Volume 3, Issue 10, October 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Survey on Machine Learning Methods in Spam Filtering

Vinod Patidar
Student (M. Tech.),
CSE Department, BUIT, Bhopal, India

Divakar Singh
HOD, CSE Department, BUIT,
Bhopal, India

Abstract: Email spam or junk e-mail (unwanted e-mail "usually of a commercial nature sent out in bulk") is one of the major issue of the today's Internet, that cause financial damage to companies and annoying individual users. Among the approaches developed to stop spam, filtering is an important and popular one. Common uses for mail filters include organizing incoming email and removal of spam and computer viruses. A less common use is to inspect outgoing email at some companies to ensure that employees comply with appropriate laws. Users might also employ a mail filter to prioritize messages, and to sort them into folders based on subject matter or other criteria. Mail filters can be installed by the user, either as separate programs, or as part of their email program (email client). In email programs, users can make personal, "manual" filters that then automatically filter mail according to the chosen criteria. In this paper, we present a survey of the performance of four commonly used machine learning methods in spam filtering. Most email programs now also have an automatic spam filtering function.

Keywords: E-mail classification, Spam, Spam filtering, Machine learning, algorithms.

1. INTRODUCTION

In recent years, e-mails have become a common and important medium of communication for most Internet users. However, spam, also known as unsolicited commercial/ bulk e-mail, is a bane of e-mail communication. Spam is commonly compared to paper junk mail. However the difference is that junk mailers pay a fee to distribute their materials, whereas with spam the recipient or ISP pays in the form of additional bandwidth, disk space, server resources, and lost productivity. If spam continues to grow at the current rate, the spam problem may become unmanageable in the near future. A study estimated that over 70% of today's business e-mails are spam [1]; therefore, there are many serious problems associated with growing volumes of spam such as filling users' mailboxes, engulfing important personal mail, wasting storage space and communication bandwidth, and consuming users' time to delete all spam mails. Spam mails vary significantly in content and they roughly belong to the following categories: money making scams, fat loss, improve business, sexually explicit, make friends, service provider advertisement, etc.[2], One example of a spam mail is shown as Figure 1.

E-mail users spend and increasing amount of time reading message and deciding whether they are spam or not and categorizing them into folders. E-mail service providers would like to relieve users from this burden by installing server-based spam filters that can classify e-mails as spam automatically. [3] Spam filtering classification due the following reasons:

- Continually changing – Spam is constantly changing as spam on new topics emerges. Also, spammers attempt to make their messages as indistinguishable from legitimate email as possible and change the patterns of spam to foil the filters. [4]
- False positives problem – false positives are simply unacceptable; thus the requirements on the spam filter are very exacting.
- OCR computational cost – the OCR computational cost in text embedded in images compatible with the huge amount of e-mails handled daily by server-side filter. [4]
- The use content obscuring techniques – Spammers are applying content obscuring techniques to images (see Figure 2), to make OCR systems ineffective without compromising human readability. [5]

Delivered-To: vinod.patidar1@gmail.com

Received: by 10.60.65.70 with SMTP id v6csp179624oes;
Wed, 17 Jul 2013 08:32:21 -0700 (PDT)
X-Received: by 10.236.223.229 with SMTP id v95mr3005941yhp.54.1374075140703;
Wed, 17 Jul 2013 08:32:20 -0700 (PDT)

Date: Wed, 17 Jul 2013 13:08:38 +0100
From: indian supplier needed urgent <supply@info.org>
Reply-to: peterowen78@outlook.com

To: undisclosed-recipients:;
Subject: indian supplier needed urgent
MIME-Version: 1.0
Content-Type: multipart/alternative;
boundary="=_1cvcrvfx704k"
Content-Transfer-Encoding: 7bit
User-Agent: Internet Messaging Program (IMP) H3 (4.1.6)

This message is in MIME format.

I am greeting you, are you from India? I got your contact from India international business and investment daily then decided to contact you direct. Our Company use to purchase industrial chemicals and natural Pharmaceutical Products, from South Africa and India but it is very scarce now in South Africa
but the products are cheaper in India. I can't come to India because of my new promoted post and I will want you to act as the direct dealer. our company's general director has asked me to contact the local dealer in India, for them to send the new purchasing manager to India to purchase the product directly from the local dealer in India.

