CHAPTER 2
REVIEW OF LITERATURE

Filtering is the most popular and effective way of blocking spam mails from mail boxes (Siponen and Stucke, 2006). The majority of research effort is expended in controlling spam is based on message text analysis. However, recently, in the past few years, spammers have changed the content of spam mails from text message to images, where the intended text is embedded in the image. Thus the idea of the research is forced to move from text-based anti-spam techniques towards images based anti-spam techniques. Spam and the spam blockers designed to combat it have spawned an upsurge in creativity and innovation. Many software developers are developing new and every more effective spam filtering software. All the methods have a common dream that is to eliminate 100% of the spam, which is still not a reality. To reduce the gap between this reality and dream, researchers have proposed many different types of spam filters and this chapter provides a review of them.

2.1. BACKGROUND

The present research work focus on spam classification using neural networks. This chapter provides a background study of both spam and neural networks.

2.1.1. History of spam

Spam mail has become a relatively big problem only in recent years, but it has been around for a long time. The discussion on history of spam explains the evolution of spam over different periods.

- In early years spam letters were addressed and sent manually
- Later spammers used machines, which dramatically raised the amount of spam
The final part is when machine learning appeared at spam filtering and made the filtering substantially effective.

In early years, letters or post cards (popularly known as snail mails) were used in spam where each and every letter has to be addressed and sent manually. The manual process was slow and required time, so spamming was kept at minimum and posed no dangerous situations. In 1978, Gary Thuerk, the first spammer, manually typed in all the email addresses of his several hundred recipients into his email program to send out a single copy of the marketing promotion. He did not randomize the text or attempt to hide his identity. This could be considered as the first bulk mail spam, which was highly tedious and time consuming.

Later, with the advent of fast processing computers and with a spread of the Internet, the potential customer audience had enlarged to millions and then quickly to tens of millions and their email clients and address books. Taking advantage of this situation, the spammers wrote custom email sending software during mid-90's to run on powerful rented servers with fast bandwidth connections to reach internet user worldwide. Access to powerful servers and lots of bandwidth to use to send out millions of spam each day started to require substantial monthly investments on the part of the spammers. At this stage, a spam was considered as a hindrance, but not as a security issue.

In 1994, for the first time in history, a programmer was hired to write a spammer program. The very first famous spammer was Jeff Slaton, the “Spam King”. Apart from being the first known man with a capability of sending millions of emails at the same time, he also had dared to offer a commercial removal option for the users from all spamming lists for just five dollars. This was the starting point for spamming to become a business and from this point, the amount of spam letters received grew in an uncontrolled fashion.

The first known email list was offered for sale in 1995 with more than 2 million addresses. After the first weak attempts in 1997, the first real spam-
filter software was made. Nevertheless, at this time, spam had turned to be totally out of control situation.

This increase, in turn made security vendors and ISP providers to create a blacklist containing IPs of servers that spammers had employed to send out spam mails along with some programs to identify common text used by spam messages. The early spam filters were rule-based software. All the new tricks of the spammers indicated to settle up new rules into the filters. Unfortunately, the spammers were able to try the filters with their messages, so they had the ability to change the letter to get through the filters.

The blacklist technique was so successful, that the returns on the investments in servers and bandwidth were diminishing with each day as the IPs of these servers had been discovered and blacklisted, dramatically reducing deliverability. Often this also resulted in termination of the contracts spammers had with their service providers and the potential discovery of the identities of the spammers. It was at this point, the some enterprising spammers first realized that they can instead utilize potentially millions of machines around the world at almost no cost to them with near absolute anonymity. Some had partnered with virus and exploit writers to get access to the machines, known as zombies, that were being infected by their malware and install specialized SOCKS and SMTP proxy software on those machines to relay connections from the servers of the spammer through the zombies. This ensured that they had access to potentially millions of zombie IP addresses.

Identifying these in real-time proved to be a new challenge for the anti-spam industry and tracking spammers from that point to bring them to justice via the criminal or civil legal systems would become an extremely difficult international undertaking. The zombie machines are often connected to the Internet via an always-on broadband connection and, in aggregate, their combined bandwidth can far exceed the bandwidth of dedicated servers spammers tended to use in the past. To further increase, the speed and volume
of their mailings, they have mostly abandoned the proxy and approach and
today use fully automated and specialized mail server software running on
millions of zombie computers worldwide to send out billions of emails each
day.

The big breakthrough was in 2002, when Paul Graham published “A
Plan For Spam”. This article introduced the usage of machine learning and
statistical classification techniques on the field of spam filtering with Bayesian
networks. Graham (2010) wrote: “I think it's possible to stop spam, and that
content-based filters are the way to do it. The Achilles heel of the spammers is
their message. They can circumvent any other barrier you set up. They have so
far, at least. But they have to deliver their message, whatever it is. If we can
write software that recognizes their messages, there is no way to get around
that”. With the use of Bayesian networks, filtering not only became
significantly more effective, but first time the filter learned the email
characteristics of the users. It was the first solution, which made it impossible
for the spammers to test the filters before sending their emails, hence every
users’ filter worked with different knowledge base.

From this time spammers could only hope that they found the right
method to pass the Bayesian filters, but they could not be sure they could reach
a defined number of users. Even if the spammers figured out a lot of new
tricks, like obfuscating or good-word attacks, the existing filters still had a
good chance to filter them out.

However, with increased effectiveness of text-based anti spam
classification engines, most notably Bayesian filters, in 2006, spammers once
again raised the stakes and deployed new software on the zombies to convert
the message template into an image attachment. By moving their message to an
image, spammers try to avoid exposing usable tokens (words) to such textual
classifiers. In addition, due to image compression, even slight changes in an
image, such as the introduction of pixel noise or randomization of the color
palette, can have great ramifications on its binary representation. This makes it infeasible to block images based on simple binary string signatures.

