Spam Detection and Filtering using Different Methods

Bhawana S.Dakhare
ME Computers SEM III
Terna Engg.College, Nerul, Navi Mumbai
Mumbai University ,Mumbai

Ujwala V.Gaikwad
Assistant Professor
Terna Engg.College, Nerul, Navi Mumbai
Mumbai University, Mumbai

ABSTRACT

Spam is an unsolicited bulk mail or junk email. Due to increased communication within shorter duration and for longer distance and fastest medium email is considered .In the recent years spam became as a big problem of Internet and electronic communication. So for overcoming these problems some techniques are developed to fight with them. In this paper the overview of existing e-mail spam filtering methods are compared. In this survey paper we focus on the classification, evaluation, and comparison of traditional methods. The methods discussed are Collaborative Spam Filtering Using E-Mail Networks, Support Vector Machines and Spam Filtering with Dynamically Updated URL Statistics. The methods are compared and performance is evaluated.

Keywords

Collaborative spam filtering, Spam, Support vector machine.

1. INTRODUCTION

1.1 What is Spam?

Spam emails are emails that the receiver does not wish to receive. For increased communication emails are used so one of the best way for advertises emails are considered and as a result spams are generated. Increasingly today large volumes of spam emails are causing serious problems for users, Internet Service Providers, and the whole Internet backbone. Spam emails not only waste resources such as bandwidth, storage and computation power, but also the time and energy of email receivers who must search for legitimate emails among the spam and take action to dispose the spam. The different methods are available. One of the SpamAssassin tool is a widely used host-level filter. This is a rule-based filter that requires constantly changing for the rule to be effective. [2]But some of the attackers figure out the rule being employed and bypass these filters by appropriately constructing the email. Rest of the paper is outlined as Section 1.2 discuss what features can be extracted from email, section 1.3 classification of filtering depending on scope, section 2 different methods for filtration ,section 3 comparison of methods, section 4 Conclusion and the references.

1.2Feature Extraction from Email Message

The mail messages can be filtering by separately by just checking some words on basis of keyword filtering or in groups i.e. a filter may consider that the arrival of a dozen of substantially identical messages in 5min is more suspicious than the arrival of one message with the same content. A filter which involves user collaboration receives also multiple user judgments about some of the new messages for the analysis. As shown in Figure.1 (a) and (b).[3] An email message consists of two parts body and header .Message body consists of text natural language ,possibly with HTML language and graphical elements.

Header is consisting of structured set of fields having name , values and specific meaning. Some of this fields, like From, To, or Subject, are standard, and others may depend on the software involved in message transmission, such as spam filters installed on mail servers. Subject field contains what the user sees as the subject of the message and is often treated as a part of the message body. The body is sometimes referred to as the content of the message. The non-content features are not limited to the features of the header. For methods of message analysis its designer must choose way of doing feature extraction, for deciding what parts of message are used for analysis.

The simplest way is to represent the message as an unstructured set of tokens namely sequences of characters separated by spaces and punctuation marks. This model can be used to characterize any part of a message, or a message as a whole. In this case, presence of a certain word in the message is considered a binary feature of the message. A somewhat more sophisticated approach is to consider the occurrences of the same word in different parts of the message eg. say, ‘John’ in the message body and ‘John’ in the ‘From’ field as different features. For the message header analysis, more sophisticated ways of selecting features take the header structure into account, extracting only some special kind of information. Some of the methods are based on non-content features, including features extracted from the header, such as sender and recipient email names, domain names and zones, and general characteristics of the message, such as the message size and the number of attachments. Some of the methods uses graphics or images for analysis instead of text. The analysis is performed on checking presence of certain predefined tokens in message body(key word filtering) or in the information about sender (blacklist/white list filtering).