I will present you to our company as the local dealer & supplier in India where I was purchasing this products. You would now purchase the products from the local dealer who I used to buy from, and supply to our company as the direct dealer. The profit would be shared between you and me. This
business is very important to me as it is a continuous business.

After purchase from original local seller you would sell to our new purchasing manager, then we share the profit. Your role must be played perfectly and the least I expect from you is betrayal.I don't want my organization to know the real cost of the product for obvious reasons. Please take out a moment of your very busy schedule to respond back to me for more details if you are interested to source and supply to us via this email peterowen78@outlook.com
regards
DR PETER OWEN
BUSINESS DIRECTOR
NOVAS PHARMACEUTICAL LTD UK
LONDON.
peterowen78@outlook.com

Figure 1: An example of a spam mail.

1.1 What is Spam?

Spam is unsolicited and unwanted email from a stranger that is sent in bulk to large mailing lists, usually with some commercial nature sent out in bulk. Some would argue that this definition should be restricted to situations where the receiver is not especially selected to receive the email – this would exclude emails looking for employment or positions as research students for instance. This difficulty in definition demonstrates that the definition depends on the receiver and strengthens the case for personalized spam filtering [1, 2].

Figure 2: An example of a spam mail.

1.2 Structure of an E-mail

In addition to the body message of an e-mail, an e-mail has another part called the header (see Figure 3). The job of the header is to store information about the message and it contains many fields, for example, tracing information about which a message has passed:

- Received: authors or persons taking responsibility for the message
- From: intending to show the envelop address of the real sender as opposed to the sender used for replying
- Return-Path: unique of ID of this message
- Message-ID: format of content
- Content-Type: format of content etc.

```
Delivered-To: vinod.patidar@gmail.com
Received: by 10.60.65.70 with SMTP id v6csp179624oes;
      Wed, 17 Jul 2013 08:32:21 -0700 (PDT)
X-Received: by 10.236.223.229 with SMTP id v95mr3005941yhp.54.1374075140703;
      Wed, 17 Jul 2013 08:32:20 -0700 (PDT)
Return-Path: <supply@info.org>
Received: from mail.credipaz.com (mail.credipaz.com. [200.81.250.66])
      by mx.google.com with ESMTPS id d69si2397346yhh.179.2013.07.17.07.47.46
      for <multiple recipients>
      (version=TLSv1 cipher=RC4-SHA bits=128/128);
      Wed, 17 Jul 2013 08:32:20 -0700 (PDT)
Received-SPF: neutral (google.com: 200.81.250.66 is neither permitted nor denied by best guess record for domain of supply@info.org)
client-ip=200.81.250.66;
Authentication-Results: mx.google.com;
     spf=neutral (google.com: 200.81.250.66 is neither permitted nor denied by best guess record for domain of supply@info.org)
smtp.mail=supply@info.org
Received: from mail (localhost [127.0.0.1])
                 by mail.credipaz.com (8.13.8/8.13.8) with ESMTP id r6HEkZ45031327;
                 Wed, 17 Jul 2013 11:47:40 -0300
Received: (from apache@localhost)
                 by mail (8.13.8/8.13.1/Submit) id r6HC8d7R000816;
                 Wed, 17 Jul 2013 13:08:39 +0100
X-Authentication-Warning: mail: apache set sender to supply@info.org using -f
Received: from 116.202.23.99 ([116.202.23.99]) by mail.credipaz.com (Horde
                 MIME library) with HTTP; Wed, 17 Jul 2013 13:08:38 +0100
Message-ID: <20130717130838.3oyp7sfcyokooogs@mail.credipaz.com>
Date: Wed, 17 Jul 2013 13:08:38 +0100
From: indian supplier needed urgent <supply@info.org>
Reply-to: peterowen78@outlook.com
To: undisclosed-recipients:;
Subject: indian supplier needed urgent
MIME-Version: 1.0
Content-Type: multipart/alternative;
                 boundary="=_1cvcrvfx704k"
Content-Transfer-Encoding: 7bit
User-Agent: Internet Messaging Program (IMP) H3 (4.1.6)

This message is in MIME format.

--=_1cvcrvfx704k
Content-Type: text/plain;
                 charset=ISO-8859-1;
                 DelSp="Yes";
                 format="flowed"
Content-Description: Plaintext Version of Message
Content-Disposition: inline
Content-Transfer-Encoding: quoted-printable



  --
```

Figure 3: Illustrates an example of the header in an e-mail.