Several filters have been proposed and implemented to meet the requirements during each period and some of them are discussed in the subsequent sections.

2.1.2. Neural Networks

Background studies on neural network started during 19th and early 20th centuries with the first implementation in 1943 (McCulloch and Pitts, 1943). Their researches considered the concept of artificial neurons, which have the capability to compute arithmetic or logical functions.

In the late 1950s, Frank Rosenblatt proposed the perceptron network and the associated learning rule. In 1957, Rosenblatt published the first major research project in neural computing which included the development of the perceptron element. The perceptron is a pattern classification system, which could identify both abstract and geometric patterns. In addition, the perceptron can make limited generalizations and can properly categorize patterns despite noise in the input. This study showed the first practical application of neural network by demonstrating how neural networks can perform pattern recognition.

In 1959, Bernard Widrow and Tedd Hoff proposed the Adaline (Adaptive Linear Element), based on simple neuron-like elements and used it to train adaptive linear networks. The Adaline and the two-layer Madaline version were used for a variety of applications including speech recognition, character recognition, weather prediction, and adaptive control. Widraw used the adaptive linear element algorithm to develop adaptive filters that eliminate phone line echoes, in the first real life neural network application.

In the mid-1960s, Marvin Minsky and Seymour Papert, considered the neural network potential limitations. They showed that these already known
networks could handle linearly separable problems only and were usually not appropriate for real life applications. As a result, neural network research faded for a while.

In the 1970s, Kohonen, Grossberg and Anderson proposed the Kohonen Network and the Self-organizing Network. Kohonen introduced the concept of the competitive learning rule in which PEs compete to respond to an input stimulus and the winner adapts itself to respond more strongly to that stimulus. This type constitutes an unsupervised learning process and the internal organization of the network is governed only by input stimuli. Grossberg’s contribution was a wealth of research towards the design and construction of neural model as he used neurological data to build neural computing models. Anderson developed a linear model, called a linear associator, which is based on models of memory storage, retrieval and recognition. In addition, Anderson improved the model by combining it with a nonlinear post-processing algorithm, which is used to clean up spurious responses. This model is called Brain-State-in-a-Box.

In the 1980’s, neural network became popular again with the back-propagation algorithm for training multilayer perceptron networks. The concept of back-propagation algorithm was presented by several researchers, such as David Parker, Yaun LeCun, David Rumelhart, James McClelland, and Geoffrey Hinton. While a perceptron network is only capable of solving linear problems, back-propagation network can solve more complex nonlinear problems. This significant capability made the back-propagation networks the most widely used networks.

In 1982, John Hopfield presented a paper describing his neural computing system called the “crossbar associative network” or known as the “Hopfield Model”. This model represented a neuron operation as a thresholding operation and illustrated memory as information stored in the interconnections between neuron units. He also illustrated and modeled the
brain’s ability to call up responses from many locations in response to a stimulus. Thus, this model represents how a neural network associates information from many storage sites for a given input.

In the 1980s, the Bi-directional Associative Memory (BAM) network, Boltzman Machine, the General Regression Neural Network, and the Learning Vector Quantization Network were developed, in addition to the backpropagation and Hopfield models.

Although the concept of neural network has been around for about 50 to 60 years, most applications have appeared in the last fifteen years and the field is still developing very rapidly. Neural networks can be found in many fields ranging from aerospace to medicine, banking and robotics. Given the work done and range of applications, neural network will most likely be a permanent fixture not only as a solution to everyday problems but also as a tool to be used in appropriate situations. It is certain that the more the structure of the brain is understood, the more advances there will be in neural network.

2.2. SPAM FILTERING

Spam filtering is an area of research that has attracted many researchers and academicians. The various proposals reported can be grouped as general techniques, techniques on spam characteristics, spam detection in transmission protocols, solutions based on local changes in email transmission process, language based filters, non-content based spam filters, collaborative spam filters and hybrid methods.

2.2.1. General Techniques

Spam-filtering techniques can be divided into three broad categories (Kolcz et al. 2006), based on:

- Sender reputation
- Email-header analysis
- Analysis of message content
In a sender-based reputation framework, senders are classified as “spammers” or “good senders.” This decision can be based on criteria such as the sender (Golbeck and Hendler, 2004), the sender domain (Taylor, 2006), or the IP address (DCC, 2006). The email-header spam filtering is based on detecting forgery in the email header and distinguishing it from malformatting and other legitimate explanations such as those resulting from forwarding activity (Ludeman and Libbey, 2006). The third category, analysis of message content, has been of particular interest to the machine learning community (Cormack and Bratko, 2006; Yih et al., 2006; Dredze et al., 2007) where approaches such as Naïve Bayes (Sahami et al., 1998; Metsis et al., 2006; Hovold, 2005; Meyer and Whateley, 2004), support vector machines (SVMs) (Drucker et al., 1999; Kolcz and Alspector, 2001; Rios and Zha, 2004), or boosting (He and Thiesson, 2007) are applied to the classification of email.

In applications of machine learning to spam detection, both batch-mode and online update models (Cormack and Bratko, 2006) have been considered. Training a classifier in batch mode allows choosing from a wider range of algorithms and optimizing performance over a large quantity of training data. Conversely, unless the classifier is frequently retrained, the system may quickly fall pray to adversarial attacks (Dalvi et al., 2004). Online learning approaches, on the other hand, allow for immediate incorporation of user feedback into the filtering function, but tend to be more difficult to tune, and the number of efficient algorithms is limited.

