Symbiotic filtering for spam email detection

Clotilde Lopes [a] Paulo Cortez [a,*] Pedro Sousa [b] Miguel Rocha [b] Miguel Rio [c]

[a] *Dep. of Information Systems/Algoritmi, University of Minho, 4800-058 Guimarães, Portugal*

[b] *Dep. of Informatics/CCTC, University of Minho, 4710-059 Braga, Portugal*

[c] *Dep. of Electrical Engineering, University College London, WC1E 7JE UK*

Abstract

This paper presents a novel spam filtering technique called Symbiotic Filtering (SF) that aggregates distinct local filters from several users to improve the overall performance of spam detection. SF is an hybrid approach combining some features from both Collaborative (CF) and Content-Based Filtering (CBF). It allows for the use of social networks to personalize and tailor the set of filters that serve as input to the filtering. A comparison is performed against the commonly used Naive Bayes CBF algorithm. Several experiments were held with the well-known Enron data, under both fixed and incremental symbiotic groups. We show that our system is competitive in performance and is robust against both dictionary and focused contamination attacks. Moreover, it can be implemented and deployed with few effort and low communication costs, while assuring privacy.

Key words: Anti-spam filtering, Naive bayes, Collaborative filtering, Content-based filtering, Word attacks

1 Introduction

Unsolicited bulk email, widely known as spam, has become a serious problem for network administrators and for Internet users in general. According to MAAWG (2009), it accounts for 89% to 92% of all email messages sent and it consumes resources (i.e. time spent reading messages, bandwidth, CPU and disk) that are far away from negligible. Spam is also an intrusion of privacy

* Corresponding author. E-mail pcortez@dsi.uminho.pt; tel.: +351 253510313; fax: +351 253510300.

and used to spread malicious content (e.g. online fraud or viruses). The cost of sending these emails is, however, very close to zero. Internet connections are cheap and with the advent of botnets (e.g. Bobax worms), criminal organizations have access to potentially millions of infected computers and thus send emails from what has to be regarded as legitimate users (Ramachandran and Feamster, 2006; Kanich et al., 2008).

In the last years, the most popular anti-spam solutions have been based in Content-Based Filtering (CBF) (in particular, Bayesian filtering) (Garriss et al., 2006; Guzella and Caminhas, 2009). These class of algorithms use message features (e.g. word frequencies) for statistically discriminating email into legitimate (ham) and illegitimate (spam) messages. However, CBF presents some drawbacks. Often, there is a large gap between the high-level concept (e.g. spam image) and the low-level message features (e.g. bit colors). Also, CBF tends to give weak performances for new users, as it requires a large number of representative examples. Moreover, spammers can mix spam with normal words (often not visible to the final user), in what is known as dictionary or focused attacks (Nelson et al., 2008). When users flag these messages as spam, their training set is contaminated and the CBF performance is heavily reduced. As an alternative, Collaborative Filtering (CF) is a distinct anti-spam strategy, where information (e.g. IP or message fingerprints) is shared about spam messages (Zhong et al., 2008). Yet, pure CF often suffers from several issues (e.g. first-rater, sparsity of data and privacy, see Section 2).

To solve the CBF drawbacks, we propose a novel distributed approach, termed Symbiotic Filtering (SF), that combines features from CBF and CF. Symbiosis is a close interaction among different entities and this phenomenon is present not only in biological species but also in business enterprises (Alocilja, 2000). We prefer the term symbiotic rather than collaborative or cooperative, since in this case each individual may have a distinct goal, as spam by definition is a personal concept. The idea is to promote a cooperation among distinct entities (e.g. email users) interested on personalized filtering (e.g. spam detection). Rather than exchanging messages, these entities will share information about what each local filter has learned (e.g. Bayesian model). The aim of SF is to foster mutual relationships, where all or most members benefit. Under SF, a given user is interested in improving filtering at a personal level. The Internet is used to gather collaborators among these (high number of) users. High group dynamics are expected, as members may join or leave the collaboration, and also there are privacy issues regarding what can be shared. SF is different from the centralized CBF-CF works (e.g. (Yu et al., 2003)) since SF data and models are distributed through different entities. Hence, there are issues of user management (e.g. adding or removing a user), privacy, security and motivation (e.g. each user should benefit from the collaboration).

The main contributions of this paper are: i) we propose the new SF concept

that combines filters from distinct entities in order to improve local filtering while assuring privacy; ii) we apply SF to spam detection and compare it with a local CBF filter (i.e. Naive Bayes); iii) the spam detection performance is measured under distinct scenarios, to test the effect of using fixed and incremental symbiotic groups and also to access the robustness to dictionary and focused attacks. This paper is structured as follows. Section 2 presents the related work. Next, we introduce the individual and symbiotic filtering methods (Section 3). The results are presented in Section 4. Finally, closing conclusions are drawn (Section 5).

