# Tightening the net: a review of current and next generation spam filtering tools

James Carpinter & Ray Hunt[*]
Department of Computer Science and Software Engineering
University of Canterbury

## Abstract

This paper provides an overview of current and potential future spam filtering approaches. We examine the problems spam introduces, what spam is and how we can measure it. The paper primarily focuses on automated, non-interactive filters, with a broad review ranging from commercial implementations to ideas confined to current research papers. Both machine learning and non-machine learning based filters are reviewed as potential solutions and a taxonomy of known approaches presented. While a range of different techniques have and continue to be evaluated in academic research, heuristic and Bayesian filtering dominate commercial filtering systems; therefore, a case study of these techniques is presented to demonstrate and evaluate the effectiveness of these popular techniques.

**Keywords:** spam, ham, heuristics, machine learning, non-machine learning, Bayesian filtering, blacklisting.

## 1 Introduction

The first message recognised as spam was sent to the users of Arpanet in 1978 and represented little more than an annoyance. Today, email is a fundamental tool for business communication and modern life, and spam represents a serious threat to user productivity and IT infrastructure worldwide. While it is difficult to quantify the level of spam currently sent, many reports suggest it represents substantially more than half of all email sent and predict further growth for the foreseeable future [18, 43, 30].

For some, spam represents a minor irritant; for others, a major threat to productivity. According to a recent study by Stanford University [36], the average Internet user loses ten working days each year dealing with incoming spam. Costs beyond those incurred sorting legitimate email from spam are also present: 15% of all email contains some type of virus payload, and one in 3,418 emails contained pornographic images particularly harmful to minors [54]. It is difficult to estimate the ultimate dollar cost of such expenses; however, most estimates place the worldwide cost of spam in 2005, in terms of lost productivity and IT infrastructure investment, to be well over US$10 billion [29, 52].

The magnitude of the problem has introduced a new dimension to the use of email: the spam filter. Such systems can be expensive to deploy and maintain, placing a further strain on IT budgets. While the reduced flow of spam email into a user's inbox is generally welcomed, the existence of false positives often necessitates the user manually double-checking filtered messages; this reality somewhat counteracts the assistance the filter delivers. The effectiveness of spam filters to improve user productivity is ultimately limited by the extent to which users must manually

[*] email: ray.hunt@canterbury.ac.nz

review filtered messages for false positives.

Unfortunately, the underlying business model of bulk emailers (spammers) is simply too attractive. Commissions to spammers of 25–50% on products sold are not unusual [30]. On a collection of 200 million email addresses, a response rate of 0.001% would yield a spammer a return of $25,000, given a $50 product. Any solution to this problem must reduce the profitability of the underlying business model; by either substantially reducing the number of emails reaching valid recipients, or increasing the expenses faced by the spammer.

Regrettably, no solution has yet been found to this vexing problem. The classification task is complex and constantly changing. Constructing a single model to classify the broad range of spam types is difficult; this task is made near impossible with the realisation that spam types are constantly moving and evolving. Furthermore, most users find false positives unacceptable. The active evolution of spam can be partially attributed to changing tastes and trends in the marketplace; however, spammers often actively tailor their messages to avoid detection, adding a further impediment to accurate detection.

The similarities between junk postal mail and spam can be immediately recognised; however, the nature of the Internet has allowed spam to grow uncontrollably. Spam can be sent with no cost to the sender: the economic realities that regulate junk postal mail do not apply to the internet. Furthermore, the legal remedies that can be taken against spammers are limited: it is not difficult to avoid leaving a trace, and spammers easily operate outside the jurisdiction of those countries with anti-spam legislation.

The remainder of this section provides supporting material on the topic of spam. Section 2 provides an overview of spam classification techniques. Sections 3.1 and 3.2 provide a more detailed discussion of some of the spam filtering techniques known: given the rapidly evolving nature of this field, it should be considered a snapshot of the critical areas of current research. Section 4 details the evaluation of spam filters, including a case study of the PreciseMail Anti-Spam system operating at the University of Canterbury. Section 5 finishes the paper with some conclusions on the state of this research area.

## 1.1 Definition

Spam is briefly defined by the TREC 2005 Spam Track as “unsolicited, unwanted email that was sent indiscriminately, directly or indirectly, by a sender having no current relationship with the recipient” [12]. The key elements of this definition are expanded on in a more extensive definition provided by Mail Abuse Prevention Systems [35], which specifies three requirements for a message to be classified as spam. Firstly, the message must be equally applicable to many other potential recipients (i.e. the identity of the recipient and the context of the message is irrelevant). Secondly, the recipient has not granted ‘deliberated, explicit and still-revocable permission for it to be sent’. Finally, the communication of the message gives a ‘disproportionate benefit’ to the sender, as solely determined by the recipient. Critically, they note that simple personalisation does not make the identity of the sender relevant and that failure by the user to explicitly opt-out during a registration process does not form consent.

Both these definitions identify the predominant characteristic of spam email: that a user receives unsolicited email that has been sent without any concern for their identity.

## 1.2 Solution strategies

Proposed solutions to spam can be separated into three broad categories: legislation, protocol change and filtering.

A number of governments have enacted legislation prohibiting the sending of spam email, including the USA (Can Spam Act 2004) and the EU (directive 2002/58/EC). American legislation requires an ‘opt-out’ list that bulk mailers are required to provide; this is arguably less effective than the European (and Australian) approach of requiring explicit 'opt-in' requests from consumers wanting to receive such emails. At present, legislation has appeared to have little effect on spam volumes, with some arguing that the law has contributed to an increase in spam by giving bulk advertisers permission to send spam, as long as certain rules were followed.

Many proposals to change the way in which we send email have been put forward, including the required authentication of all senders, a per email charge and a method of encapsulating policy within the email address [28]. Such proposals, while often providing a near complete solution, generally fail to gain support given the scope of a major upgrade or replacement of existing email protocols.

Interactive filters, often referred to as 'challenge-response' (C/R) systems, intercept incoming emails from unknown senders or those suspected of being spam. These messages are held by the recipient's email server, which issues a simple challenge to the sender to establish that the email came from a human sender rather than a bulk mailer. The underlying belief is that spammers will be uninterested in completing the 'challenge' given the huge volume of messages they sent; furthermore, if a fake email address is used by the sender, they will not receive the challenge. Selective C/R systems issue a challenge only when the (non-interactive) spam filter is unable to determine the class of a message. Challenge-response systems do slow down the delivery of messages, and many people refuse to use the system[1].

Non-interactive filters classify emails without human interaction (such as that seen in C/R systems). Such filters often permit user interaction with the filter to customise user-specific options or to correct filter misclassifications; however, no human element is required during the initial classification decision. Such systems represent the most common solution to resolving the spam problem, precisely because of their capacity to execute their task without supervision and without requiring a fundamental change in underlying email protocols.

$$SR = \frac{\#\text{ spam correctly classified}}{\text{Total }\#\text{ of spam messages}}$$

$$SP = \frac{\#\text{ spam correctly classified}}{\text{Total }\#\text{ of messages classified as spam}}$$

$$F_1 = \frac{2 \times SP \times SR}{SP + SR}$$

$$A = \frac{\#\text{ email correctly classified}}{\text{Total }\#\text{ of emails}}$$

Figure 1: Common experimental measures for the evaluation of spam filters.