Naïve Bayes tends to be most popular (Metsis et al., 2006), although online variants of logistic regression (Goodman and Yih, 2006) and SVMs (Sculley and Wachman, 2007) have been proposed. In either approach, changes to the classification function may require a significant number of new examples. This has prominent effect when the amount of data used to derive the current models is already very large. However, sometimes large quantities of highly similar spam (i.e., a campaign) are sent within a relatively short period of time. The diversity of messages within a campaign may be too low to
effectively adjust the decision function quickly enough. In such situation, it is beneficial to consider spam filters that can track and eliminate high-volume spam campaigns.

One of the problems in automating spam classification is the lack of a consensus definition for spam (Taylor, 2006; Prakash and O’Donnell, 2005). What some people consider spam may be considered solicited mail by others. Some email-service providers allow users to mark emails they consider spam and report them to their ESP (Email Service Provider). Sometimes, users can also report opposite errors, i.e., when legitimate email is mistakenly classified as spam. However, the cost of a large ESP’s incorporating each individual’s judgments into the filtering system may outweigh the benefits (Kolcz et al., 2006). Nevertheless, spam reports provided by users, as well as other forms of data acquisition, such as honeypot accounts (Symantec, 2004; Prince et al., 2005), have been used to build and validate spam detection systems.

Of particular interest is the use of such data to track spam campaigns sent in volume over some periods of time, with a campaign assumed to consist of highly similar and often near-duplicate messages. In this context, when many users report nearly identical emails as spam, one can reasonably label the nature of a campaign by the volume of reports received. A key requirement to the success of such a scheme is the ability to identify emails belonging to the same campaign, despite small or irrelevant differences (some tactically inserted by the spammer to complicate detection). The problem can be otherwise described as near-duplicate message detection, which has received considerable attention in the field of Information Retrieval (Chowdhury et al., 2002; Broder, 1997; Henzinger, 2006). In the email domain, near or exact replica based spam detection (Hall, 1999; Kolcz et al., 2004) has not been studied as extensively as other content-based methodologies. Nevertheless, such techniques have been applied in practical systems (e.g., Distributed Checksum Clearinghouse – DCC (2006) and Vipul’s Razor (Prakash and O’Donnell, 2005).
Signature-based duplication is a form of clustering. In that context it has been suggested that the stream of all incoming emails is clustered to identify high-density spikes in the content distribution, which are likely to correspond to email campaigns (Yoshida et al., 2004). That requires an ESP to have a dedicated setup in the path of message delivery and is not easy to experiment with in a research environment. Another use of clustering is to verify cluster membership. In that context, once a cluster signature becomes known (e.g., via user reports), it is easy to determine whether an arbitrary message falls into the same cluster.

2.2.2. Techniques based on General Characteristics of spam

This section reviews papers, concentrating on researching general characteristics of a spam.

There is a growing scientific literature addressing the characteristics of the spam phenomenon. In general, spam is used to advertise different kinds of goods and services. The percentage of advertisements dedicated to a particular kind of goods or services changes over time (Hulten et al., 2004). The changeability of spam was addressed by Delany et al. (2004), who point out in particular the local nature on the concept drift in spam. Quite often spam serves the needs of online frauds. Drake et al. (2004) conducted a study on a special case of spamming activity, phishing. A study of malicious spam content spreading viruses was reported and studied by Lugaresi (2004). Spam attacks that upset the work of a mail server was studied and reported by Nagamalai et al. (2007).

Characteristics of spam traffic are different from those of legitimate mail traffic, in particular legitimate mail is concentrated on diurnal periods, while spam arrival rate is stable over time. This behaviour of spam mail was reported by Gomes et al. (2005). In the same period, spammers harvesting activities, which can be used to recognize and identify spammers was reported by Prince et al. (2005).
A very important fact is that spammers are reactive, namely they actively oppose every successful anti-spam effort (Fawcett, 2003), so that performance of a new method usually decreases after its deployment. Pu and Webb (2006) analyze the evolution of spamming techniques. They showed that spam constructing methods become extinct if filters are effective to cope with them or if other successful efforts are taken against them. A study of the network-level behavior of spammers by Ramachandran and Feamster (2006) showed that the majority of spam comes from a few concentrated parts of IP address space. Moreover, they also found that only a small subset of sophisticated spammers uses temporary route announcements in order to remain untraceable.

2.2.3. Modifying Email Transmission Protocols for spam detection

One of the proposed ways of stopping spam is to enhance or even substitute the existing standards of email transmission by new, spam-proof variants. The main drawback of the commonly used Simple Mail Transfer Protocol (SMTP) is that it provides no reliable mechanism of checking the identity of the message source. Overcoming this disadvantage, namely providing better ways of sender identification, is the common goal of Sender Policy Framework (SPF, formerly interpreted as Sender Permitted From) (SPF FAQ, 2010), Designated Mailers Protocol (DMP) (Fecyk, 2003), Trusted Email Open Standard (TEOS) (Schiavone et al., 2003), and SenderID (sometimes also spelled Sender ID) (SenderID, 2004). A comparison and discussion of this kind of proposals are given by Levine and DeKok (2004). SenderID, being released in 2004, has grown quite popular already.

According to Goodman et al. (2007), almost 40% of legitimate email is today SenderID-compliant. The principle of its work resulted in the following: the owner of a domain publishes the list of authorized outbound mail servers, thus allowing recipients to check whether a message which pretends to come from this domain really originates from there. A discussion of the problem of
fake IP addresses in email messages and ways of overcoming it by changes in
standards is given by Goodman (2004).

The idea underlying another group of proposals to amend the existing
protocols is to add a step to the mail sending process that

(i) represents a minor obstacle for sending few emails and
(ii) a major one for sending great number of messages.

Efforts in this direction were made already in 1992 (Dwork and Naor, 1992),
when it was proposed to ask sender to compute a moderately hard function
before granting him the permission to sent a message. Another proposal
(Seltzer, 2003) was to establish a small payment for sending an email message,
negligible for a common user, but big enough to prevent a spammer to broadcast
millions of messages. An interesting version of this approach is Zmail protocol
(Kuipers et al., 2005). In Zmail, a small fee is paid by a sender to the receiver.
Thus, a common user who sends and receives messages gets neither damage
nor profit from using email, while spamming becomes a costly operation.