2 Related Work

Several solutions have been proposed to fight spam, which can fall into three main categories (Garriss et al., 2006; Méndez et al., 2008): designing New Mail Protocol Systems (NMPS), Collaborative Filtering (CF) and Content-Based Filtering (CBF). Examples of the first approach are: digitally signing mail, where recipients authenticate the sender's address and mail content (Wong, 2005); requiring the sender to "pay" (e.g. give a small fee or solve a computational puzzle) for each message sent (Loder et al., 2004) or limiting the amount of email any sender may send (Walfish et al., 2006). Yet, none of these solutions is currently adopted in a massive fashion and we are still far from a worldwide acceptance of a NMPS. Also, the proposal of payment systems did not take into account the effect spamming botnets, where a large number machines are controlled for malicious messaging (Ramachandran and Feamster, 2006; Kanich et al., 2008).

CF is based on sharing information about spam messages and it can be based on lists (e.g. blacklists with IP addresses of known spammers), or digest/fingerprints extracted from spam messages. Often, DNS-based Blackhole Lists (DNSBLs) work in a centralized fashion, being vulnerable to Denial-of-Service (DoS) attacks. Also, they may blacklist legitimate users, with a false negative rate of about 50%, and spammers that use BGP spectrum agility techniques are rarely listed in DNSBLs (Ramachandran and Feamster, 2006). Several CF systems based on social networks have also been proposed. For example, Kong et al. (2005) suggest a system where users manually identify spam and then publish a digest through her/his social network. Garriss et al. (2006) propose the propagation of whitelists among socially connected users. Zhong et al. (2008) introduce a large-scale privacy CF based on digests.

CBF filters use a text classifier, such as the popular Naive Bayes algorithm (used by the Thunderbird client), that learns to discriminate spam from message features (e.g. common spam words). At the present time, CBF is the most used anti-spam solution (Garriss et al., 2006). Current research relies

mainly on improving individual classifier performance, by a better preprocessing (Méndez et al., 2008) or enhancement of the learning algorithm (Chang et al., 2008; Guzella and Caminhas, 2009). Ensembles that combine distinct spam classifiers have also been proposed (Hershkop and Stolfo, 2005).

Both CF and CBF have drawbacks. CF often suffers from first-rater, sparsity of data and privacy problems. The first issue is due to the difficulty of classifying emails that have not been rated before, the second problem is present when users rate few messages and the last problem depends on what is shared. For example, while presenting a better privacy digest protocol (when compared with previous CF solutions), the approach of Zhong et al. (2008) is still vulnerable to privacy breaches. In addition, people have personal views of what is spam and CF often discards this issue (Gray and Haahr, 2004). On the other hand, CBF frequently suffers from lack of sufficient training messages and is highly vulnerable to contamination attacks (as explained in the previous section).

By fusing the CF and CBF views there is a potential for a better personalized filtering. However, the number of studies that unify CBF and CF is scarce and mainly focused towards recommendation systems that run at centralized systems (Yu et al., 2003). Garg et al. (2006) describe a spam strategy based on sharing of email filters among collaborating users. The authors point out that exchanging filters requires less communication than CF systems, as the need to share filters is relatively rare when compared with exchanging digests each time an email is received. Yet, they fail to acknowledge motivational (i.e. each user should benefit from the collaboration), temporal (i.e. how to synchronize several distinct filters), privacy and security issues regarding filter sharing. More recently, Lai et al. (2009) presented a collaborative approach to exchange spam rules. However, this approach was designed for rule sharing at the server level, demanding secure channels between all these servers. Also, the authors only explored simple attributes (e.g. message length) and not word content. Moreover, explicit (and simple) human understandable rules are required, thus approaches such as Neural Networks or Support Vector Machines are not suitable. In contrast, our SF approach is more flexible, since it can address any type of CBF filter. Also, it is suited for the Web 2.0 paradigm, where users can exchange filters from their social networks. The SF approach should also be more robust to contamination when compared to pure CBF, since it aggregates responses from several (possible unknown) users and targeting a specific victim may be easy but contaminating the whole symbiotic group is not.

3 Filtering Methods

3.1 Naive Bayes filtering

We will address only textual content (i.e. word frequencies) of email messages. This popular approach (e.g. Thunderbird filter) has the advantage of being generalizable to wider contexts, such as spam instant messaging (spim) detection. While different data mining algorithms can be adopted for spam filtering, such as Support Vector Machines (Cheng and Li, 2006), we will use the simpler Naive Bayes (NB), which is widely adopted by anti-spam filtering tools (Metsis et al., 2006; Guzella and Caminhas, 2009). As both individual and symbiotic strategies will be compared using the same learning algorithm, we believe that most of the results presented in this paper can be extended to other text classifiers. We will also adopt the preprocessing proposed in (Kosmopoulos et al., 2008):

(1) The word frequencies are extracted from the subject and body message (with the HTML tags previously removed). Each message j is encoded into a vector $\mathbf{x}_j = (x_{1j}, \ldots, x_{mj})$, where x_{ij} is the number of occurrences of token X_i in the text.
(2) The feature selection is applied, which consists in ignoring any words when $x_{ij} < 5$ in the training set and then selecting up to the 3000 most relevant features according to the Mutual Information ($MI(X_i)$) criterion:

$$MI(X_i) = \sum_{c \in \{s, \neg s\}} p(X_i|c) \log\left(\frac{p(X_i|c)}{p(X_i)p(c)}\right) \tag{1}$$

where c is the message class (s - spam or $\neg s$ - ham), $p(X_i|c)$ is the probability of finding token X_i in emails from class c, $p(X_i)$ and $p(c)$ are the proportions of X_i terms and c class examples present in the data.
(3) Each x_{ij} value is transformed into: $x'_{ij} = \log(x_{ij} + 1)$ (TF transform), $x''_{ij} = x'_{ij} \cdot \log(k/\sum_k \delta_{ik})$ (IDF transform) and $x'''_{ij} = x''_{ij}/\sqrt{\sum_l (x_{lj})^2}$ (length normalization), where δ_{ik} is 1 if the token i exists in the message k and 0 otherwise.