### 1.3 Statistical evaluation

Common experimental measures include spam recall (SR), spam precision (SP), $F_1$ and accuracy (A) (see figure 1 for formal definitions of these measures). Spam recall is effectively spam accuracy. A legitimate email classified as spam is considered to be a 'false positive'; conversely, a spam message classified as legitimate is considered to be a 'false negative'.

The accuracy measure, while often quoted by product vendors, is generally not useful when evaluating anti-spam solutions. The level of misclassifications $(1 - A)$ consists of both false positives and false negatives; clearly a 99% accuracy rate with 1% false negatives (and no false positives) is preferable to the same level of accuracy with 1% false positives (and no false negatives). The level of false positives and false negatives is of more interest than total system accuracy. Furthermore, accuracy can be severely distorted by

[1] A cynical consideration of this approach may conclude that the recipient considers their time is of more value that the sender's.

the composition of the corpus; clearly, if the false positive and negative rates are different, overall accuracy will largely be determined by the ratio of legitimate email to spam.

A clear trade-off exists between false positives and false negatives statistics: reducing false positives often means letting more spam through the filter. Therefore, the reported levels of either statistic will be significantly affected by the classification threshold employed during the evaluation. False positives are regarded as having a greater cost than false negatives; cost sensitive evaluation can be used to reflect this difference. This imbalance is reflected in the $\lambda$ term: misclassification of a legitimate email as spam is considered to be $\lambda$ times as costly as misclassifying a spam email as legitimate. $\lambda$ values of 1, 9 and 999 are often used [47, 26] to represent the cost differential between false positives and false negatives; however, no evidence exists [26] to support the assumption that a false positive is 9 or 999 times more costly as a false negative. The value of $\lambda$ is difficult to quantify, as it depends largely on the likelihood of a user noticing a misclassification and on the importance of the email in question. The definition and measurement of this cost imbalance ($\lambda$) is an open research problem.

The recall measure (see figure 1) defines the number of relevant documents identified as a percentage of all relevant documents; this measures a spam filter's ability to accurately identify spam (as $1 - SR$ is the false negative rate). The precision measure defines the number of relevant documents identified as a percentage of all documents identified; this shows the noise that filter presents to the user (i.e. how many of the messages classified as spam will actually be spam). A trade-off, similar to that between false positives and negatives, exists between recall and precision. $F_1$ is the harmonic mean of the recall and precision measures and combines both into a single measure.

As an alternative measure, Hidalgo [26] suggests ROC curves (Receiver Operating Characteristics). The curve shows the trade off between true positives and false positives as the classification threshold parameter within the filter is varied. If the curve corresponding to one filter is uniformly above that corresponding to another, it is reasonable to infer that its performance exceeds that of the other for any combination of evaluation weights and external factors [10]; the performance differential can be quantified using the area under the ROC curves. The area represents the probability that a randomly selected spam message will receive a higher 'score' than a randomly selected legitimate email message, where the 'score' is an indication of the likelihood that the message is spam.

## 2 Overview

Filter classification strategies can be broadly separated into two categories: those based on machine learning (ML) principles and those not based on ML (see figure 2). Traditional filter techniques, such as heuristics, blacklisting and signatures, have been complemented in recent years with new, ML-based technologies. In the last 3–4 years, a substantial academic research effort has taken place to evaluate new ML-based approaches to filtering spam; however, this work is ongoing.

ML filtering techniques can be further categorised (see figure 2) into complete and complementary solutions. Complementary solutions are designed to work as a component of a larger filtering system, offering support to the primary filter (whether it be ML or non-ML based). Complete solutions aim to construct a comprehensive knowledge base that allows them to classify all incoming messages independently. Such complete solutions come in a variety of flavours: some aim to build a unified model, some compare incoming email to previous examples (previous likeness), while others use a collaborative approach, combining multiple classifiers to evaluate email (en-

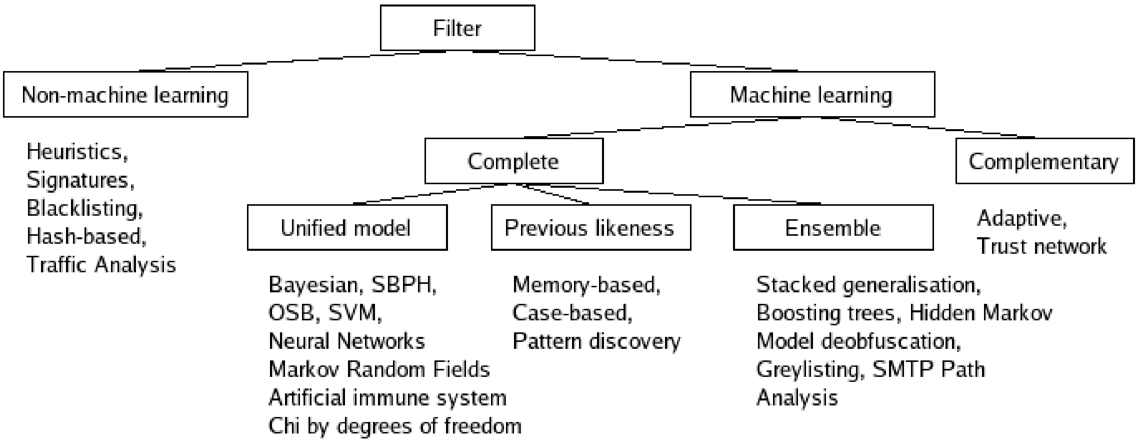

Figure 2: Classification of the various approaches to spam filtering detailed in section 2.

semble).

Filtering solutions operate at one of two levels: at the mail server or as part of the user's mail program. Server-level filters examine the complete incoming email stream, and filter it based on a universal rule set for all users. Advantages of such an approach include centralised administration and maintenance, limited demands on the end user, and the ability to reject or discard email before it reaches the destination.

User-level filters are based on a user's terminal, filtering incoming email from the network mail server as it arrives. They often form a part of a user's email program. ML-based solutions often work best when placed at the user level [19], as the user is able to correct misclassifications and adjust rule sets.

Spam filtering systems can operated either on-site or off-site. On-site solutions can give local IT administrators greater control and more customisation options, in addition to relieving any security worries about redirecting email off-site for filtering. According to Cain [5], of the META Group, it is likely that on-site solutions are cheaper than their service (off-site) counterparts. He estimates on-premises solutions have a cost of US$6–12 per user (based on one gateway server and 10,000 users), compared to a cost of US$12–24 per user for a similar hosted (off-site) solution. On-site filtering can take place at both the hardware and software level.

Software-based filters comprise many commercial and most open source products, which can operate at either the server or user level. Many software implementations will operate on a variety of hardware and software combinations [49].