1.3 Spam Filtering

Spam filtering in Internet email can operate at two levels, an individual user level or an enterprise level (see Figure 4). An individual user is typically a person working at home and sending and receiving email via an ISP. Such a user who wishes to identify and filter spam email installs a spam filtering system on her individual PC. This system will either interface directly with their existing mail user agent (MUA) (more generally known as the mail reader) or more typically will act as a MUA itself with full functionality for composing and receiving email and for managing mailboxes.

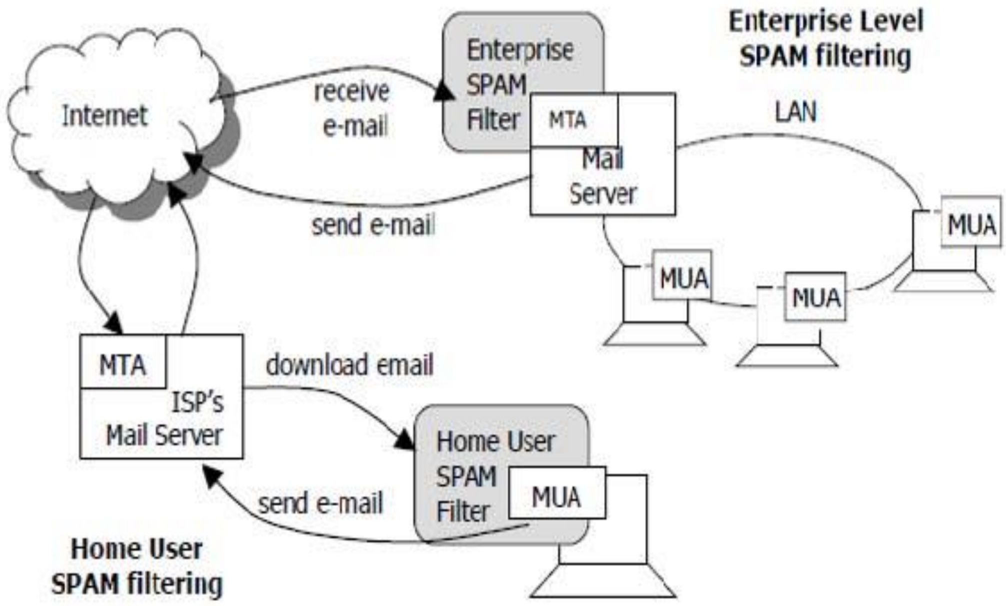

Figure 4: Alternatives for spam filtering in Internet e-mail.

Enterprise-level spam filtering filters mail as it enters the internal network of an enterprise. The software is installed on the mail server and interacts with the mail transfer agent (MTA) classifying messages as they are received. Spam email, which is identified by the enterprise spam filter, will be categorized as a spam message for all users on that network. Spam can be filtered at an individual level on a LAN also. A networked user can choose to filter spam locally as it is downloaded to their PC on the LAN by installing an appropriate system. The vast majority of current spam filtering systems uses rule-based scoring techniques. A set of rules is applied to a message and a score accumulates based on the rules that are true for the message. Systems typically include hundreds of rules and these rules need to be updated regularly as spammers alter content and behavior to avoid the filters. Systems also incorporate list-based techniques where messages from identified users or domains can be automatically blocked or allowed through the filter. If the score for an email exceeds a threshold, the email is classified as spam. Limited learning capabilities are beginning to appear in systems such as Mozilla and the MacOS X Mail program but these systems are still in their infancy. Naïve Bayes seems to be the technique of choice for adding a learning capability to commercial spam filtering systems. The architecture of spam filtering is shown in Figure 5. Firstly, the model will collect individual user emails which are considered as both spam and legitimate email. After collecting the emails the initial transformation process will begin. This model includes initial transformation, the user interface, feature extraction and selection, email data classification, and analyzer section. Machine learning algorithms are employed at last to train and test whether the demanded email is spam or legitimate [4].