The NB computes the probability that a document j is spam (s) for a filter trained over $\mathcal{D}_u$ email data from user u, according to:

$$p(s|\mathbf{x}_j, \mathcal{D}_u) = \alpha \cdot p(s|\mathcal{D}_u) \prod_i^m p(X_i|s, \mathcal{D}_u) \tag{2}$$

where α is normalization constant that ensures that $p(s|\mathbf{x}, \mathcal{D}_u)+p(\neg s|\mathbf{x}, \mathcal{D}_u) = 1$, $p(s|\mathcal{D}_u)$ is the $p(s)$ of dataset $\mathcal{D}_u$. The $p(X_i|s, \mathcal{D}_u)$ estimation depends on the NB version. In this work, we will use the multi-variate Gauss NB (as implemented in the **R** tool, see Section 4) (Metsis et al., 2006):

$$p(X_i|c, \mathcal{D}_u) = \frac{1}{\sigma_{i,c}\sqrt{2\pi}} \exp(-\frac{(x'''_{ij} - \mu_{i,c})^2}{2\sigma^2_{i,c}}) \tag{3}$$

where $\mu_{i,s}$ and $\sigma_{i,s}$ are the mean and standard deviation estimated from the $c = s$ or $c = \neg s$ messages of $\mathcal{D}_u$.

In (Nelson et al., 2008), it has been shown that local spam filters are vulnerable to dictionary and focused contamination attacks. The former attack is used to reduce the CBF efficiency, leading the victim to read spam, while the latter can be used to prevent the victim from reading an important email. Both attacks can be achieved by sending spam messages mixed with normal words. Once the victim labels these messages as spam, the training set is contaminated and the filter will be affected the next time it is retrained. A dictionary aggression consists in sending a large amount of normal words, while the focused assault assumes that the attacker has some knowledge of a specific message that the victim will receive in the future (e.g. a competing offer for a given contract).

3.2 Symbiotic filtering

In our proposed SF, the individual predictions can be combined by using a collaborative ensemble of the local filters. To tackle the concept drift (i.e. the learning tasks changes through time) nature of spam (Fdez-Riverola et al., 2007), the ideal symbiotic combination function should be dynamic. To achieve this we propose a hierarchical learning, where the outputs of the distinct filters are used as the inputs of another (meta-level) learner. Hence, each user has a local meta-learner that is responsible for aggregating the distinct filter responses. This meta-learner is dynamically trained to get a high accuracy on the user past data, thus it assigns different weights to the CBF filters through time (see Figure 6). While several algorithms could be used for this hierarchical learning (e.g. SVM), we will adopt the same NB described in the previous section. The rationale is that NB is commonly adopted by anti-spam solutions, thus incorporating SF into these tools would be simpler by reuse of code.

We assume that each user u trains a local filter $\theta_{u,t}$ over her/his $\mathcal{D}_u$ training data. Filters can be trained asynchronously and L filters will be available for each user at time t: $\{\theta_{1,t}, \ldots, \theta_{L,t}\}$. The Symbiotic NB (SNB) meta-model

spam probability is given by:

$$
\begin{aligned}
p(s|\mathbf{x}_j, \mathcal{D}'_u) &= \alpha \cdot p(s|\mathcal{D}'_u) \prod_{i=1}^{L} p(\theta_{i,t}|s, \mathcal{D}'_u) \\
p(\theta_{i,t}|c, \mathcal{D}'_u) &= \frac{1}{\sigma_{i,c}\sqrt{2\pi}} \exp(-\frac{(p(s|\mathbf{X}_j, \theta_{i,t}, \mathcal{D}'_u) - \mu_{i,c})^2}{2\sigma_{i,c}^2})
\end{aligned}
\tag{4}
$$

where $\mathcal{D}'_u$ is the SNB training set and $p(s|\mathbf{x}_j, \theta_{i,t}, \mathcal{D}'_u)$ is the probability given by the filter $\theta_{i,t}$, as computed in Equation 2. To reduce memory and computational requirements, we allow that $\mathcal{D}'_u \subseteq \mathcal{D}_u$, where $M = |\mathcal{D}'_u|$ denotes the most recent messages from u mailbox. It should be noted that any token from $\mathbf{x}_j$ that is not considered by $\theta_{i,t}$ will simply be discarded by the filter from user i. Similarly, any input attribute from $\theta_{i,t}$ that is not included in $\mathbf{x}_j$ will be set to 0.