Appliance (hardware-based) on-site solutions use a piece of hardware dedicated to email filtering. These are generally quicker to deploy than a similar software-based solution, given that the device is likely to be transparent to network traffic [37]. The appliance is likely to contain optimised hardware for spam filtering, leading to potentially better performance than a general-purpose machine running a software-based solution. Furthermore, general-purpose platforms, and in particular their operating systems, may have inherent security vulnerabilities: appliances may have pre-hardened operating systems [8].

Off-site solutions (service) are based on the subscribing organisation redirecting their MX records[2] to the off-site vendor, who then filters the incoming email stream, before redi-

[2] Mail exchange records are found in a domain name database and specify the email server used for handling emails addressed to that domain.

recting the email back to the subscriber [41]. Theoretically, spam email will never enter the subscriber's network. Given that the organisation's email traffic will flow through external data centres, this raises some security issues: some vendors will only process incoming email in memory, while others will store to disk [5]. Initial setup of an off-site filter option is substantially quicker: it can be operational within a week, while similar software solutions can take IT staff between 4–8 weeks to install, tune and test [5]. Off-site solutions require only a supervisory presence from local IT staff and no upfront hardware or software investments in exchange for a monthly fee.

# 3 Filter technologies

## 3.1 Non-machine learning

### 3.1.1 Heuristics

Heuristic, or rule-based, analysis uses regular expression rules to detect phrases or characteristics that are common to spam; the quantity and seriousness of the spam features identified will suggest the appropriate classification for the message. The historical and current popularity of this technology has largely been driven by its simplicity, speed and consistent accuracy. Furthermore, it is superior to many advanced filtering technologies in the sense that it does not require a training period.

However, in light of new filtering technologies, it has several drawbacks. It is based on a static rule set: the system cannot adapt the filter to identify emerging spam characteristics. This requires the administrator to construct new detection heuristics or regularly download new generic rule files. The rule set used by a particular product will be well known: it will be largely identical across all installation sites. Therefore, if a spammer can craft a message to penetrate the filter of a particular vendor, their messages will pass unhindered to all mail servers using that particular filter. Open source heuristic filters, provide both the filter and the rule set for download, allowing the spammer to test their message for its penetration ability.

Graham [22] acknowledges the potentially high levels of accuracy achievable by heuristic filters, but believes that as they are tuned to achieve near 100% accuracy, an unacceptable level of false positives will result. This prompted his investigation of Bayesian filtering (see section 3.2.1 and 4.2).

### 3.1.2 Signatures

Signature-based techniques generate a unique hash value (signature) for each known spam message. Signature filters compare the hash value of an incoming email against all stored hash values of previously identified spam emails to classify the email. Signature generation techniques make it statistically improbable for a legitimate email message to have the same hash as a spam message. This allows signature filters to achieve a very low level of false positives.

Cloudmark[3] provides a commercial implementation of a signature filter, integrating with the network mail server and communicating with the Cloudmark server to submit and receive spam signatures. Vipul's Razor[4] is an open source alternative, using a distributed, collaborative mechanism to distribute signatures with appropriate trust safeguards that prohibit the network's penetration by a malicious spammer.

However, signature-based filters are unable to identify spam emails until such time as the email has been reported as spam and its hash distributed. Furthermore, if the signature distribution network is disabled, local filters will be unable to catch newly created spam messages.

Simple signature matching filters are trivial for spammers to work around. By inserting a string of random characters in each spam

[3] http://www.cloudmark.com

[4] http://razor.sourceforge.net

message sent, the hash value of each message will be changed. This has led to new, advanced hashing technique, which can continue to match spam messages that have minor changes aimed at disguising the message.

Spammers do have a window of opportunity to promote their messages before a signature is created and propagated amongst users. Furthermore, for the signature filter to remain efficient, the database of spam hashes has to be properly managed; the most common technique is to remove older hashes [42]. Once the spammer's message hash has been removed from the network, they can resume sending their message.

Yoshida et al. [57] use a combination of hashing and document space density to identify spam. Substrings of length L are extracted from the email, and hash values generated for each. The first N hash values form a vector representation of the email. This allows similar emails to be identified and their frequency recorded; given the high volumes of email spammers are required to send to generate a worthwhile economic benefit, there is a heavy maldistribution of spam email traffic which allows for easy identification. Document space density is therefore used to separate spam from legitimate email, and when this method is combined with a short whitelist for solicited mass email, the authors report results of 98% recall and 100% precision, using over 50 million actual pieces of email traffic.

Damiani et al. [15] use message digests, addresses of the originating mail servers and URLs within the message to identify spam mail. Each message maps to a 256-bit digest, and is considered the same as another message if it differed by at most 74 bits. Previous work [16] has identified that this approach is robust against attempts to disguise the message. This email identification approach is then implemented within a P2P (peer-to-peer) architecture. Similarly, Gray & Haahr [25] present the CASSANDRA architecture for a personalised, collaborative spam filtering system, using a signature-based filtering technology and P2P distribution network.

#### 3.1.3 Blacklisting

Blacklisting is a simplistic technique that is common within nearly all filtering products. Also known as block lists, black lists filter out emails received from a specific sender. Whitelists, or allow lists, perform the opposite function, automatically allowing email from a specific sender. Such lists can be implemented at the user or at the server level, and represent a simple way to resolve minor imperfections created by other filtering techniques, without drastically overhauling the filter.

Given the simplistic nature of technology, it is unsurprising that it can be easily penetrated. The sender's email address within an email can be faked, allowing spammers to easily bypass blacklists by inserting a different (fake) sender address with each bulk mailing. Correspondingly, whitelists can also be targeted by spammers. By predicting likely whitelisted emails (e.g. all internal email addresses, your boss's email address), spammers can penetrate other filtering solutions in place by appropriately forging the sender address.

DNS blacklisting operates on the same principles, but maintains a substantially larger database. When a SMTP session is started with the local mail server, the foreign host's address is compared against a list of networks and/or servers known to allow the distribution of spam. If a match is recorded, the session is immediately closed, preventing the delivery of the spam message. This filtering approach is highly effective at discarding substantial amounts of spam email, yet requires low system requirements to operate, and enabling it often requires only minimal changes to the mail server and filtering solution.

However, such lists often have a notoriously high rate of false positives, making them "dangerous" to use as a standalone filtering system [51]. Once blacklisted, spammers can cheaply acquire new addresses. Often several people must complain before an address is blacklisted; by the time the list is updated and distributed, the spammer can often send millions of spam messages. Spammers can also masquerade as legitimate sites. Their motivation here is twofold: either they will escape being blacklisted or they will cause a legitimate site to be blacklisted (reducing the accuracy, and therefore the attractiveness, of the DNS blacklist) [42].

Several filters now use such lists as part of a complete filtering solution, weighting information provided by the DNS blacklist and incorporating it into results provided by other techniques to produce a final classification decision.