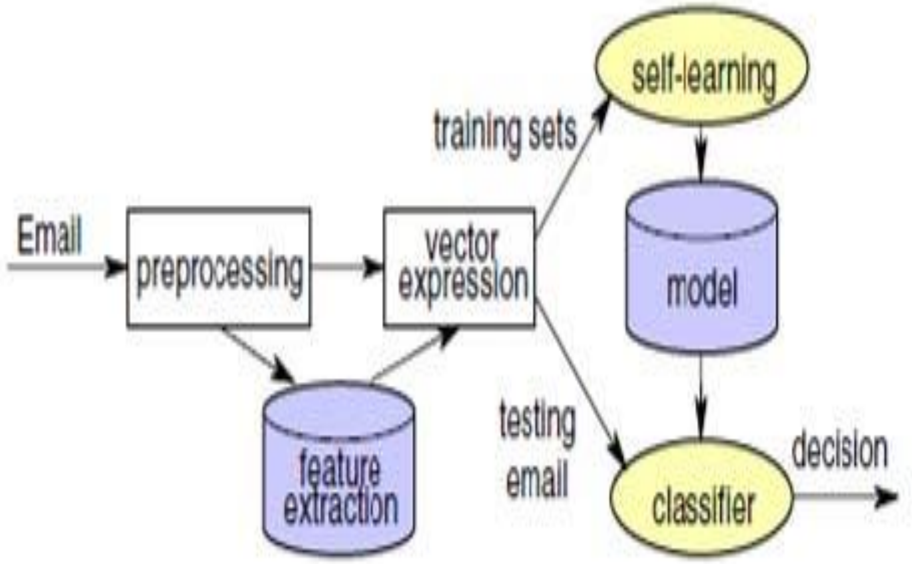

Figure 5: The process of spam filtering.

2. SPAM TECHNIQUES

If a marketer has one database containing names, addresses, and telephone numbers of prospective customers, they can pay to have their database matched against an external database containing email addresses. The company then has the means to send email to persons who have not requested email, which may include persons who have deliberately withheld their email address [6].

2.1 Image Spam

Image spam is an obfuscating method in which the text of the message is stored as a GIF or JPEG image and displayed in the email. This prevents text based spam filters from detecting and blocking spam messages. Image spam was reportedly used in the mid 2000s to advertise "pump and dump" stocks.[7] Often, image spam contains nonsensical, computer-generated text which simply annoys the reader. However, new technology in some programs tries to read the images by attempting to find text in these images. They are not very accurate, and sometimes filter out innocent images of products like a box that has words on it. A newer technique, however, is to use an animated GIF image that does not contain clear text in its initial frame, or to contort the shapes of letters in the image (as in CAPTCHA) to avoid detection by OCR tools.

2.2 Blank Spam

Blank spam is spam lacking a payload advertisement. Often the message body is missing altogether, as well as the subject line. Still, it fits the definition of spam because of its nature as bulk and unsolicited email. Blank spam may be originated in different ways, either intentional or unintentionally:

1. Blank spam can have been sent in a directory harvest attack, a form of dictionary attack for gathering valid addresses from an email service provider. Since the goal in such an attack is to use the bounces to separate invalid addresses from the valid ones, spammers may dispense with most elements of the header and the entire message body, and still accomplish their goals.
2. Blank spam may also occur when a spammer forgets or otherwise fails to add the payload when he or she sets up the spam run.
3. Often blank spam headers appear truncated, suggesting that computer glitches may have contributed to this problem—from poorly-written spam software to malfunctioning relay servers, or any problems that may truncate header lines from the message body.
4. Some spam may appear to be blank when in fact it is not. An example of this is the VBS. Davinia. B email worm[8] which propagates through messages that have no subject line and appears blank, when in fact it uses HTML code to download other files

2.3 Backscatter spam

Backscatter is a side-effect of email spam, viruses and worms, where email servers receiving spam and other mail send bounce messages to an innocent party. This occurs because the original message's envelope sender is forged to contain the email address of the victim. A very large proportion of such email is sent with a forged From: header, matching the envelope sender. Since these messages were not solicited by the recipients, are substantially similar to each other, and are delivered in bulk quantities, they qualify as unsolicited bulk email or spam. As such, systems that generate email backscatter can end up being listed on various DNSBLs and be in violation of internet service providers' Terms of Service.