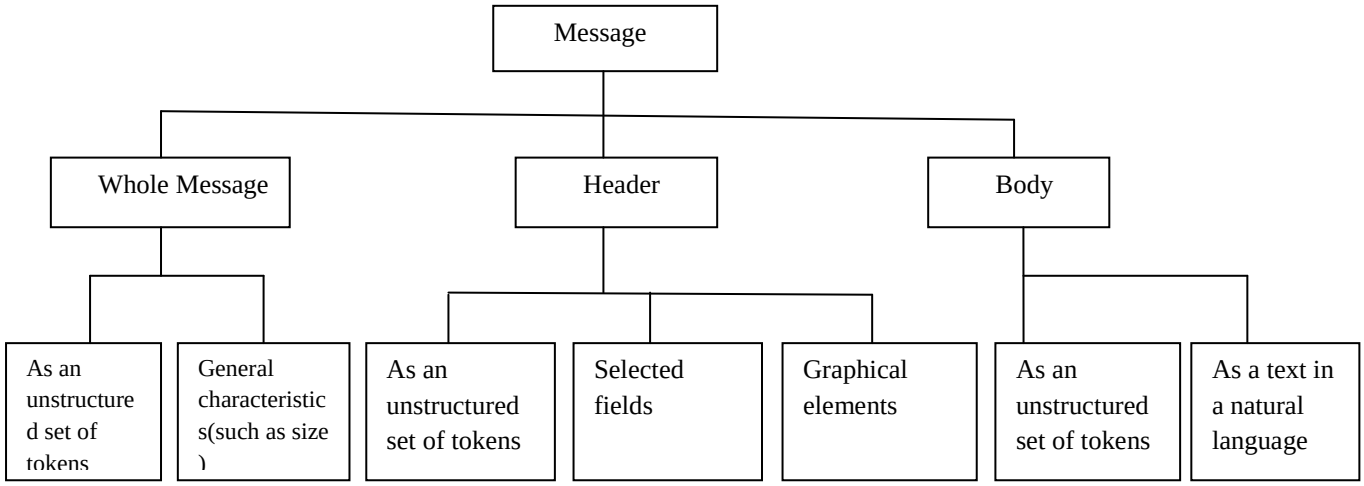

Figure: 1(a) Feature Extraction

1.3Classification of Spam Filtering Methods Depending on Filtration Scope :

Depending on filtration scope spam filtration methods are divided into the following categories[4].

1.3.1 Client Side/Personal Filters:

Client side filters works directly on user's computer. In client side filtration email loading to the user's local computer. In client side filters users' personal information are used, in server side filters the filtration model is defined at once for all users. In spite of the fact that for the majority of users it is obvious what is spam, the concept of spam for each of them is enough personified. The email message marked as spam by someone may be the important information for other one. On the other hand, use of personal model of email classification involves an inevitable overhead cost. Firstly the user should construct his personal model of filtration himself as only he can define what legal email is, and what spam is for him. Secondly, construction, storage and use of personal model demands additional computing resources.

1.3.2 Server Side/General Filters:

Server side filters work at mail server level. Generally in server side filtration systems the traditional methods of filtration are applied Server side filtration also own priority. The centralized solution reduces expenses and simplifies support and control of this system. User becomes more mobile and simplified so that it is easier to store mail centralized in server and to have an access to him from different points, using different devices.

1.3.3 Spam Filtering In Public Mail Servers

This solution sometimes is better than client or server solution. In this case users are mobile as in case of server side filtration, and personalized as in case of client side solution. But disadvantage of usage of public mail servers is that users depend on filtration product installed there. For example, the mail server of Google. Inc company gmail.com uses its own products against spam . This system considers personal information about user to minimize false positives. The public mail provider Mail.ru uses Kaspersky Anti-Spam product based on "Spamtest" technology, and absolutely based on traditional filtration methods.

The different methods are listed here. There are several popular content filters such as Bayesian filters, Rule Based Filters, Support Vector Machines (SVM) and Artificial Neural Network (ANN). Many machine learning approaches have been explored for this task. For example rule-based methods, such as Ripper , PART, Decision tree, and Rough Sets etc. However, pure rule-based methods have not achieved high performance because spam emails cannot easily be covered by rules, and rules do not provide any sense of degree of evidence. Besides Bayesian methods, other machine learning methods, including Support Vector Machine (SVM), Rocchio, kNN and Boosting, have also been applied in the context of anti-spam filtering. Spam filtering is required for not only technical reasons such as overspend the network bandwidth and email storage, but also social issues such as child safety, phishing email, and so on. Spam makes users look through and sort out additional email, not only wasting their time and causing loss of work productivity, but also irritating them and, as many claim, violating their privacy rights .Spam causes legal problems by advertising.