#### 3.1.4 Traffic analysis

While strictly not a spam filtering technology at present, Gomes et al. [21] provide a characterisation of spam traffic patterns. By examining a number of email attributes, they are able to identify characteristics that separate spam traffic from non-spam traffic. Several key workload aspects differentiate spam traffic; including the email arrival process, email size, number of recipients per email, and popularity and temporal locality among recipients. An underlying difference in purpose gives rise to these differences in traffic: legitimate mail is used to interact and socialise, where spam is typically generated by automatic tools to contact many potential, mostly unknown users. They consider their research as the first step towards defining a spam signature for the construction of an advanced spam detection tool.

### 3.2 Machine learning

#### 3.2.1 Unified model filters

Bayesian filtering now commonly forms a key part of many enterprise-scale filtering solutions. No other machine learning or statistical filtering technique has achieved such widespread implementation and therefore represents the 'state-of-the-art' approach in industry.

It addresses many of the shortcomings of heuristic filtering. It uses an unknown (to the sender) rule set: the tokens and their associated probabilities are manipulated according to the user's classification decisions and the types of email received. Therefore each user's filter will classify emails differently, making it impossible for a spammer to craft a message that bypasses a particular brand of filter. The rule set is also adaptive: Bayesian filters can adapt their concepts of legitimate and spam email, based on user feedback, which continually improves filter accuracy and allows detection of new spam types.

Bayesian filters maintain two tables: one of spam tokens and one of 'ham' (legitimate) mail tokens. Associated with each spam token is a probability that the token suggests that the email is spam, and likewise for ham tokens. For example, Graham [22] reports that the word 'sex' indicates a 0.97 probability that an email is spam. Probability values are initially established by training the filter to recognise spam and legitimate email, and are then continually updated (and created) based on email that the filter successfully classifies. Incoming email is tokenised on arrival, and each token is matched with its probability value from the user's records. The probability associated with each token is then combined, using Bayes' Rule, to produce an overall probability that the email is spam. An example is provided in figure 3.

Bayesian filters perform best when they operate on the user level, rather than at the network mail server level. Each user's email and definition of spam differs; therefore a token database populated with user-specific data will result in more accurate filtering [19].

The use of Bayes formula as a tool to identify spam was initially applied to spam filtering in 1998 by Sahami et al. [46] and Pantel & Lin [39]. Graham [22] [23] later implemented a Bayesian filter that caught 99.5% of spam with 0.03% false positives. Androutsopoulos

For example, the following set of keywords were extracted from an unseen email:

`prescription` (0.9) `tomorrow` (0.1) `student` (0.1) `james` (0.01) `quality` (0.85)

A value of 0.9 for prescription indicates 90% of previously seen emails that included that word were ultimately classified as spam, with the remaining 10% classified as legitimate email.

To calculate the overall probability of an email being spam ($P$):

$$\begin{aligned} P &= \frac{x_1 \cdot x_2 \cdots x_n}{x_1 \cdot x_2 \cdots x_n + (1 - x_1) \cdot (1 - x_2) \cdots (1 - x_n)} \\ &= \frac{0.9 \cdot 0.1 \cdot 0.1 \cdot 0.01 \cdot 0.85}{0.9 \cdot 0.1 \cdot 0.1 \cdot 0.01 \cdot 0.85 + (1 - 0.9) \cdot (1 - 0.1) \cdot (1 - 0.1) \cdot (1 - 0.01) \cdot (1 - 0.85)} \\ &= 0.006 \text{ (to three decimal places)} \end{aligned}$$

This value indicates that it is unlikely that the email message is spam; however, the ultimate classification decision would depend on the decision boundary set by the filter.

Figure 3: A simple example of Bayesian filtering.

et al. [2] established that a naive Bayesian filter clearly surpasses keyword-based filtering, even with a very small training corpus. More recently, Zdziarski [58] has introduced Bayesian Noise reduction as a way of increasing the quality of the data provided to a naive Bayes classifier. It removes irrelevant text to provide more accurate classification by identifying patterns of text that are commonplace for the user.

Given the high levels of accuracy that a Bayesian filter can potentially provide, it has unsurprisingly emerged as a standard used to evaluate new filtering technologies. Despite such prominence, few Bayesian commercial filters are fully consistent with Bayes' Rule, creating their own artificial scoring systems rather than relying on the raw probabilities generated [53]. Furthermore, filters generally use 'naive' Bayesian filtering, which assumes that the occurrence of events are independent of each other; i.e. such filters do not consider that the words 'special' and 'offers' are more likely to appear together in spam email than in legitimate email.

In attempt to address this limitation of standard Bayesian filters, Yerazunis et al. [56, 50] introduced sparse binary polynomial hashing (SBPH) and orthogonal sparse bigrams (OSB). SBPH is a generalisation of the naive Bayesian filtering method, with the ability to recognise mutating phrases in addition to individual words or tokens, and uses the Bayesian Chain Rule to combine the individual feature conditional probabilities. Yerazunis et al. reported results that exceed 99.9% accuracy on real-time email without the use of whitelists or blacklists. An acknowledged limitation of SBPH is that the method may be too computationally expensive; OSB generates a smaller feature set than SBPH, decreasing memory requirements and increasing speed. A filter based on OSB, along with the non-probabilistic Winnow algorithm as a replacement for the Bayesian Chain rule, saw accuracy peak at 99.68%, outperforming SBPH by 0.04%; however, OSB used just 600,000 features, substantially less than the 1,600,000 features required by SBPH.

Support vector machines (SVMs) are generated by mapping training data in a nonlinear manner to a higher-dimensional feature space, where a hyperplane is constructed which maximises the margin between the sets. The hyperplane is then used as a nonlinear decision boundary when exposed to real-world data. Drucker et al. [17] applied the technique to spam filtering, testing it against three other text classification algorithms: Ripper, Rocchio and boosting decision trees. Both boosting trees and SVMs provided "acceptable" performance, with SVMs preferable given their lesser training requirements. A SVM-based filter for Microsoft Outlook has also been tested and evaluated [55]. Rios & Zha [45] also experiment with SVMs, along with random forests (RFs) and naive Bayesian filters. They conclude that SVM and RF classifiers are comparable, with the RF classifier more robust at low false positive rates; they both outperform the naive Bayesian classifier.

While chi by degrees of freedom has been used in authorship identification, it was first applied by O'Brien & Vogel [38] to spam filtering. Ludlow [34] concluded that tens of millions of spam emails may be attributable to 150 spammers; therefore authorship identification techniques should identify the textual fingerprints of this small group. This would allow a significant proportion of spam to be effectively filtered. This technique, when compared with a Bayesian filter, was found to provide equally good or better results.

Clark et al. [9] construct a backpropogation trained artificial neural network (ANN) classifier named LINGER. ANNs require relatively substantial amount of time for parameter selection and training, when compared against other previously evaluated methods. The classifier can go beyond the standard spam/legitimate email decision, instead classifying incoming email into an arbitrary number of folders. LINGER outperformed naive Bayesian, $k$-NN, stacking, stumps and boosted trees filtering techniques, based on their reported results, recording perfect results (across many measures) on all tested corpora, for all $\lambda$. LINGER also performed well when feature selection was based on a different corpus to which it was trained and tested.