3. THE ALGORITHMS

This section gives a brief overview of the underlying theory and algorithms we consider. We shall discuss the Naïve Bayesian Classifier, Neural Network Classifier, the K-NN Classifier and the Artificial Immune System Classifier.

3.1 Naive Bayes Classifier

The Naive Bayes classifier is a simple statistical algorithm with a long history of providing surprisingly accurate results. It has been used in several spam classification studies [9, 10, 11, 12], and has become somewhat of a benchmark. It gets its name from being based on Bayes' rule of conditional probability, combined with the "naive" assumption that all conditional probabilities are independent [13].

Naive Bayes classifier examines all of the instance vectors from both classes. It calculates the prior class probabilities as the proportion of all instances that are spam (Pr[spam]), and not-spam (Pr[notspam]). Then (assuming binary attributes) it estimates four conditional probabilities for each attribute: Pr[true|spam], Pr[false|spam], Pr[true|notspam], and Pr[false|notspam]. These estimates are calculated based on the proportion of instances of the matching class that have the matching value for that attribute. To classify an instance of unknown class, the "naive" version of Bayes's rule is used to estimate first the probability of the instance belonging to the spam class, and then the probability of it belonging to the not-spam class. Then it normalizes the first to the sum of both to produce a spam confidence score between 0.0 and 1.0. Note that the denominator of Bayes's rule can be omitted because it is cancelled out in the normalization step. In terms of implementation, the numerator tends to get quite small as the number of attributes grows, because so many tiny probabilities are being multiplied with each other. This can become a problem for finite precision floating point numbers. The solution is to convert all probabilities to logs, and perform addition instead of multiplication. Note also that conditional probabilities of zero must be avoided; instead a "Laplace estimator" (a very small probability) is used. It is important to note that using binary attributes in the instance vectors makes this algorithm both simpler and more efficient. Also, given the prevalence of sparse instance vectors in text classification problems like this one, binary attributes offer the opportunity to implement very significant performance optimizations. Figure 6. presents the Naive Bayes training and classification algorithms used.

Naive Bayes Training Algorithm:
priorProbSpam = proportion of training set that is spam
priorProbNotSpam = proportion of training set that is notspam
For each attribute i:
probT rueSpam[i] = prop. of spams with attribute i true
probF alseSpam[i] = prop. of spams with attribute i false
probT rueNotSpam[i] = prop. of not-spams with attribute i true
probF alseNotSpam[i] = prop. of not-spams with attribute i false
Naive Bayes Classification Algorithm:
probSpam = priorProbSpam
probNotSpam = priorProbNotSpam
For each attribute i:
if value of attribute i for message to be classified is true:
probSpam = probSpam × probT rueSpam[i]
probNotSpam = probNotSpam × probT rueNotSpam[i]
else:
probSpam = probSpam × probF alseSpam[i]
probNotSpam = probNotSpam × probF alseNotSpam[i]
spamminess = probSpam/(probSpam + probNotSpam)

3.2 Artificial Neural Networks

An artificial neural network (ANN), usually called neural network (NN), is a mathematical model or computational model that is inspired by the structure and/or functional aspects of biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Modern neural networks are non-linear statistical data modeling tools. They are usually used to model complex relationships between inputs and outputs or to find patterns in data. By definition, a “neural network” is a collection of interconnected nodes or neurons. See figure 7. The best-known example of one is the human brain, the most complex and sophisticated neural network. Thanks to this cranial-based neural network, we are able to make very rapid and reliable decisions in fractions of a second. [13].