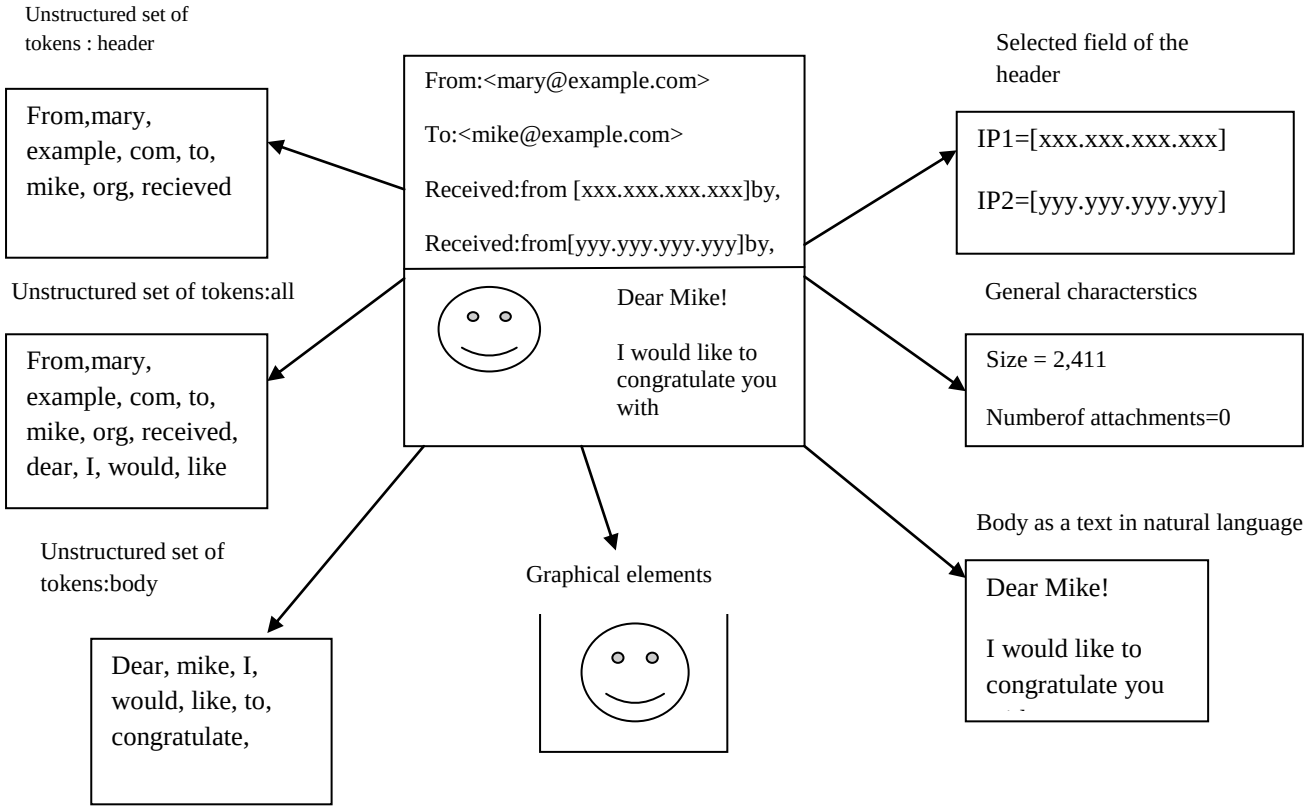

Figure: 1(b) Message Structure from Point of Feature Extraction

2. Different Methods Use for Filtration

Some of methods are discussed here for filtration:

2.1 Spam Filtering With Dynamically Updated URL Statistics:

Many URL-based spam filters rely on "white" and "black" lists to classify email. [6] The proposed method URL-based spam filter instead analyzes URL statistics to dynamically calculate the probabilities of whether email with specific URLs are 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, E, reaches an email system, our filter extracts the email's URLs and host names (h1, h2, …, hn). Multiple appearances of identical hi are treated as a single appearance of hi. The filter then calculates two probabilities: that the email is spam, P(Spam|E), and that it is legitimate, P(Legitimate| E). We calculate these probabilities using a frequency table and naïve Bayesian algorithm. If P(Spam|E) is greater than P(Legitimate|E), the filter classifies the email as spam and pushes it into its spam pool. Otherwise, it considers the email legitimate and sends it to the client. Periodically, the filter sends the list of spam in the pool to the email clients so they can recover any misclassified email. If the filter can't calculate an email's probabilities, it classifies it as legitimate. Comparison with other filters: We compared our filter with SpamAssassin on the same email set. SpamAssassin is a collaborative filter that combines more than 20 filters including keyword-based, Bayesian, and URL based Filters to classify email. In SpamAssassin, each filter assigns a message a credit; if the message's accumulated credit is greater than a threshold, SpamAssassin classifies it as a spam.