Another approach is to use simple tests that allow the system to
distinguish human senders from robots (CAPTCHA, 2005), for example to ask
the user to answer a moderately easy question before sending the message. One
disadvantage of this approach is that such protection is annoying to human
senders. Duan et al. (2005) propose to use a differentiated email delivery
architecture to handle messages from different classes of senders in different
ways. For example, some messages are kept on the sender’s mail server until
the receiver asks to transmit them to him.

2.2.4. Local Changes in Email Transmission Process

Some solutions do not require global protocol changes but propose to
manage email in a different way locally. Li et al. (2004) and Saito (2005)
propose slowing down the operations with messages that are likely to be spam.
A similar idea is discussed in the technical report by Twining et al. (2004), who
propose to use the past behaviour of senders for fast prediction of message category. The spam mails are then maintained in a lower priority queue, while the ham mails in a higher priority queue. In this way, the delivery of legitimate mail is guaranteed, but it becomes hard to broadcast many spam messages at once.

Yamai et al. (2005) point out that when a spammer falsifies the sender identity in the messages, the server corresponding to the falsified address receives a great number of error mails. Yamai and collaborators propose to solve this problem by using a separate mail transfer agent for the error messages. Goodman and Rounthwaite (2004) point to the possibility of controlling not only ingoing, but also outgoing spam, stopping it on the level of email service provider used by a spammer.

### 2.2.5. Language-based filters

Another group of methods use the fact that the message body is a text in a natural language. These methods can be applied to message headers or whole messages. The main motivation for their application on spam filtering relies on the fact that they are effective in natural language text classification. In fact, the same motivation can also be applied to the methods based on compression models. Examples of such models include dynamic Markov compression and prediction by partial matching. They were successfully used with the data extracted from both bodies and headers of the messages (Bratko et al., 2006).

Chi by degrees of freedom method, previously used for document authorship identification, is proposed for spam filtering by O’Brien and Vogel (2003). Messages are represented in terms of character or word Ngrams. The idea of the method is to compare the new message to the spam and legitimate messages in the training data using the chi-by-degrees-of freedom (CDF) test. The CDF is calculated by dividing the value of the $\chi^2$ test by the number of degrees of freedom.
Smoothed N-gram language models, proposed by Medlock (2006), used smoothed higher order N-gram models. N-gram language models are based on the assumption that the existence of a certain word at a certain position in a sequence depends only of the previous N-1 words.

2.2.6. Filters based on non-content features

The methods based on structured analysis of the header and of meta-level features, such as number of attachments, use specific technical aspects of email and so they are specific to spam filtering.

Leiba *et al.* (2005) proposed a method called analyzing SMTP path to detect spam. The filtering method was based on analyzing IP addresses in the reverse-path and ascribing reputation to them according to amount of spam and legitimate mail delivered through them. Both this and the subsequent method can be viewed as development of the idea of blacklisting and whitelisting.

Analyzing the user’s social network is another algorithm proposed by Boykin and Roychowdhury (2005). They analyzed the ‘From’, ‘To’, ‘Cc’ and ‘Bcc’ fields of the message headers in order to build a graph of social relations of the user, and then uses this graph in order to classify new messages. The idea of extracting the user’s social network from his mailbox was further developed by Chirita *et al.* (2005) and by Golbeck and Hendler (2004).

Behavior-based filtering rests on extracting knowledge about the behavior behind a given message or group of messages from their non-content features. Later detect spam by comparing it to the predefined or extracted knowledge about the typical behaviors of malicious and normal users. Examples are the works of Yeh *et al.* (2005), and Hershkop (2006). Yeh *et al.* (2005) use well-known behaviors of spammers, such as using incorrect dates. Hershkop (2006) proposes a number of behavior models based on non-content features, which can be used to detect spam and viruses as anomalies in the
email flow. Examples of such models include recipient frequency and histograms of user’s past activity.

2.2.7. Collaborative spam filtering

Certain efforts are made to achieve better spam filtering through the collaboration of users. The usual way of such collaboration is sharing the knowledge about spam between P2P users (Lazzari et al., 2005; Zhou et al., 2003), or gathering spam reports from the users on a mail server (like in Google’s Gmail2). In such situation of data exchange between users the issue of privacy arises. Damiani et al. (2004) propose a privacy-preserving approach to P2P spam filtering system. In particular, spam reports in their system are sent without indicating the user who is the source of the report.

Mo et al. (2006) propose a multi-agent system for collaborative spam filtering. In this system, each message is first classified as spam, legitimate mail or suspicious mail by a local agent. A collaborative judgement is required for suspicious messages only. Usually the users are proposed to exchange opinions or information about emails,

Garg et al. (2006) propose to exchange trained filters instead, thus significantly reducing the amount of data transmitted. Another interesting effort for collaborative spam fighting is Project HoneyPot (2010), intended to identify email address harvesters with the help of specially generated email addresses.

2.2.8. Hybrid approaches

Studies were also conducted, which analyze the possibility of combining different algorithms for spam filtering. Most of the research implements this approach if they use unrelated features to produce a solution (Leiba et al., 2005; Zhang et al., 2004). Existing technologies and algorithms focus on individual parameters of the malicious content. However, efficiency of the filtering techniques gets significantly reduced when special forging techniques are used and the shortcomings of individual algorithms are exploited. The
hybrid technique can be implemented by using various models, considering available resources with the server. He et al. (2008) proposed a framework which combines white/black listing and challenge-response methods. Bhuleskar et al. (2009), after identifying the advantages and disadvantages of various filters, combines the advantages of the various filtering techniques and proposes a hybrid filter.