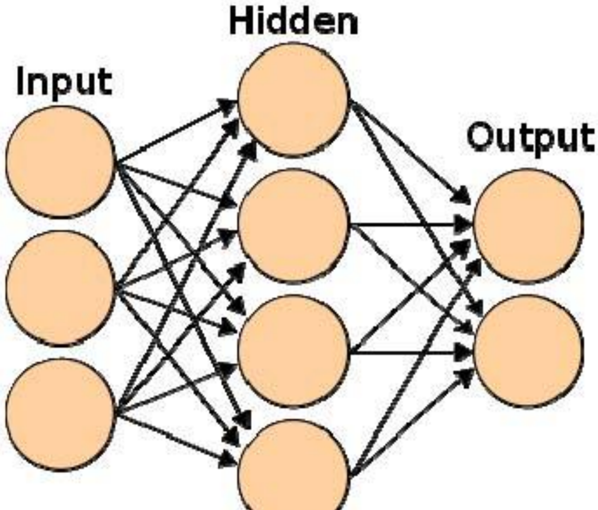

Figure 7: an artificial neural network is an interconnected group of nodes, akin to the vast network of neurons in the human brain.

Spam presents a unique challenge for traditional filtering technologies: both in terms of the sheer number of messages (millions of messages daily) and in the breadth of content (from pornographic to products and services, to finance). Add to that the fact that today’s economic fabric depends on email communication – which is equally broad and plentiful and whose subject matter contextually overlaps with that of many spam messages – and you’ve got a serious challenge. How it works - Since a neural network is based on pattern recognition, the underlying premise is that each message can be quantified according to a pattern. This is represented below in Figure 8. Each plot on the graph (also known as a "vector") represents an email message. Although this 2-D example is an over-simplification, it helps to visualize the principle used behind neural networks.

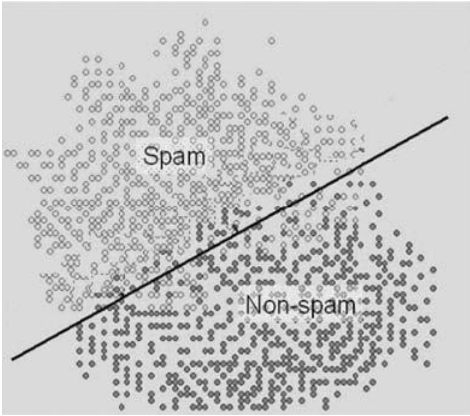

Figure 8: Distinctive patterns of good and spam messages cluster into relatively distinct groups.

To identify these patterns, the neural network must first be "trained". This training involves a computational analysis of message content using large representative samples of both spam and non-spam messages. Essentially the network will "learn" to recognize what we humans mean by "spam" and "non-spam". To aid in this process, we first need to have a clear, concise definition of "spam": Spam, an email sent in bulk where there is no direct agreement in place between the recipient and the sender to receive email solicitation.

U.B.E. (Unsolicited Bulk Email) is another acronym for spam that effectively encapsulates this definition. To create training sets of spam and non-spam emails, each email is carefully reviewed according to this simple, yet restrictive definition of spam. Although the average user often considers all unwanted emails as "spam", emails that border on "solicited" (it was likely requested at some point by the user) should be rejected outright. Examples of these might include email sent from easily recognizable domains, such as Amazon.com or Yahoo.com. A good motto to follow here is: "when in doubt, throw it out". Similarly, non-spam email should be restricted to personal email communications between individuals or groups, and avoid any forms of bulk mailings, regardless of whether they were solicited or not. Once these sets have been gathered and approved, the neural network is ready for training. The ANN system now preprocesses each email in the respective training sets to determine exactly which of these relevant words are found in each spam email, and which of these words are found in the non-spam email. Next, the ANN is trained to recognize certain combinations or patterns of interesting or relevant words to identify spam, or if it sees other combinations, to identify non-spam. The artificial neural network uses a set of sophisticated mathematical equations to perform this type of computation. As some spam and non-spam messages will often "share" characteristics, a clear distinction cannot always be made. By the "non-spam" plots or vectors that find themselves in the "spam" cluster and vice versa. In this "grey area" lies the potential for false positives. After the training is complete, the ANN can now be used to scan live-stream email. Each message is scanned to identify relevant words, which are then processed by the ANN. If the ANN again sees certain types of combinations of word usage indicating a probability of spam, it will report spam, along with a probability value. Following the example in Figure 9, if the vector or plot computed for the message landed above the dividing line, it would be considered "spam". Its probability or confidence level would depend on the relative distance away from the line.