While sharing models is less sensitive than exchanging email messages, there are still privacy issues to be considered. For instance, if user A has access to the filter of user B, then A may feed a given token (or set of tokens) into the model and thus know some probability that such token was classified by B as spam or ham. Our privacy solution resides in an anonymous exchange of the filters, which can occur under a centralized server or a Peer-to-Peer (P2P)-like application.

Under the first option, all users register into a centralized and secure service. This service could be implemented by large companies or email providers (e.g. Gmail or Hotmail), when all emails are stored at a given server. For scalability, user profiles could be defined (e.g. country or profession) and clustering algorithms could be used to group users with similar interests. Another variant would be the definition of social networks, where users could choose their "friends". These systems could, for example, be implemented by social networking websites (e.g. Facebook or MySpace). Alternatively, when the messages are stored locally at the client side, the server would be responsible for a blind exchange of the filters, using secure transfers (left of Figure 1). To exchange the filters, a standard format should be adopted, such as the Predictive Model Markup Language (PMML) (Grossman et al., 2002), which is compatible with a large number of data mining tools. In should be noted that exchanging filters requires less communication costs. For example, a filter built from a millions of emails can be described by a few hundreds or thousands of bytes (depending on the filter algorithm used) (Garg et al., 2006). When a new email is received, the user can easily compute the SF, as a copy of all filters is available locally.

The above systems have the disadvantage of having to depend and trust in a centralized service. As an alternative, the use of a P2P-like distribution scheme is also possible to be adopted (right of Figure 1). Under this solution all peers may donate, store and fetch filters among each other. This approach

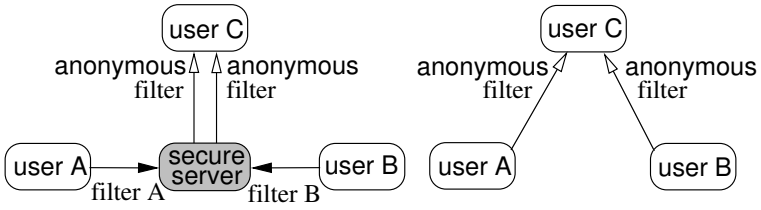

Fig. 1. Anonymous exchange of filters by using a secure server (left) or a P2P-like application (right)

could be implemented as a trusted and secure plug-in of an email client (e.g. Thunderbird). The filter sharing process among the peers could work similarly to the explained in the secure server scheme (i.e. using PMML). Further privacy increase could be achieved if each individual does not know the symbiotic group composition. However, for some scenarios it may also be attractive that the symbiotic group composition could be assessed by all the participants in a given group. Yet, even in such scenarios it might be very difficult to "guess" who created each particular model, as in SF there will be typically a large number of users that dynamically may join or leave the collaboration.

4 Experiments and Results

4.1 Spam data

To evaluate SF, ideally there should be real mailboxes collected from distinct users (possibly from a social network) during a given time period. Yet, due to logistic and privacy issues, it is quite difficult to obtain such data (in particular personal messages) and make it public. Hence, we will use a synthetic mixture of real spam and ham messages, in a strategy similar to what has been proposed in (Metsis et al., 2006; Zhong et al., 2008). The ham messages will come from the Enron email collection, which was originally used for a global evaluation of filters by merging all Enron user messages into a single corpus. In particular, we will use the cleaned-up form provided by (Beckermann et al., 2004) and we will select the five Enron employees with the largest mailboxes collected during the same time period: kaminski-v (kam), farmer-d (far), beck-s (bec), lokay-m (lok) and kitchen-l (kit). Since these employees worked at the same organization, it is reasonable to assume that they would know each other, i.e. belong to a social network. We will also use the spam collection of Bruce Guenter (http://untroubled.org/spam/), which is based in spam traps (i.e. fake emails published in the Web), during the years of 2006 and 2007 (our dataset was built in 2008). Only messages with Latin character sets were selected, because the ham messages use this type of character coding and non-Latin mails would be easy to detect. Also, since this collection con-

tains several copies of the same messages (due to the use of multiple traps), we removed duplicates by comparing MD5 signatures of the body messages.

We propose a mixture algorithm that is based on the time that each message was received (date field, using the GMT time zone). By preserving the temporal order of the emails, we believe that a more realistic mixture is achieved than the sampling procedures adopted in (Metsis et al., 2006) or (Zhong et al., 2008). Since the Enron data is from a previous period (see Table 1), we first added 6 years to the date field of all ham messages. Let S_t denote a spam message received at time t, $S_{i,f} = (S_{t_i}, S_{t_{i+1}}, \ldots, S_{t_f})$ the time ordered sequence of the Bruce Guenter spam, $H_{u,i,f}$ and $S_{u,i,f}$ the sequences of ham and spam messages for user u from time t_i to t_f. For a given time period $t \in (t_i, \ldots, t_f)$, the algorithm randomly selects $|S'_{i,j}|$ spam messages from $S_{i,j}$. Then, $S_{u,i,j}$ is set by sampling messages from $S'_{i,j}$ with a probability of P for each message selection. The size of $S'_{i,j}$ (cardinality) is given by:

$$|S'_{i,j}| = \frac{R \cdot \sum_{i=1}^{L} |H_{u,i,j}|}{P \cdot L} \tag{5}$$

where L denotes the total number of users available at the time period and R is the overall (i.e. including all user and time data) spam/ham ratio. Since the time periods are different for each user (Table 1), four time sequences (i.e. t_i and t_j values) were used by the algorithm (Figure 2).