2.2 Support Vector Machines:

Support vector machines (SVMs) can classify objects by projecting them into a n-dimensional space. [7]The dimensional size is determined by the number of characteristics of the training or query vector. The actual classification is done by filling the vector space with labeled elements from the training set and creating a hyperplane that separates the points according to their labels. A query can then be categorized by simply projecting it into the same space and determining on which side of the plane it resides. For this method execution speed is very fast but has disadvantage is that the training time more if there are large number of examples.[10]The key concepts use are the following: there are two classes $y_i = €\{-1,1\}$, and there are N $(x_1,y_1),….(x_N,y_N)$, $x € R^d$ where d is dimensionality of vector. If the two classes are linearly separable ,then one can find an optimal vector w* such that $\|w^*\|^2$ is minimum and $W^* \bullet x_i - b \geq 1$ if $y_i = 1$ and $w^* \bullet x_i - b \leq -1$ if $y_i = -1$ or equivalently $y_i (w^* \bullet x_i - b) \geq 1$. Training examples that satisfy the equality are termed support vectors. The support vectors define two hyperplanes, one that goes through the support vectors of one class and one goes through the support vectors

of the other class. The distance between the two hyperplanes defines a margin and this margin is maximized when the norm of the weight vector ||w*|| is minimum. Authors shows minimization and maximizing the following function with respect to variable α_j : $W(\alpha) = \sum\alpha_i - 0.5 \sum \sum \alpha_i \alpha_j(x_i \bullet x_j) y_i y_j$ subject to constraint : $0 \leq \alpha_j$ where it is assumed that N are training examples, x_i is one of the training vectors, and $\bullet$ represents the dot product. The advantage of the linear representation is that w* can be calculated after training and classification amounts to computing the dot product of this optimum weight vector with the input vector.

2.3 Collaborative Spam Filtering Using E-Mail Networks:

Collaborative spam filters use the collective memory of, and feedback from, users to reliably identify spam. [8]That is, for every new spam sent out, some user must first identify it as spam for example, via locally generated blacklists or human inspection; any subsequent user who receives a suspect e-mail can then query the user community to determine whether the message is already tagged as spam. In this method spam-filtering system uses two key mechanisms to exploit the topological properties of social e-mail networks: the novel percolation search algorithm, which reliably retrieves content in an unstructured network by looking through only a fraction of the network, and the well -known digest-based indexing scheme.

Percolation search: search algorithm: This algorithm passes messages on direct links only and includes three key steps: This algorithm passes messages on direct links only and includes three key steps:

Cache or content implantation: Each node performs a short random walk in the network and caches its content list on each visited node. The length of this short random walk is referred to as the time to live (TTL).

Query implantation: A node making a query executes a short random walk of the same length as the TTL used in the content implantation process and implants its query requests on the nodes visited.

Bond percolation: The algorithm propagates all implanted query requests through the network in a probabilistic manner; upon receiving the query, a node relays it to each neighboring node with percolation probability p, which is a constant multiple of the percolation threshold, pc, of the underlying network.

It consists of following functions: Digest publication. If the client program determines that the e-mail is definitely spam, it calls the digest function to generate a digest, De, for the message and caches the digest on a short random walk of length l, which is the TTL.

Query implantation : If the client program suspects that the e-mail is spam, it can query the system to determine whether any other user in the network already has De on its spam list. It implants each query message for this digest via a random walk of length l, node receives a suspected message and implants a query via a random walk with a TTL equal to 2.

Bond percolation : Nodes with an implanted query request percolate the query message containing De through the e-mail contact network. Each node that the query visits declares a hit if the digest matches any messages cached on that node.

Hit routeback :The client program routes all hits back to the node that originated the query through the same path by which the query message arrived at the hit node.

The system routes the hits at nodes and Other back to first node through the same path.

Hit processing :After routing all hits back, the client program calculates the number of hits received. If this HitScore exceeds a constant threshold value, the program declares the message in question as spam; otherwise, it determines the message not to be spam. The client program places all e-mail messages declared as spam in the user's spam folder. It then calls the function that generates the digest of the spam message, De, and caches this on a short random walk, taking the process back to the digest publication step. That the system exchanges all messages via background e-mails. Users are not required to click and open any system message or file. Moreover, the system can program clients to reject all messages that do not match a predefined format and thus are potentially malicious. Finally, we recommend adding a personalization feature that lets the user blacklist only spam addressed to the public.