To maximize detections while avoiding false positives, a well-designed heuristics engine will accommodate different sensitivity thresholds, or levels of aggressiveness, in identifying spam. What this means is that the cut-off or dividing point between spam and non-spam can be adjusted so that the likelihood of a false positive match will be greatly reduced. This can be seen in Figure 9 below.

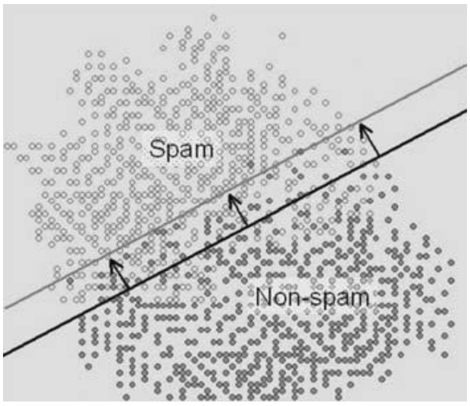

Figure 9: The sensitivity threshold can be adjusted to avoid the "grey" area.

In other words, the further away from the central dividing line between ham and spam email clusters, the lower the chance of false positive detections. Note in Figure 9 that there are far fewer non-spam vectors or patterns above the new cut-off or dividing line.

3.3 K-Nearest Neighbor Classifier

The k-nearest neighbor (K-NN) classifier is considered an example-based classifier, that means that the training documents are used for comparison rather than an explicit category representation, such as the category profiles used by other classifiers. As such, there is no real training phase. When a new document needs to be categorized, the k most similar documents (neighbors) are found and if a large enough proportion of them have been assigned to a certain category, the new document is also assigned to this category, otherwise not. Additionally, finding the nearest neighbors can be quickened using traditional indexing methods. To decide whether a message is legitimate or not, we look at the class of the messages that are closest to it. The comparison between the vectors is a real time process. This is the idea of the k nearest neighbor algorithm:

Stage1. Training

Store the training messages.

Stage2. Filtering

Given a message x, determine its k nearest Neighbors among the messages in the training set. If there are more spam's among these neighbors, classify given message as spam. Otherwise classify it as legitimate mail.

We should note that the use of an indexing method in order to reduce the time of comparisons induces an update of the sample with a complexity O(m), where m is the sample size. As all of the training examples are stored in memory, this technique is also referred to as a memory-based classifier [24]. Another problem of the presented algorithm is that there seems to be no parameter that we could tune to reduce the number of false positives. This problem is easily solved by changing the classification rule to the following l/k rule: If l or more messages among the k nearest neighbors of x are spam, classify x as spam, otherwise classify it as legitimate mail. The k nearest neighbor rule has found wide use in general classification tasks. It is also one of the few universally consistent classification rules.

3.4 Artificial Immune System Classifier Method

Biological immune System has been successful at protecting the human body against a vast variety of foreign pathogens. A role of the immune system is to protect our bodies from infectious agents such as viruses, bacteria, fungi and other parasites. On the surface of these agents are antigens that allow the identification of the invading agents (i.e., pathogens) by the immune cells and molecules, thus provoking an immune response Recognition in the immune system is performed by lymphocytes. Each lymphocyte expresses receptor molecules of one particular shape on its surface (called antibody). An elaborate genetic mechanism involving combinatorial association of a number of gene segments underlies the construction of these receptors. The overall immune response involves three evolutionary methods: gene library evolution generating effective antibodies, negative selection eliminating inappropriate antibodies and clonal selection cloning well performing antibodies (see Figure 10).

In gene library evolution, antibodies recognize antigens by the complementary properties that belong only to antigens, not self-cells. Thus, some knowledge of antigen properties is required to generate competent antibodies. Because of this evolutionary self-organization process, in spam management the gene libraries act as archives of information on how to detect commonly observed antigens. An important constraint that the immune has to satisfy is not to attack self-cells. Negative selection eliminates inappropriate and immature antibodies which bind to self. Clonal selection clones antibodies performing well. In contrast, antibodies performing badly die off after a given lifetime. Thus, according to currently existing antigens, only the fittest antibodies survive. Similarly, instead of having the predefined information about specific antigens, it organizes the fittest antibodies by interacting with the current antigens. The above description is used in the following algorithm [14].