Table 1
The S-Enron corpus main characteristics

user	**ham size**	**spam size**	**time period**	**spam /ham**
kam	4363	2827	[12/05,05/07]	0.6
far	3294	2844	[12/05,05/07]	0.9
bec	1965	2763	[01/06,05/07]	1.4
lok	1455	2202	[06/06,05/07]	1.5
kit	789	623	[02/07,05/07]	0.8

The mixture is affected by the R and P parameters. Since a high number of experiments is addressed in this work, we will fix these parameters to reasonable values. While the global spam/ham ratio is $R = 1$, the individual ratios range from 0.6 to 1.5. Also, the spam/ham ratios fluctuate through time (as shown in Figure 3). On the other hand, the probability of spam selection affects the percentage of common spam between users. If two users have similar profiles (e.g. email exposure), then they should receive similar spam. We assume that this scenario is expected for the Enron employees and thus set $P = 0.5$. Under this setup and for a given time period, any 2 users will receive around 50% of

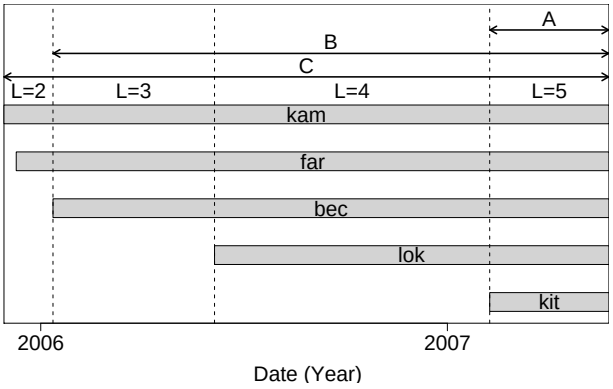

Fig. 2. Time view of the S-Enron mailboxes

similar spam, 3 users will share around 25% of spam and so on. The resulting corpus is named S-Enron and it is publicly available in its raw form at: http://www3.dsi.uminho.pt/pcortez/S-Enron.

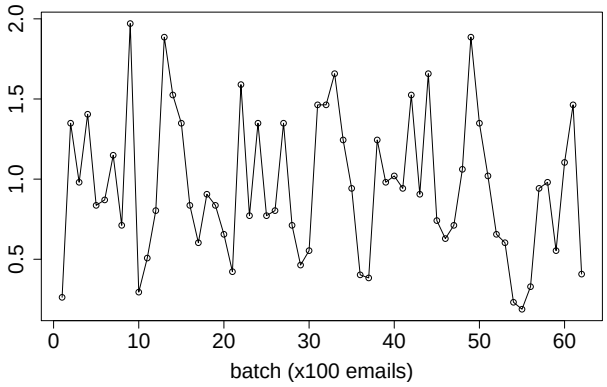

Fig. 3. Evolution of the spam/ham ratio for the user far and scenario C

4.2 Evaluation

As explained in Section 3.2, spam detection suffers from concept drift (Fdez-Riverola et al., 2007). Features such as the amount of spam received, the ham/spam ratio and even the content itself evolve through time. Hence, to evaluate spam filters, we will adopt the more realistic incremental retraining evaluation procedure, which periodically trains and tests filters. Under this procedure, a mailbox is split into batches $b_1, \ldots, b_n$ of K adjacent messages ($|b_n|$ may be less than K) (Metsis et al., 2006). For $i \in \{1, \ldots, n-1\}$, the filter is trained with $\mathcal{D}_u = b_1 \cup \ldots \cup b_i$ and tested with the messages from b_{i+1} (Figure 4). This procedure is more realistic than the simple 50% train/test split adopted in (Zhong et al., 2008).

The predicted class for a probabilistic filter is given by s if $p(s|\mathbf{x}_j, \mathcal{D}_u) > D$, where $D \in [0.0, 1.0]$ is a decision threshold. For a given D and test set, it is

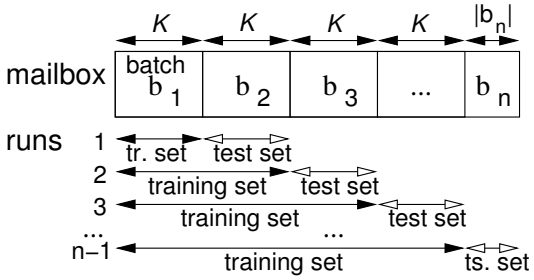

Fig. 4. The incremental retraining procedure

possible to compute the true (TPR) and false (FPR) positive rates:

$$\begin{aligned} TPR &= TP/(TP+FN) \\ FPR &= FP/(TN+FP) \end{aligned} \tag{6}$$

where TP, FP, TN and FN denote the number of true positives, false positives, true negatives and false negatives. The receiver operating characteristic (ROC) curve shows the performance of a two class classifier across the range of possible threshold (D) values, plotting FPR (x-axis) versus TPR (y-axis) (Fawcett, 2006). The global accuracy is given by the area under the curve (AUC= $\int_0^1 ROCdD$). A random classifier will have an AUC of 0.5, while the ideal value would be 1.0. Since the cost of losing normal email (FP) is much higher than receiving spam (FN), D is usually set to favor points in the low false-positive region of the ROC. Thus, we will also adopt the Normalized AUC (NAUC), which is the AUC area in the section $FPR \leq r$, divided by r (Chang et al., 2008). Typically, the target FPR rate (r) is close to 0.0. We will also compute the relative Gain of the symbiotic performance over the local filter: Gain= $\xi_{\mathrm{SNB}}/\xi_{\mathrm{NB}} - 1$, where ξ is the evaluation metric (i.e. AUC or NAUC) and SNB and NB are the symbiotic and individual filters. With the incremental retraining procedure, one ROC will be computed for each b_{i+1} batch and the overall results will be presented by adopting the vertical averaging ROC (i.e. according to the FPR axis) algorithm presented in (Fawcett, 2006). Statistical confidence will be given by a paired t-student test, at the 95% confidence level (Flexer, 1996).

4.3 Experimental setup

All experiments were conducted in the **R** environment, an open source and high-level programming language for data analysis (R Development Core Team, 2008). In particular, the NB algorithm described in Section 3.1 is implemented by the naiveBayes function of the **e1071** R package, while the text preprocessing uses several functions from the **tm** package (Feinerer et al., 2008).

During the all experiments, we set $K = 100$ (a reasonable value also adopted in (Metsis et al., 2006)). For the SNB, we used a similar number for the hierarchical training set size, i.e. $M = 100$. This small value has the advantage of reducing memory requirements (the user only needs 100 messages in his mailbox) and some initial experiments with larger values of M revealed no gain in performance. The AUC, NAUC and Gain values will be shown in percentage. All NAUC values will be computed with $r = 0.01$ (1%).

4.4 Fixed symbiotic group

Two distinct scenarios will be tested, according to the time periods A and B of Figure 2. Given the S-ENRON corpus characteristics, in this work we will explore a small number of fixed symbiotic users: $L = 5$ for A and $L = 3$ for B.

The incremental retraining method (Section 4.2) was applied to both scenarios, by considering all messages within the corresponding time period. Thus, the number of kam, far and bec batches (n) will be different for A and B. The obtained results are summarized as the mean of all test sets (b_{i+1}, $i \in \{1, \ldots, n-1\}$) and shown in Table 2 and Figure 5. The best values are in **bold**, while underline denotes a statistical significance (i.e. p-value<0.05). In Figure 5, bars denote 95% t-student confidence intervals and only the most interesting region of FPR is shown for the ROC curves.

For the first scenario (A), the symbiotic strategy outperforms the local filter for all users and metrics, except for lok and NAUC. A similar behavior occurs for the B setting, where SNB is better than NB except for kam and AUC. As false positives have higher costs in spam detection, the NAUC results are particularly important. Thus, it is interesting to notice that there is a high AUC improvement (i.e. Gain) given by the symbiotic method in several cases (kam, far and kit for A and bec for B).

To demonstrate the SNB dynamics, Figure 6 shows the first two consecutive graphs of the SNB input importances under scenario II. Each edge represents the influence (in %) of the NB filter (the origin) in the symbiotic model (the destination), as measured by applying a sensitivity analysis procedure (Kewley et al., 2000). The text in bold (e.g. $\mathbf{b_2}$) denotes the last batch used to train the NB classifier. For example, the first SNB model of user far (left graph) uses a NB filter from kam that was trained using 200 messages ($\mathcal{D}_{\text{kam}} = b_1 \cup \mathbf{b_2}$).

Table 2
The results for scenarios A and B

scenario A			**AUC**			**NAUC**	
user	n	**NB**	**SNB**	**Gain**	**NB**	**SNB**	**Gain**
kam	16	62.1	**95.6**	54	0.8	**74.4**	9804
far	13	93.5	**95.1**	2	15.6	**58.6**	276
bec	9	91.5	**94.0**	3	53.5	**65.8**	23
lok	9	91.4	**95.2**	4	**79.9**	75.6	-3
kit	15	74.6	**95.3**	28	18.2	**71.7**	294
scenario B			**AUC**			**NAUC**	
user	n	**NB**	**SNB**	**Gain**	**NB**	**SNB**	**Gain**
kam	70	**94.7**	94.3	-0.4	54.0	**73.0**	35
far	60	89.4	**91.7**	2.6	54.3	**66.9**	23
bec	48	83.5	**93.4**	11.9	23.8	**74.3**	212

4.5 Incremental symbiotic group

A more realistic scheme is adopted for the time period C, where users join the symbiotic group in an incremental fashion, at different time stages according to Figure 2. Thus, L will grow from 2 to 5. The results are presented in Table 3. As expected, the symbiotic strategy (SNB) clearly favors newcomers, which have small mailboxes and thus benefit from the collaboration. In effect, the NAUC differences are quite large, such as in bec and lok for $L = 4$ and $L = 5$; and kit for $L = 5$. For demonstration purposes, the ROC curves are plotted for kam, far and bec, when $L = 3$ and $L = 5$ (Figure 7). However, the results show that even "veteran" users benefit from the symbiotic relation when the number of users grow. For instance, the kam and far NAUC results for $L = 5$ improve, with a gains of 28% and 16%, respectively.

