Marcelo H.P.C Chaves, Klaus Steding-Jessen, Cristine Hoepers "Spam Detection Using Web Page Content: a New Battleground" CEAS 2011 - Eighth annual Collaboration, Electronic messaging, AntiAbuse and Spam Conference September 1-2, 2011, Perth, Western Australia, Australia 2011 ACM 978-1-4503-0788-8.

[11] Joseph S. Kong, Behnam A. Rezaei, Nima Sarshar, and Vwani P. Roychowdhury, 2006, " Collaborative Spam Filtering Using E-Mail Networks", University of California, Los Angeles P. Oscar Boykin University of Florida: 0018-9162/06/$20.00 © 2006 IEEE Publised by the IEEE Computer Society.