In an enterprise environment, the commercial hybrid system Brightmail (Brightmail, 2010) enjoys a good reputation. Companies rely on Brightmail, since it offers a good performance and the necessary professional support to keep the system up to date. Since Brightmail does not offer a home user solution, the freeware Spampal (2004) is better suited for individual users. There are several different, free filters available on the internet which can be included in Spampal, and the user has a free choice which of them to use. Unfortunately, Spampal is only available for Microsoft Windows operating systems. Linux users can work with Spamassassin (2004), which is also evolving into a full featured hybrid tool.

Hybrid solutions need to be carefully designed as the combination might increase time complexity while increasing security and accuracy.

2.3. COMMERCIAL ANTI-SPAM SOFTWARE SOLUTIONS

Spam filtering is not only a subject of scientific research, but also a wide and well-established field of software development. Available commercial and non-commercial solutions combine different techniques of message filtering. Moreover, they use protocol extensions and are sometimes integrated into single software solutions with anti-virus protection. An overview of some methods is given in Table 2.1. The website addresses of these solutions are provided in Table 2.2. Tables 2.1 and 2.2 are reproduced from Bryl (2008). The meanings of the column titles are as follows:
• Whitelists/blacklists: use of various personal and public blacklists and whitelists

• Managing replies: using additional mechanisms to ensure that replies to the user's messages are not classified as spam

• Using decoy accounts: collecting spam messages on decoy accounts for future extraction of fingerprints or rules

• Protocol extensions: support of protocol extensions intended to prevent falsifying the sender's identity or to ensure that a message is legitimate by asking the sender for confirmation

• Anti-virus/anti-spyware: integrating an anti-virus and/or antispyware solution into the same product

• User collaboration: support of sharing data about spam among the users of the product

• Message analysis: methods of filtering more sophisticated than blacklisting and whitelisting

• Bayesian: Bayesian algorithm is used for message analysis, probably in combination with other techniques

• Image analysis: use of algorithms of analysis of graphical content

• Downloading updates: the product regularly downloads updates for its database from a server

• Price: the price of the product as given on the official site, as of May, 2010
TABLE 2.1
METHODS USED IN SOFTWARE ANTI-SPAM SOLUTIONS

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Symantec Mail Security for SMTP</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>Not stated on the site</td>
<td></td>
</tr>
<tr>
<td>MailCleaner</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>Complex sys. of prices</td>
<td></td>
</tr>
</tbody>
</table>

Solutions suitable both for client and server side

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SpamAssassin</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Free</td>
<td></td>
</tr>
<tr>
<td>Bogofilter</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Free</td>
<td></td>
</tr>
</tbody>
</table>

Client-side software solutions

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CA Anti-Spam</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
<td>6.39, 95</td>
</tr>
<tr>
<td>Vanquish vqeME</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td>834.95/year</td>
</tr>
<tr>
<td>Cloudmark Desktop</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td>8.39, 95</td>
</tr>
<tr>
<td>Altumis SpamCatcher</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td>829, 99</td>
</tr>
<tr>
<td>MailWasher Pro</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td>83, 7</td>
</tr>
<tr>
<td>POPFile</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td>Free</td>
</tr>
<tr>
<td>Spamihilator</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td>Free</td>
</tr>
<tr>
<td>SpamPali</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td>Free</td>
</tr>
<tr>
<td>Ko</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td>Free</td>
</tr>
<tr>
<td>G-Lock SpamCombat</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
<td>Free</td>
</tr>
</tbody>
</table>

Software solutions supplied with a hardware base

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BorderWare Email Security Gateway</td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td>Not stated on the site</td>
<td></td>
</tr>
<tr>
<td>Barracuda Spam Firewall</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td>Complex sys. of prices</td>
<td></td>
</tr>
</tbody>
</table>

The table is based only on the explicit statements on the official websites of the products, and thus may be incomplete. It does not provide real performance comparison and is not intended to advice any choice between this products, but rather to show which techniques are used in practical solutions.
<table>
<thead>
<tr>
<th>S.No</th>
<th>Product</th>
<th>Website Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Symantec Mail Security for SMTP</td>
<td><a href="http://www.symantec.com/enterprise/products/overview.jsp?pvid=845_1">http://www.symantec.com/enterprise/products/overview.jsp?pvid=845_1</a></td>
</tr>
<tr>
<td>2</td>
<td>MailCleaner</td>
<td><a href="http://www.mailcleaner.net">http://www.mailcleaner.net</a></td>
</tr>
<tr>
<td>3</td>
<td>SpamAssassin</td>
<td><a href="http://spamassassin.apache.org">http://spamassassin.apache.org</a></td>
</tr>
<tr>
<td>4</td>
<td>Bagofilter</td>
<td><a href="http://bogofilter.sourceforge.net">http://bogofilter.sourceforge.net</a></td>
</tr>
<tr>
<td>5</td>
<td>CA Anti-Spam</td>
<td><a href="http://home3.ca.com/STContent/landingpages/Products/Antispam/ASPM001/index.aspx">http://home3.ca.com/STContent/landingpages/Products/Antispam/ASPM001/index.aspx</a></td>
</tr>
<tr>
<td>6</td>
<td>Vanquish vqME</td>
<td><a href="https://www.vqme.com">https://www.vqme.com</a></td>
</tr>
<tr>
<td>7</td>
<td>Cloudmark Destop</td>
<td><a href="http://cloudmark.com/desktop">http://cloudmark.com/desktop</a></td>
</tr>
<tr>
<td>8</td>
<td>Allume SpamCatcher</td>
<td><a href="http://www.allume.com/win/spamcatcher">http://www.allume.com/win/spamcatcher</a></td>
</tr>
<tr>
<td>9</td>
<td>MailWasher Pro</td>
<td><a href="http://www.mailwasher.net">http://www.mailwasher.net</a></td>
</tr>
<tr>
<td>10</td>
<td>POPFile</td>
<td><a href="http://popfile.sourceforge.net">http://popfile.sourceforge.net</a></td>
</tr>
<tr>
<td>11</td>
<td>Spamihilator</td>
<td><a href="http://www.spamihilator.com">http://www.spamihilator.com</a></td>
</tr>
<tr>
<td>12</td>
<td>SpamPal</td>
<td><a href="http://www.spampal.org">http://www.spampal.org</a></td>
</tr>
<tr>
<td>13</td>
<td>K9</td>
<td><a href="http://keir.net/k9.html">http://keir.net/k9.html</a></td>
</tr>
<tr>
<td>15</td>
<td>BorderWare EmailSecurity Gateway</td>
<td><a href="http://www.borderware.com/products/email-security-gateway">http://www.borderware.com/products/email-security-gateway</a></td>
</tr>
</tbody>
</table>