Artificial Immune System algorithm (an email message m)

```
For (each term t in the message) do {
If (there exists a detector p, based on base
String r, matches with t) then {
If (m is spam) then {
Increase r's spam score by s-rate;
} else {
Increase r's ham score by ns-rate;
}
} else {
If (m is spam) then {
If (detector p recognizes t and edmf (p, t) >
threshold) then {
The differing characters are added to its
corresponding entry in the library of
character generalization rules;
} else {
A new base string t is added into the
library of base strings;
}
}
}
Decrease the age of every base string by a-rate;
}
```

Figure 10: Artificial Immune System algorithm (an email message m).

4. CONCLUSION

Spam is becoming a very serious problem to the Internet community, threatening both the integrity of the networks and the productivity of the users. In this paper, we surveyed four machine learning methods for spam filtering. In this paper we discussed the problem of spam and gave an overview of learning based spam filtering techniques. We can say that the field of anti-spam protection is by now mature and well-developed. Then a question arises, why our inboxes are still often full of spam? Reactivity of spammers plays a role surely, and so does the complex nature of spam data. But one more issue not to be underestimated here is that we usually do not protect against spam in all the available ways. In other words, one point which should always be remembered by server administrators and end users is that the anti-spam technologies should be not only designed and developed, but also deployed and used.

REFERENCES

[1] Aladdin Knowledge Systems, "Anti-spam white paper", www.csisoft.com/security/aladdin/esafe_antispam_whitepaper.pdf Retrieved December 28, 2011.

[2] F.Smadja, H.Tumblin, "Automatic spam detection as a text classification task", in: Proc. of Workshop on Operational Text Classification Systems, 2002.

[3] A. Hassanien, H. Al-Qaheri,"Machine Learning in Spam Management",IEEE TRANS.,VOL.X, NO. X, February 26, 2009.

[4] P. Cunningham, N. Nowlan,"A Case-Based Approach to Spam Filtering that Can Track Concept Drift", [Online] Available:https://www.cs.tcd.ie/publications/techreports/ reports.03/TCD-CS-2003-16.pdf Retrieved December 28, 2011 IJCSNS International Journal of Computer Science and Network Security, VOL.12 No.2, 2012.

[5] F. Roli , G.Fumera, "The emerging role of visual pattern recognition in spam filtering: challenge and opportunity for IAPR researchers" http://www.iapr.org/members/ newsletter/Newsletter07-02/index_files/Page465.htm Retrived, December 28, 2011.

[6] H. West, "Getting it Wrong: Corporate America Spams the Afterlife". Clueless Mailers, January 19, 2008.

[7] B.Parizo, "Image spam paints a troubling picture" Search Security, July 26, 2006.

[8] Symantec(2012) VBS. Davinia.B, [Online] Available: http://www.symantec.com/security_response/writeup.jsp?do cid=2001-020713-3220-99 Retrieved, December 28, 2011.

[9] I. Androutsopoulos, J. Koutsias,"An evaluation of naïve bayesian anti-spam filtering". In Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning (ECML 2000), pages 9–17, Barcelona, Spain, 2000.

[10] I. Androutsopoulos, G. Paliouras, "Learning to filter spam E-mail: A comparison of a naïve bayesian and a memorybased approach". In Proceedings of the Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2000), pages 1–13, Lyon, France, 2000.

[11] J. Hidalgo, "Evaluating cost-sensitive unsolicited bulk email categorization". In Proceedings of SAC-02, 17th ACM Symposium on Applied Computing, pages 615–620, Madrid, ES, 2002.

[12] K. Schneider, "A comparison of event models for naïve bayes anti-spam e-mail filtering". In Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics, 2003.

[13] I. Witten, E. Frank, "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations". Morgan Kaufmann, 2000.

[14] C. Miller,"Neural Network-based Antispam Heuristics", Symantec Enterprise Security (2011), www.symantec. com Retrieved, December 28, 2011.

[15] C. Wu, "Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks", Expert Syst., 2009.