Chhabra et al. [7] present a spam classifier based on a Markov Random Field (MRF) model. This approach allows the spam classifier to consider the importance of the neighbourhood relationship between words in an email message (MRF cliques). The inter-word dependence of natural language can therefore be incorporated into the classification process; this is normally ignored by naive Bayesian classifiers. Characteristics of incoming emails are decomposed into feature vectors and are weighted in a superincreasing manner, reflective of inter-word dependence. Several weighting schemes are considered, each of which differently evaluates increasingly long matches. Accuracy over 5000 test messages is shown to be superior to that shown by a naive Bayesian-equivalent classifier (97.98% accurate), with accuracy reaching 98.88% with a window size (i.e. maximum phrase length) of five and an exponentially superincreasing weighting model.

### 3.2.2 Previous likeness based filters

Memory-based, or instance-based, machine learning techniques classify incoming email according to their similarity to stored examples (i.e. training emails). Defined email attributes form a multi-dimensional space, where new instances are plotted as points. New instances are then assigned to the majority class of its $k$ closest training instances, using the $k$-Nearest-Neighbour algorithm, which classifies the email. Sakkis et al. [47] [3] use a $k$-NN spam classifier, implemented using the TiMBL memory-based learning software [14]. The basic $k$-NN classifier was extended to weight attributes according to their importance and to weight nearer neighbours with greater importance (distance weighting). The classifier was compared with a naive Bayesian classifier using cost sensitive evaluation. The memory-based classifier compares "favourably" to the naive Bayesian approach, with spam recall improving at all levels (1, 9,

999) of $\lambda$, with a small cost of precision at $\lambda$ = 1, 9. The authors conclude that this is a "promising" approach, with a number of research possibilities to explore.

Case-based reasoning (CBR) systems maintain their knowledge in a collection of previously classified cases, rather than in a set of rules. Incoming email is matched against similar cases in the system's collection, which provide guidance towards the correct classification of the email. The final classification, along with the email itself, then forms part of the system's collection for the classification of future email. Cunningham et al. [13] construct a case-based reasoning classifier that can track concept drift. They propose that the classifier both adds new cases and removes old cases from the system collection, allowing the system to adapt to the drift of characteristics in both spam and legitimate mail. An initial evaluation of their classifier suggests that it outperforms naive Bayesian classification. This is unsurprising given that naive Bayesian filters attempt to learn a "unified spam concept" that will identify all spam email; spam email differs significantly depending on the product or service on offer.

Rigoutsos and Huynh [44] apply the Teiresias pattern discovery algorithm to email classification. Given a large collection of spam email, the algorithm identifies patterns that appear more than twice in the corpus. Negative training occurs by running the pattern identification algorithm over legitimate email; patterns common to both corpora are removed from the spam vocabulary. Successful classification relies on training the system based on a comprehensive and representative collection of spam and legitimate email. Experimental results are based on a training corpus of 88,000 pieces of spam and legitimate email. Spam precision was reported at 96.56%, with a false positive rate of 0.066%.

### 3.2.3 Ensemble filters

Stacked generalisation is a method of combining classifiers, resulting in a classifier ensemble. Incoming email messages are first given to ensemble component classifiers whose individual decisions are combined to determine the class of the message. Improved performance is expected given that different ground-level classifiers generally make uncorrelated errors. Sakkis et al. [48] create an ensemble of two different classifiers: a naive Bayesian classifier ([2] [1]) and a memory-based classifier ([47] [3]). Analysis of the two component classifiers indicated they tend to make uncorrelated errors. Unsurprisingly, the stacked classifier outperforms both of its component classifiers on a variety of measures.

The boosting process combines many moderately accurate weak rules (decision stumps) to induce one accurate, arbitrarily deep, decision tree. Carreras and Marquez [6] use the AdaBoost boosting algorithm and compare its performance against spam classifiers based on decision trees, naive Bayesian and $k$-NN methods. They conclude that their boosting based methods outperform standard decision trees, naive Bayes, $k$-NN and stacking, with their classifier reporting $F_1$ rates above 97% (see section 1.3). The AdaBoost algorithm provides a measure of confidence with its predictions, allowing the classification threshold to be varied to provide a very high precision classifier.

### 3.2.4 Complementary filters

Adaptive spam filtering [40] targets spam by category. It is proposed as an additional spam filtering layer. It divides an email corpus into several categories, each with a representative text. Incoming email is then compared with each category, and a resemblance ratio generated to determine the likely class of the email. When combined with Spamihilator, the adaptive filter caught 60% of the spam that passed through Spamihilator's keyword filter.

Boykin & Roychowdhury [4] identify a user's trusted network of correspondents with an automated graph method to distinguish between legitimate and spam email. The classifier was able to determine the class of 53% of all emails evaluated, with 100% accuracy. The authors intend this filter to be part of a more comprehensive filtering system, with a content-based filter responsible for classifying the remaining messages. Golbeck and Hendler [20] constructed a similar network from 'trust' scores, assigned by users to people they know. Trust ratings can then be inferred about unknown users, if the users are connected via a mutual acquaintance(s).

Content-based email filters work best when words inside the email text are lexically correct; i.e. most will rapidly learn that the word 'viagra' is a strong indicator of spam, but may not draw the same conclusions from the word 'V.i-a.g*r.a'. Assuming the spammer continues to use the obfuscated word, the content-based filter will learn to identify it as spam; however, given the number of possibilities available to disguise a word, most standard filters will be unable to detect these terms in a reasonable amount of time. Lee and Ng [31] use a hidden Markov model in order to deobfuscate text. Their model is robust to many types of obfuscation, including substitutions and insertions of non-alphabetic characters, straightforward misspellings and the addition and removal of unnecessary spaces. When exposed to 60 obfuscated variants of 'viagra', their model successfully deobfuscated 59, and recorded an overall deobfuscation accuracy of 94% (across all test data).

Spammers typically use purpose-built applications to distribute their spam [27]. Greylisting tries to deter spam by rejecting email from unfamiliar IP addresses, by replying with a soft fail (i.e. 4xx). It is built on the premise that the so-called 'spamware' [33] does little or no error recovery, and will not retry to send the message. Any correct client should retry; however, some do not (either due to a bug or policy), so there is the potential to lose legitimate email. Also, legitimate email can be unnecessarily delayed; however, this is mitigated by source IP addresses being automatically whitelisted after they have successfully retried once. An analysis performed by Levine [33] over a seven-week period (covering 715,000 delivery attempts), 20% of attempts were greylisted; of those, only 16% retried. Careful system design can minimise the potential for lost legitimate email; certainly greylisting is an effective technique for rejecting spam generated by poorly implemented spamware.