From the table, it can be seen that existing solutions often combine various ways of blacklisting and whitelisting with more complex filtering methods. Many products use Bayesian filtering, because approaches based on Naive Bayes, though shown by many studies to be slightly outperformed by other techniques, have the advantage of being very fast and fit for continuous online training.
2.4. EVALUATION METHODS

The great number and variety of spam filtering methods results in the need for evaluation and comparison of them. The usual way of testing a filter is applying it to a corpus of previously gathered mail messages sorted into spam and legitimate mail. The most simple measure used to express the results of such testing is filtering accuracy, namely percentage of messages classified correctly (Lai and Tsai, 2004), which has the disadvantage of making no difference between false positives and false negatives. More informative measures are spam/ham recall and spam/ham precision. Androutsopoulos et al. (2000) propose to use the relational cost $\lambda$ of the two types of errors as a variable parameter, and introduce several new measures based on it: weighted accuracy, weighted error rate, and a Total Cost Ratio (TCR). TCR is the relative cost of using the filter (and so having some false positives and some false negatives) to using no filter at all (and so having all the spam misclassified, but all the legitimate mail classified correctly). Table 2.3 gives the formulae of the measures named above.

In the table $n_{L\rightarrow L}$ and $n_{S\rightarrow S}$ are the number of ham and spam messages classified correctly. Similarly, $n_{L\rightarrow S}$ and $n_{S\rightarrow L}$ are the numbers of ham and spam messages misclassified and $\lambda$ is the relative cost of the two type of errors.

2.5. IMAGE SPAM DETECTION

As already mentioned, Image-Based Spam is a mail which contains unwanted content inside an embedded graphic file (typically appears in GIF, JPG, PNG, BMP etc.), making it difficult for spam filters to identify. These unsolicited emails contain no relevant text or hyperlinks and the message appear to be a text message, but however, is an image of text. Image-Based Spam has rapidly spread in recent years and is now recognized as a major global issue.
## TABLE 2.3

MEASURES OF FILTERING PERFORMANCE

<table>
<thead>
<tr>
<th>Measure</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>$\frac{n_{L\rightarrow L} + n_{S\rightarrow S}}{n_{S\rightarrow S} + n_{S\rightarrow L} + n_{L\rightarrow S} + n_{L\rightarrow L}}$</td>
</tr>
<tr>
<td>Error rate</td>
<td>$\frac{n_{L\rightarrow S} + n_{S\rightarrow L}}{n_{S\rightarrow S} + n_{S\rightarrow L} + n_{L\rightarrow S} + n_{L\rightarrow L}}$</td>
</tr>
<tr>
<td>False positive rate</td>
<td>$\frac{n_{L\rightarrow S}}{n_{L\rightarrow S} + n_{L\rightarrow L}}$</td>
</tr>
<tr>
<td>Spam recall</td>
<td>$\frac{n_{S\rightarrow S}}{n_{S\rightarrow S} + n_{S\rightarrow L}}$</td>
</tr>
<tr>
<td>Spam precision</td>
<td>$\frac{n_{S\rightarrow S}}{n_{S\rightarrow S} + n_{L\rightarrow S}}$</td>
</tr>
<tr>
<td>Ham recall</td>
<td>$\frac{n_{L\rightarrow L}}{n_{L\rightarrow S} + n_{L\rightarrow L}}$</td>
</tr>
<tr>
<td>Ham precision</td>
<td>$\frac{n_{L\rightarrow L}}{n_{L\rightarrow L} + n_{S\rightarrow L}}$</td>
</tr>
<tr>
<td>Weighted accuracy</td>
<td>$\frac{\lambda n_{L\rightarrow L} + n_{S\rightarrow S}}{\lambda (n_{L\rightarrow L} + n_{L\rightarrow S}) + n_{S\rightarrow S} + n_{L\rightarrow L}}$</td>
</tr>
<tr>
<td>Weighted error rate</td>
<td>$\frac{\lambda n_{L\rightarrow S} + n_{S\rightarrow L}}{\lambda (n_{L\rightarrow L} + n_{L\rightarrow S}) + n_{S\rightarrow L} + n_{S\rightarrow S}}$</td>
</tr>
<tr>
<td>Total Cost Ratio</td>
<td>$\frac{n_{S\rightarrow S} + n_{S\rightarrow S}}{\lambda n_{L\rightarrow S} + n_{S\rightarrow L}}$</td>
</tr>
<tr>
<td>False Detection Rate (FDR)</td>
<td>$\frac{n_{L\rightarrow S}}{n_{L\rightarrow S} + n_{S\rightarrow S}}$</td>
</tr>
<tr>
<td>False Positive Rate (FPR)</td>
<td>$\frac{n_{L\rightarrow S}}{n_{L\rightarrow S} + n_{L\rightarrow L}}$</td>
</tr>
<tr>
<td>ROC</td>
<td>True positive rate plotted against false positive rate</td>
</tr>
</tbody>
</table>

Source: Androutsopoulos et al. (2008a, 2008b)
2.5.1. Filtering techniques

The majority of effort expended in controlling spam is based on message text analysis. However, there has been research addressing the problem of image spam detection. These attempts are primarily based on trying to detect text inside the images.