3. COMPARISION OF METHODS

Email spam is the bulk, promotional, and unsolicited message. Email spam causes a serious problem in waste of time and resources. In this paper we surveyed existing techniques and algorithms created to fight against web spam. Discussed how spam affects users and search engine companies, and motivate academic research. Then we turn to the discussion of algorithms for web spam detection, and analyze their characteristics and underlying ideas. At the end, we summarize all the key principles behind anti-spam algorithms:

Support vector machine is a powerful and popular machine learning tool in solving supervised classification problems due to its good generalization performance. The AT&T staff worked on 3000 email messages using SVM found that 850 messages were considered as spam, and at the time of experiments it consider body of message. It shows that method will be if it consider binary features. The training time required is more in SVM. The accuracy can be improve by generating list of acceptable senders which are considered as non spam no matter what the subject and body contents. A major drawback of collaborative filtering schemes is that they ignore the already present and pervasive social communities in cyberspace and instead try to create new ones of their own to facilitate information sharing. The method anti spam system is social-network-based, it is important to protect users' privacy by preventing anybody from using the network to map out social links. The requirement is to be able to provide enough benefits to users to encourage their participation, which is relatively easy when it comes to spam management. If users become accustomed to a spam-filtering system, queries for other information will follow. In Spam Filtering With Dynamically Updated URL Statistics extracted URL information from all URLs in the email and, where

relevant, extracted HTML tags. used 39,965 email messages for the study; about 95 percent were spam. About 80 percent of legitimate email contained one or more URLs, and 99 percent of spam contained URLs. In addition to advertising, spam might have such a high rate of URLs because many include only figures (with URLs) to avoid content filters. We compared legitimate and spam email based on the number of URLs that pointed to figures or linked to Web pages. If a URL was included in HTML IMGE or AREA tags, we classified the URL as representing an image. If a URL accompanied an A, FORM, INPUT, LINK, BUTTON, IFRAME, OPTION, or NIL tag, we classified it as linking to a Web page. Although spam includes many URLs with image and linking tags, our statistics still don't offer a clear-cut case for using tags to distinguish between spam and legitimate email. The given method is compared with SPAM ASSASSIAN filter with same emails it found better performance.

4. CONCLUSION

The rapid growth of users in the Internet and the abuse of e-mail by unsolicited users cause an exponential increase of e-mails in user mailboxes. Spam messages are nuisance and huge problem to most users since they clutter their mailboxes and waste their time to delete all the junk mails before reading the legitimate ones. They also cost user money with dial up connections; waste network bandwidth and disk space. Summarizing the article , In this paper we discussed the problem of spam. And how spam can be detected.[11][12] We surveyed different existing techniques and algorithms to fight against spam. And compared the methods. In SVM based method which is content based the training required which required time and accuracy is less. In collaborative system, all system has to participate in communication and due to disadvantages in this method URL method can be considered better. As URL can be mostly occurring factor in email. So using this method, spam filtering can have better performance.

REFERENCES

[1] Bin Wang Æ Gareth J. F. Jones Æ Wenfeng Pan, "Using Online Linear Classifiers to Filter Spam Emails", Springer-Verlag London Limited 2006, Published online: 3 October 2006.

[2] Venkatesh Ramanathan and Harry Wechsler,"Phishing Detection Methodology Using Probabilistic Latent Semantic Analysis", AdaBoost, and co-training EURASIP Journal on Information Security, 2012:1,phishGILLNET, 2012.

[3] Enrico Blanzieri,Anton Bryl, 10 July 2009, "A Survey Of Learning-Based Techniques of Email Spam Filtering", Springer, Published online: 10 July 2009 © Springer Science+ Business Media B.V. 2009

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

[5] Wuying Liu Ting Wang, 2011, "Online active multi-field learning for efficient email spam Filtering", Springer Received: 5 August 2010 / Revised: 10 October 2011 / Accepted: 15 November 2011 © Springer- Verlag London Limited 2011.

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

[7] Manuel Egele, Clemens Kolbitsch, Christian Platzer, 2009, "Removing Web Spam Links From Search Engine Results", Springer ,Received: 22 December 2008 / Accepted: 3 August 2009 / Published online: 22 August 2009 © Springer- Verlag France 2009`06.

[8] 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.

[9] 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.

[10] 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.

[11] Taiiki Takashita,Tsuyoshi Itokawa,Teruaki Kitasuka, and Masayoshi Aritsugi,2008,"A Spam Filtering Method Learning From Web Browsing Behavior",Springer-Verlag Berlin Hedelberg 2008.

[12] Chi-Yao Tseng, Pin-Chieh Sung, and Ming-Syan Chen,Fellow IEEE,"A Collaborative Spam Detection System with a novel E-mail Abstraction Scheme", IEEE Transactions on Knowledge and data engineering,Vol 23,no5,may 2011.