SMTP Path Analysis [32] learns the reputation of IP addresses and email domains by examining the paths used to transmit known legitimate and spam email. It uses the 'received' line that the SMTP protocol requires that each SMTP relay add to the top of each email processed, which details its identity, the processing timestamp and the source of the message. Despite the fact that these headers can easily be spoofed, when operating in combination with a Bayesian filter, overall accuracy is approximately doubled.

# 4 Evaluation

## 4.1 Barriers to comparison

This paper outlines many new techniques researched to filter spam email. It is difficult to compare the reported results of classifiers presented in various research papers given that each author selects a different corpora of email for evaluation. A standard 'benchmark' corpus, comprised of both spam and legitimate email is required in order to allow meaningful comparison of reported results of new spam filtering techniques against existing systems.

However, this is far from being a straightforward task. Legitimate email is difficult to find: several publicly available repositories of spam exist (e.g. www.spamarchive.org); however, it is significantly more difficult to locate a similarly vast collection of legitimate emails, presumably due to the privacy concerns. Spam is also constantly changing. Techniques used by spammers to communicate their message are continually evolving [27]; this is also seen, to a lesser extent, in legitimate email. Therefore, any static spam corpus would, over time, no longer resemble the makeup of current spam email.

Graham-Cumming [24], maintainer of the Spammers' Compendium, has identified 18 new techniques used by spammers to disguise their messages between 14 July 2003 and 14 January 2005. A total of 45 techniques are currently listed (as of 11 December 2005). While the introduction of modern spam construction techniques will affect a spam filter's ability to detect the actual content of the message, it is important to note that most heuristic filter implementations are updated regularly, both in terms of the rule set and underlying software.

Several alternatives to a standard corpus exist. SpamAssassin (spamassassin.apache.org) maintains a collection of legitimate and spam emails, categorised into easy and hard examples. However, the corpus is now more than two years old. Androutsopoulos et al. [1] have built the 'Ling-Spam' corpus, which imitates legitimate email by using the postings of the moderated 'Linguist' mailing list. The authors acknowledge that the messages may be more specialised in topic than received by a standard user but suggest that it can be used as a reasonable substitute for legitimate email in preliminary testing. SpamArchive maintains an archive of spam contributed by users. Archives are created that contain all spam received by the archive on a particular day, providing researchers with an easily accessible collection of up-to-date spam emails. As a result of the Enron bankruptcy, 400 MB of realistic workplace email has become publicly available: it is likely that this will form part of future standard corpora, despite some outstanding issues [11].

Building an artificial corpus or a corpus from presorted user email ensures the class of each message is known with certainty. However, when dealing with a public corpus (like the Enron corpus), it is more difficult to determine the actual class of a message for accurate evaluation of filter performance. Therefore, Cormack and Lynam [11] propose establishing a 'gold standard' for each message, which is considered to be the message's actual class. They use a bootstrap method based on several different classifiers to simplify the task of sorting through this massive collection of email; it remains as a work in progress. Their filter evaluation toolkit, given a corpus and a filter, compares the filter classification of each message with the gold standard to report effectiveness measures with 95% confidence limits.

In order to compare different filtering techniques, a standard set of legitimate and spam email must be used; both for the testing and the training (if applicable) of filters. Independent tests of filters are generally limited to usable commercial and open source products, excluding experimental classifiers appearing only in research. Experimental classifiers are generally only compared against standard techniques (e.g. Bayesian filtering) in order to establish their relative effectiveness; however this makes it difficult to isolate the most promising new techniques. NetworkWorldFusion [51] review 41 commercial filtering solutions, while Cormack and Lyman review six open source filtering products [10].

## 4.2 Case study

Throughout this paper we have discussed the advances made in spam filtering technology. In this section, we evaluate the extent to which users at the University of Canterbury could potentially benefit from these advances in filtering techniques. Furthermore, we hope to collect data to substantiate some recommendations when evaluating spam filters.

The University of Canterbury maintains a two-stage email filtering solution. A subscription DNS blacklisting system is used in conjunction with Process Software's PreciseMail Anti-spam System (PMAS). The University of Canterbury receives approximately 110,000 emails per day, of which approximately 50,000 are eliminated by the DNS blacklisting system before delivery is complete. Of those emails that are successfully delivered, PMAS discards around 42% and quarantines around 35% for user review. In its standard state, PMAS filters are based on a comprehensive heuristic rule collection and be combined with both server-level and user-level block and allow lists. However, the software has a Bayesian filtering option, that works in conjunction with the heuristic filter, and which was not currently active before the evaluation.

Two experiments were conducted. The first used the publicly available SpamAssassin corpus to provide a comparable evaluation of PMAS in terms of false positives and false negatives. This experiment aimed to evaluate the overall performance of the filter, as well as the relative performance of the heuristic and Bayesian components. The second used spam collected from the SpamArchive repository to evaluate false positive levels on spam collected at various points over the last two years. The aim of this experiment was to observe whether the age of spam has any effect on the effectiveness of the filter, as well as attempting to compensate for the age of the SpamAssassin corpus.

The training of the PMAS Bayesian filter took place over 2 weeks. PMAS automatically (as recommended by the vendor) trains the Bayesian filter by showing it emails that score[5] above and below defined thresholds, as examples of spam and non-spam respectively.

The results of passing the partial SpamAssassin corpus through the PMAS filter can be seen in figure 4. The partial corpus has the 'hard' spam removed, which consists of email with unusual HTML markup, coloured text, spam-like phrases etc. The use of the full corpus increases false positives made by the overall filter from 1 to 4% of all legimitate messages filtered.

The spam corpus drawn from the SpamArchive was constructed from the spam email submitted manually (by users) to SpamArchive on the 14, 15 and 16th of each month used. These dates were randomly chosen. The total number of emails collected at each point varied from approximately 1700 to 3200.

The performance of each filter (heuristic, Bayesian and combined) steadily declined over time as newer spam from the SpamAssassin corpus was introduced. It is assumed that spam more recently submitted to the archive would be more likely to employ newer message construction techniques. No effort has been made to individually examine the test corpus to identify these characteristics. Any person with an email account can submit spam to the archive: this should create a sufficiently diverse catchment base, ensuring a broad range of spam messages are archived. A broad corpus of spam should reflect, to some extent, new spam construction techniques. The fact that updates are regularly issued by major anti-spam product vendors indicates that such techniques are becoming widespread.

Overall results are consistent with those published by NetworkWorldFusion [51]: they recorded 0.75% false positives, and 96% accuracy, while we recorded 0.75% (with the partial SpamAssassin corpus) false positives and 97.67% accuracy.

Under both the full and partial SpamAssassin corpora, the combined filtering option surpasses the alternatives in the two key areas: a lower level of false positives, and a higher level of spam caught (i.e. discarded). This can be clearly seen in figure 4. In terms of these measures, the heuristic filter is closest to the performance of the combined filter. This is unsurprising given that the Bayesian component of the combined filter contributes relatively little and that it was initially trained by the heuristic filter. The Bayesian filter performs comparatively worse than the other two filtering option, as less email is correctly treated

[5] Scores were generated by the heuristic filter.