Learning Fast Classifiers for image spam is one such attempt. They assert that their method exceeds 90% accuracy. They classify spam and ham images focusing on simple properties of the image and also a just-in-time (JIT) feature extraction. JIT is added to the feature classification dataset as needed by the classifier. The features of images used for basic classification are file format, file size, image metadata, and image size. Other image features they considered are average color, color saturation, edge detection, prevalent color coverage and random pixel test. Their feature set evaluation was accomplished using a number of learning models – Maximum Entropy, naïve Bayes, and ID3 decision tree. All three algorithms were implemented in Mallet, using the Java 1.5 imageio library for image processing and feature extraction. For “feature prediction”, they computed mutual information for each feature with the target label spam or ham (Dredze et al., 2007).

Barreno et al. (2006) explains the different types of attacks on the machine learning algorithms and the systems. A variety of defenses against those attacks and the ideas that are important to secure the machine learning are also discussed. This approach illustrates the methods that spammers handle to attack a system to design an image spam. The issue of machine learning security goes beyond intrusion detection systems and spam email filters. The different measures of defenses involved in their discussion are robustness, detecting the attacks, disinformation, randomization for targeted attack, and cost of countermeasures.

A modification of Latent Dirichlet allocation (LDA), Known as multi-corpus LDA technique was introduced by Biro et al. (2008). In their proposal,
they created a bag-of-words document for every web site and run LDA both on the corpus of sites labeled as spam and as non-spam. This assisted them to collect spam and non-spam topics during their training phase. They implemented these collections on an unseen test site to detect the spam messages. This method in combination with web spam challenge 2008 public features, and the connectivity sonar features is used to test images. Using logistic regression to aggregate these classifiers, the multi-corpus LDA yields an improvement of around 11 per cent in F-measures and 1.5 per cent in ROC.

Spam web page detection through content analysis is put forth by Ntoulas et al. (2006). It projected some earlier undefined techniques for automatic spam message detection. They also discussed the effectiveness of those techniques in isolation and when aggregated using some classification algorithms, which proved to be truth worthy in detecting the image spam.

Clustering based spam detection is put forth by Wei et al. (2009) which propagates a fuzzy-matching algorithm to group subjects found spam emails, which are generated by malware. The subjects similar to each other are found out using a dynamic programming. The main proposal is that the recursive seed selection strategy allows the algorithm to detect similar patterns even the spammer creates a variation of the original pattern. This proved to be an effective approach in detecting and grouping spam emails using templates. Clustering algorithm is utilized to find the similarity of strings, similarity of spam subjects and for clustering spam subjects.

Youn and McLeod (2009) described the method of filtering gray email using personalized ontologies. They proposed a personalized ontology spam filter to make decisions for gray email. Gray email is a message that could reasonably be estimated as either spam or ham. A user profile has been created for each user or a class of users to handle gray email. This profile ontology creates a blacklist of contacts and topic words.
Filtering Image Spam with Near-Duplicate Detection is another attempt at filtering spam images. In their research, they use a “content-based image similarity searching” technique to classify spam images. A false positive rate of 0.001% is maintained through their detection system (Wang et al., 2005).

Finally, research has been conducted to identify a number of useful visual features including banner images, computer generated graphics and embedded-text. These features are then combined with message text features. They also use an SVM to train their classifier to distinguish between ham and spam. This approach achieved about an 81% detection rate with about a 1% false positive rate with their corpus (Wu, 2005; Wu et al., 2006). ANNs have been used by AISK (2010) to identify spam by looking at the text-based header portion of spam email. PUREmail is a second generation email filter uses artificial Intelligence to process images by visualization (PUREmail, 2010). Bowling et al. (2008) is another work which used neural network to different spam and ham image mails.

Although the results given are encouraging, to further increase the accuracy, this research work uses an artificial neural network to classify spam and ham. Artificial Neural Networks, BPNN, have the capability to mimic human intelligence when trained with good input data and is used in the detection of image spam.

2.5.2. Image Spam Classification Based On Content

One popular practice when creating spam pages is “Keyword Stuffing”. In Content based image spam detection we investigate whether an excessive number of words within a web page (excluding markup) is a good indicator spam. In the next step, have to determine whether there is excessive appearance of keywords in the title of a page. Uncommon practice that was observed in manually tagged data set is the use of “composite” words in spam pages. Content-based Naive Bayes (PGRAM) is another technique for the classification of Image spam. Graham (2010) found out that the task of spam
detection has floated the idea of a partial Naïve Bayes approach, biased towards low false positive rates. It also uses word tokens, but filters out predefined common tokens.

The content and the header of the incoming email are mostly analyzed by the available anti-spam techniques (Donelli, 2006). They try to infer something about the kind of the material contained in the message by looking for specific pattern typical of a spam message. For these reasons, these filters are known as “content based.” There are many anti-spam techniques available that falls under this category.

Blacklist and White list filters check whether the incoming message is from a known and trusted email address. Rule based filters correlate a score to every incoming email calculated according to a set of rules based on typical features of spam messages (fake SMTP components, Keywords, HTML formatting, etc) (Cohen, 1996). In case the score exceeds the given threshold value it is recognized to be a spam message. Major problem in this method is that, since its semantics are not well defined, it is difficult to aggregate rules and ascertains a threshold that limits the number of false positives. Spam Assassin (Apache Group, 2005), results from the successful implementation of the above-mentioned technique.