4.6 Contamination attacks

We will repeat the experiments of Section 4.4, by considering only scenario A and user bec to test the effects of mailbox contamination. The dictionary assault is simulated by replacing the first 10 spam emails at batch 4 from bec by the GNU aspell (http://aspell.net/) English dictionary (version 6.0, with 138599 tokens). In Figure 8, gray lines denote the behavior of NB and SNB without the attack (i.e. results of Section 4.4), black lines show the performance under the attack and the dot-dashed vertical line shows when the

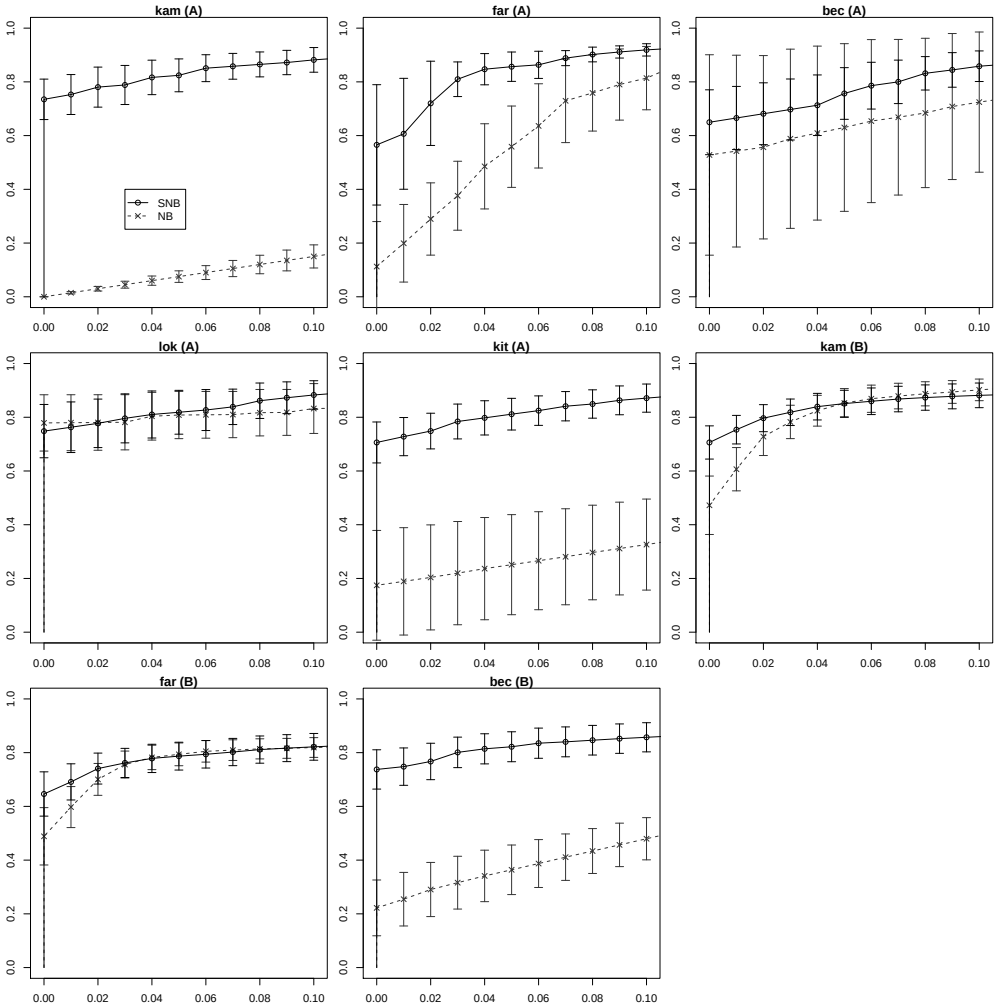

Fig. 5. ROC curves for scenarios A and B

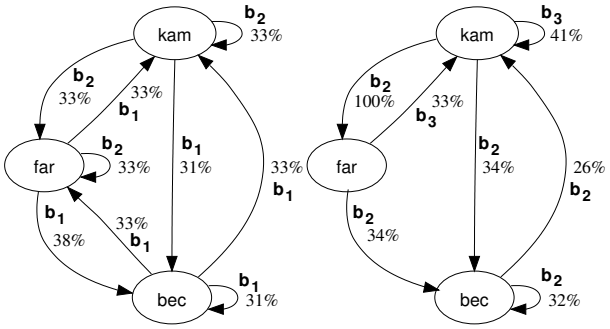

Fig. 6. Examples of SNB input importances for B

attack starts. The filters of bec are not trained with contaminated messages at batch 4, yet these messages appear in the test set and thus the NB and SNB performances suffer a moderate decay. The true effect of the attack is only visible at batch 5, where local CBF is highly affected. Only 10 messages were replaced and yet the filter detection capability is reduced to a random classifier (since AUC=0.5) through all remaining batches. In contrast, the symbiotic