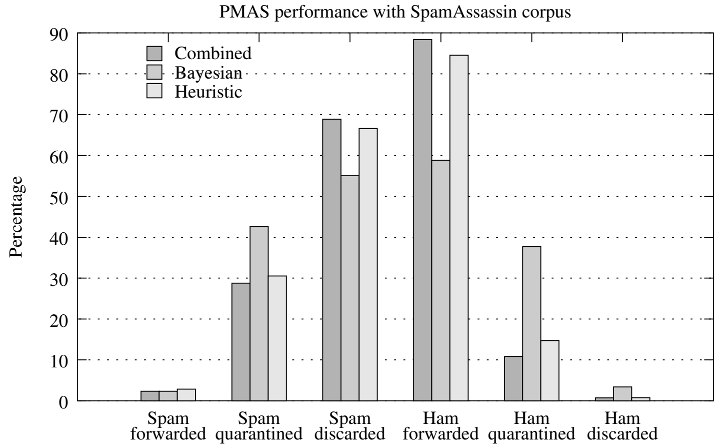

Figure 4: Performance of the PMAS filtering elements using the partial SpamAssassin public corpus.

(i.e. spam discarded or ham forwarded) and notably more email is quarantined for user review. This is consistent with Garcia et al. [19], who suggested such a filtering solution was best placed at the user, rather than the server, level.

The performance of the heuristic filter deteriorates as messages get more recent. This would suggest that the PMAS rule set and underlying software has greater difficulty identifying a spam message when its message is deliberately obscured by advanced spam construction techniques. This is despite regular updates to the filter rule set and software. The combined filter performs similarly to the heuristic filter. This is unsurprising given that the heuristic filter contributes the majority of the message's score (which then determines the class of the message). The introduction of Bayesian filtering improved overall filter performance in all respects when dealing with both the SpamAssassin archive and the SpamArchive collections.

The results from the Bayesian filter are less obvious. One would expect the Bayesian filter to become more effective over time, given that it has been trained exclusively on more recent messages. In the broadest sense, this can be observed: the filter's performance improves by 7% on the January 2005 collection when compared against the July 2003 collection. However, the filter appears to perform best on the 2004 collections (January and July). It is possible that this is due to the training of the Bayesian filter; the automated training performed by PMAS may have incorrectly added some tokens to the ham/spam databases. Furthermore, the spam received by the University of Canterbury may not reflect the spam received by the SpamArchive; this would therefore impact the training of the Bayesian filter.

New spam construction techniques are likely to have impacted on the lower spam accuracy scores; heuristic filters seem especially vulnerable to these developments. It is reasonable to say that such techniques are effective: a regularly updated heuristic filter becomes less effective and therefore reinforces the need for a complementary machine learning approach when assembling a filtering solution.

Broadly, one can conclude two things from this experiment. Firstly, the use of a Bayesian filtering component improves overall filter performance; however, it is not a substitute for the traditional heuristic filter, but more a complement (at least at the server level). Secondly, the concerns raised about the effects of time on the validity of the corpora seem to be justified: older spam does seem to be more readily identified, suggesting changing techniques.

It is interesting to note that, despite improved performance, the Bayesian filtering component was deactivated some months after the completion of this evaluation due to increasing CPU and memory demands on the mail filtering gateway. This can be primarily attributed to the growth of the internal token database, as the automatic training system remained active throughout the period; arguably this could have been disabled once a reasonably sized database had been constructed but this would have negated some of the benefits realised by a machine learning-based filtering system (such as an adaptive rule set). This is a weakness of both the implementation, as no mechanism was provided to reduce the database size, and of the Bayesian approach and unified model machine learning approaches in general. When constructing a unified model, the text of each incoming message affects the current model; however, reversing these changes can be particularly difficult. In the case of a Bayesian filter, a copy of each message processed (or some kind of representative text) would be necessary to reverse the impact of past messages on the model.

## 5 Conclusion

Spam is rapidly becoming a very serious problem for the internet community, threatening both the integrity of networks and the productivity of users. Anti-spam vendors offer a wide array of products designed to keep spam out; these are implemented in various ways (software, hardware, service) and at various levels (server and user). The introduction of new technologies, such as Bayesian filtering, is improving filter accuracy; we have confirmed this for ourselves after examining the Precise-Mail Anti-Spam system. The net is being tightened even further: a vast array of new techniques have been evaluated in academic papers, and some have been taken into the community at large via open source products. The implementation of machine learning algorithms is likely to represent the next step in the ongoing fight to reclaim our inboxes.

## References

[1] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, and C. Spyropoulos. An evaluation of naive bayesian anti-spam filtering. In *Proc. of the workshop on Machine Learning in the New Information Age*, 2000.

[2] I. Androutsopoulos, J. Koutsias, K. Chandrinos, and C. Spyropoulos. An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–167. ACM Press, 2000.

[3] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. Spyropoulos, and P. Stamatopoulos. Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. In *Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, 2000.

[4] P.O. Boykin and V. Roychowdhury. Personal email networks: An effective anti-spam tool. In *MIT Spam Conference*, Jan 2005.

[5] M. Cain. Spam blocking: What matters. META Group, 2003. www.postini.com/brochures.

[6] X. Carreras and L. Márquez. Boosting trees for anti-spam email filtering. In

*Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing*, 2001.

[7] S. Chhabra, W. Yerazunis, and C. Siefkes. Spam filtering using a markov random field model with variable weighting schemas. In *Data Mining, Fourth IEEE International Conference on*, pages 347–350, 1–4 Nov. 2004.

[8] T. Chiu. Anti-spam appliances are better than software. NetworkWorldFusion, March 1 2004. www.nwfusion.com/-columnists/2004/0301faceoffyes.html.

[9] J. Clark, I. Koprinska, and J. Poon. A neural network based approach to automated e-mail classification. In *Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on,*, pages 702–705, 13–17 Oct 2003.

[10] G. Cormack and T. Lynam. A study of supervised spam detection applied to eight months of personal e-mail. http://plg.uwaterloo.ca/ gvcormac/spamcormack.html, July 1 2004.

[11] G. Cormack and T. Lynam. Spam corpus creation for TREC. In *Conference on Email and Anti-Spam*, 2005.

[12] G. Cormack and T. Lynam. TREC 2005 spam track overview. In *Text Retrieval Conference*, 2005.

[13] P. Cunningham, N. Nowlan, S. Delany, and M. Haahr. A case-based approach to spam filtering that can track concept drift. In *ICCBR'03 Workshop on Long-Lived CBR Systems*, June 2003.

[14] W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. Timbl: Tilburg memory based learner, version 3.0, reference guide. ILK, Computational Linguistics, Tilburg University. http://ilk.kub.nl/ ilk/papers, 2000.

[15] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. P2P-based collaborative spam detection and filtering. In *P2P '04: Proceedings of the Fourth International Conference on Peer-to-Peer Computing (P2P'04)*, pages 176–183. IEEE Computer Society, 2004.