2.5.3. Spam identification using image features

The Connected Component based (CC) and the texture-based approaches are the two leading approaches that were used in the past to extract the characteristics for the text detection task. These characteristics include coherence in space, geometry and color (Zhu et al., 2006). In CC-based methods (Loncaric, 1988), the image is segmented into a set of CCs and is grouped into potential text regions based on their geometric relation. These potential regions are then examined using some rule-based heuristics. The heuristics makes use of the characteristics like size, the aspect ratio and the orientation of the region. The efficiency of this method becomes questionable
when the text is multi-colored, textured, with a small font size, or overlapping with other graphical objects. In texture-based methods (Kim et al., 2003), it is assumed that the texts have distinct textural properties, and this can be used to distinguish them from the background. Even though this method performs well for images with noisy, degraded, or complex texts and/or background, it seems to be time-consuming as texture analysis is essentially computational intensive.

In recent years, after the spammer’s shift to use images instead of text, image features were given more importance during spam detection. In image-based filtering, the main issue is to find features both relevant and easy to extract, while the classification itself can be further performed by state-of-the-art algorithms.

The fully-functional optical character recognition (OCR) procedure is computationally expensive, so usually simplified models are proposed to recognize spam in images. In particular, Aradhye et al. (2005) extract five features from the images, namely the fraction of the image occupied by regions identified as text, and color saturation and color heterogeneity calculated separately for text and non-text regions.

A similar approach to feature extraction for image-based filtering was proposed by Wu et al. (2005). In addition to detecting the size and the number of embedded text regions without actual text recognition, they characterize a banner as a special kind of image (very narrow in width or height, and with a large aspect ratio), and use the number of banner-like images as an additional feature.

After this, another effort of Dredze et al. (2007) introduced a new approach, which relies only on features that take very small time to extract. This had the advance of avoiding not only OCR, but in general any computations more complicated than simple edge detection. Thus, the features used in this work are selected among those that do not require image analysis at all (for example, file format, height and width of the image, or file size), and
those that are retrieved through very simple analysis of images (for example, average color or color saturation).

Nhung et al. (2007) used simple edge based features combines with similarity score between an image and a set of templates, which was fed as input to SVM, for image spam mails.

Mehta et al. (2008) used visual features and SVMs to classify spam and ham images and showed more than 98% accuracy, which was an improvement of at least six per cent compared to existing solutions. They also proposed another method which used Gaussian mixture Models (GMM) based on Agglomerative Information Bottleneck (AIB) principle using Jensen-Shannon divergence as the distance measure to identify image-based spam mails.

Recently, Zuo et al. (2009) used local invariant features of images on a one-class SVM classifier using pyramid match kernel as kernel function for detecting image spam.

2.5.4. Intensity as image feature for spam detection

Intensity of an image is used in spam analysis and detection in many different ways. In most of the works, image intensity was used as a way to identify edges that separate text from background. These methods after identifying text region normally perform Optical Character Recognition (OCR) techniques, to recognize the text, which are then used for spam detection.

In the work proposed by Aradhye et al. (2005), a method that separates spam images from other common categories of email images based on extracted overlay text and color features was used. Image intensities combined with thresholding were used to detect edges of text during text region extraction in an image spam. Successful detection of edges paved way to successful spam detection.
According to Hsia and Chen (2009) a spam image consists of embedded images which are composed of complex or simple backgrounds overlaid with text. The main idea of the text-region extraction method is that text often contrasts a lot with background. A region with massive intensity changes (i.e., edges) would be potentially a text region. This fact indicates the wide use of intensity feature in image spam identification.

Chen and Zhang (2009) proposed a method to detect spam by tracking the source of the spam distributors based on image spam clustering. The principle behind their work is that two spam mails are said to be from the same source, if they have similar composition, that is, similar content, layout and editing style. For this purpose, they segmented the image into text, background and foreground region. They worked on the assumption that each of these regions has different intensity distribution and using the intensity distribution detail for successful segmentation.

A similar methodology was also used by Zhang et al. (2009), who used color features, layout features, text features and texture features combined to trace spam source and thus filter them. Zuo et al. (2009) combined image intensity and Fourier-Mellin invariant features to detect spam images using Support Vector Machine (SVM).

2.5.5. Histogram based Image Spam Detection

He et al. (2008) proposed a new method of filtering image spam, which is called FH (File properties and Histogram) algorithm. FH algorithm utilizes file properties and histogram (gray histogram or color histogram), and is a fast method because of not extracting text and analyzing the content of email. The same authors in 2009 (He et al., 2009a, 2009b) proposed a simple method for filtering image spam, which utilizes file properties and histogram (gray histogram or color histogram).
Klangpraphant and Bhattarakosol (2010) proposed a method that can detect a set of image spam email called a Partial Image Spam Inspector (PIMSI). The method focuses on spam that consists of both texts and images over the email system. The method extensively used histogram analysis of the image and extracted features for spam email. The significant feature of this technique is that it will sensitively protect the distribution of all messages which have a partial similarity of spam e-mail.

2.6. CONCLUSION

In this chapter, different solutions to the problem of spam were presented. From the study, it can be concluded that the most popular and well-developed approach to anti-spam is learning-based filtering. The current state of the art includes lots of filters based on various classification techniques applied to different parts of email messages. Since research in image spam is recent, with less than four years since the emergence of the problem, more work is needed to improve the existing identification of spam mails. Moreover, the application of neural network to the task of image spam identification is sparse and requires more careful analysis. This research uses BPNN as for training the proposed spam filters and uses mainly the image content features for this purpose. Seven new filters which are enhanced versions of the existing filters are proposed and from them hybrid varieties are also developed. The steps used during filtering are feature extraction, dimensionality reduction and spam classification. The various methods and techniques used are given in Chapter 3, Methodology.