[16] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. Using digests to identify spam messages. Technical report, University of Milan, 2004.

[17] H. Drucker, D. Wu, and V.N. Vapnik. Support vector machines for spam categorization. *Neural Networks, IEEE Transactions on*, 10(5):1048–1054, Sep. 1999.

[18] T. Espiner. Demand for anti-spam products to increase. ZDNet UK, Jun 2005.

[19] F.D. Garcia, J.-H. Hoepman, and J. van Nieuwenhuizen. Spam filter analysis. In *Proceedings of 19th IFIP International Information Security Conference, WCC2004-SEC*, Toulouse, France, Aug 2004. Kluwer Academic Publishers.

[20] J. Golbeck and J. Hendler. Reputation network analysis for email filtering. In *Conference on Email and Anti-Spam*, 2004.

[21] Luiz Henrique Gomes, Cristiano Cazita, Jussara M. Almeida, Virgilio Almeida, and Jr. Wagner Meira. Characterizing a spam traffic. In *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 356–369. ACM Press, 2004.

[22] P. Graham. A plan for spam. http://paulgraham.com/spam.html, August 2002.

[23] P. Graham. Better bayesian filtering. In *Proc. of the 2003 Spam Conference*, January 2003.

[24] J. Graham-Cumming. The spammers' compendium. www.jgc.org/tsc/index.htm, Feb 2005.

[25] A. Gray and M. Haadr. Personalised, collaborative spam filtering. In *Conference on Email and Anti-Spam*, 2004.

[26] J.M.G. Hidalgo. Evaluating cost-sensitive unsolicited bulk email categorization. In *SAC '02: Proceedings of the 2002 ACM symposium on Applied computing*, pages 615–620. ACM Press, 2002.

[27] R. Hunt and A. Cournane. An analysis of the tools used for the generation and prevention of spam. *Computers & Security*, 23(2):154–166, 2004.

[28] J. Ioannidis. Fighting spam by encapsulating policy in email addresses. In *Network and Distributed System Security Symposium*, Feb 6–7 2003.

[29] R. Jennings. The global economic impact of spam, 2005 report. Technical report, Ferris Research, 2005.

[30] T. Zeller Jr. Law barring junk e-mail allows a flood instead. The New York Times, Feb 1 2005.

[31] H. Lee and A. Ng. Spam deobfuscation using a hidden markov model. In *Conference on Email and Anti-Spam*, 2005.

[32] B. Leiba, J. Ossher, V. Rajan, R. Segal, and M. Wegman. SMTP path analysis. In *Conference on Email and Anti-Spam*, 2005.

[33] J. Levine. Experiences with greylisting. In *Conference on Email and Anti-Spam*, 2005.

[34] M. Ludlow. Just 150 'spammers' blamed for e-mail woe. The Sunday Times, 1 December 2002.

[35] Mail Abuse Prevention Systems. Definition of spam. www.mail-abuse.com/spam_def.html, 2004.

[36] N. Nie, A. Simpser, I. Stepanikova, and L. Zheng. Ten years after the birth of the internet, how do americans use the internet in their daily lives? Technical report, Stanford University, 2004.

[37] R. Nutter. Software or appliance solution? NetworkWorldFusion, March 1 2004. www.nwfusion.com/columnists/-2004/0301nutter.html.

[38] C. O'Brien and C. Vogel. Spam filters: bayes vs. chi-squared; letters vs. words. In *ISICT '03: Proceedings of the 1st international symposium on Information and communication technologies.* Trinity College Dublin, 2003.

[39] P. Pantel and D. Lin. Spamcop—a spam classification & organisation program. In *Learning for Text Categorization: Papers from the 1998 Workshop*, Madison, Wisconsin, 1998. AAAI Technical Report WS-98-05.

[40] L. Pelletier, J. Almhana, and V. Choulakian. Adaptive filtering of spam. In *Communication Networks and Services Research, Second Annual Conference on*, pages 218–224, 19–21 May 2004.

[41] Postini Inc. Postini perimeter manager makes encrypted mail easy and painless. www.postini.com/brochures, 2004.

[42] Process Software. Explanation of common spam filtering techniques (white paper). http://www.process.com/, 2004.

[43] Radicati Group. Anti-spam 2004 executive summary. Technical report, Radicati Group, 2004.

[44] I. Rigoutsos and T. Huynh. Chung-kwei: a pattern-discovery-based system for the automatic identification of unsolicited e-mail messages (spam). In *Conference on Email and Anti-Spam*, 2004.

[45] G. Rios and H. Zha. Exploring support vector machines and random forests for spam detection. In *Conference on Email and Anti-Spam*, 2004.

[46] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. A bayesian approach to filtering junk E-mail. In *Learning for Text Categorization: Papers from the 1998 Workshop*, Madison, Wisconsin, 1998. AAAI Technical Report WS-98-05.

[47] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. Spyropoulos, and P. Stamatopoulos. A memory-based approach to anti-spam filtering. Technical report, Tech Report DEMO 2001., 2001.

[48] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C.D. Spyropoulos, and P. Stamatopoulos. Stacking classifiers for anti-spam filtering of e-mail. In *Empirical Methods in Natural Language Processing*, pages 44–50, 2001.

[49] K. Schneider. Anti-spam appliances are not better than software. NetworkWorldFusion, March 1 2004. www.nwfusion.com/columnists/2004/-0301faceoffno.html.

[50] C. Siefkes, F. Assis, S. Chhabra, and W. Yerazunis. Combining winnow and orthogonal sparse bigrams for incremental spam filtering. In *Proceedings of ECML/PKDD 2004, LNCS*. Springer Verlag, 2004.

[51] J. Snyder. Spam in the wild, the sequel. http://www.nwfusion.com/-reviews/2004/122004spampkg.html, Dec 2004.

[52] J. Spira. Spam e-mail and its impact on it spending and productivity. Technical report, Basex Inc., 2003.

[53] S. Vaughan-Nichols. Saving private e-mail. *Spectrum, IEEE*, pages 40–44, Aug 2003.

[54] M. Wagner. Study: E-mail viruses up, spam down. Internetweek.com, Nov 9 2002. http://www.internetweek.com/-story/INW20021109S0002.

[55] M. Woitaszek, M. Shaaban, and R. Czernikowski. Identifying junk electronic email in microsoft outlook with a support vector machine. In *Applications and the internet, 2003 Symposium on*, pages 166–169, 27–31 Jan. 2003 2003.

[56] W. Yerazunis. Sparse binary polynomial hashing and the crm114 discriminator. In *MIT Spam Conference*, 2003.

[57] K. Yoshida, F. Adachi, T. Washio, H. Motoda, T. Homma, A. Nakashima, H. Fujikawa, and K. Yamazaki. Density-based spam detector. In *KDD '04: Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 486–493. ACM Press, 2004.

[58] J. Zdziarski. Bayesian noise reduction: contextual symmetry logic utilizing pattern consistency analysis. http://www.nuclearelephant.com/-papers/bnr.html, 2004.