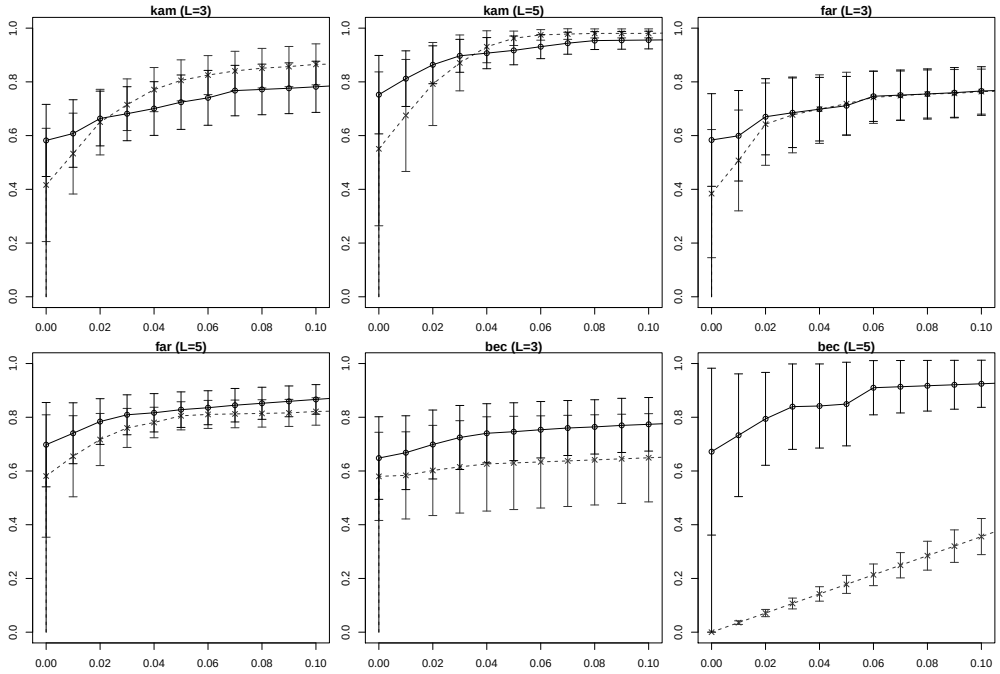

Fig. 7. ROC curves for users kam, far and bec (L=3 at left and L=5 at right)

method is only initially affected, since as time goes by the performance gets closer to the no attack scenario. Also, the remaining symbiotic users maintain their spam detection capabilities, as shown by the far results, which is a representative example. This behavior is explained by the SNB algorithm, which simply discards a given filter if it does not help to predict the recent past M messages of the user. Hence, this experiment shows that SF is robust also to saboteurs, i.e. if a particular user intentionally feeds the group with a random or bad filter then this filter will be simply ignored.

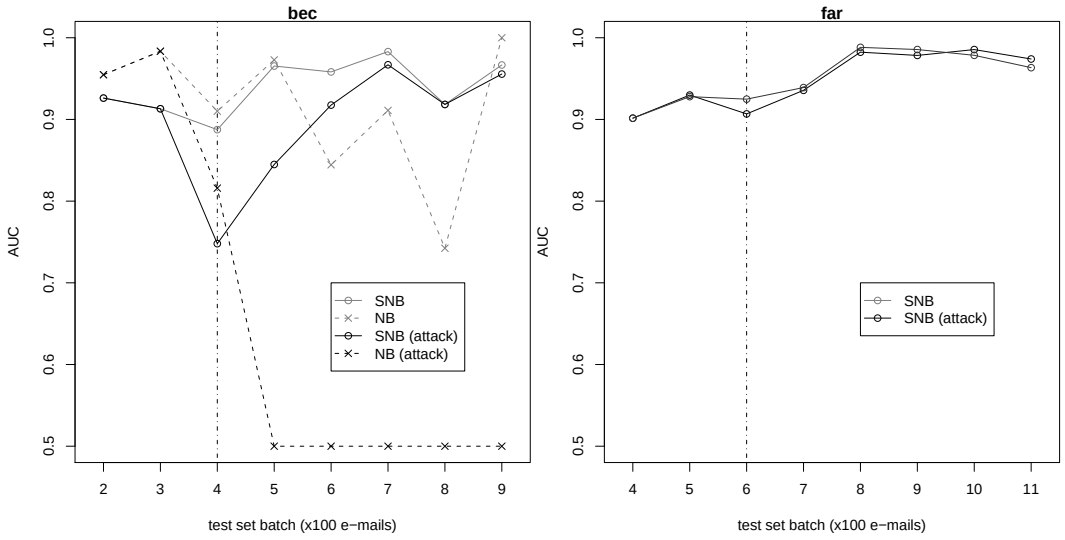

Fig. 8. The effect of the dictionary attack

The dictionary attack can be solved by performing a rollback (i.e. returning

Table 3
The results for scenario C

L=2			**AUC**			**NAUC**	
user	n	**NB**	**SNB**	**Gain**	**NB**	**SNB**	**Gain**
kam	4	**94.4**	88.1	-3	30.1	**48.4**	61
far	3	**89.9**	82.1	-9	**60.7**	54.3	-11
L=3			**AUC**			**NAUC**	
user	n	**NB**	**SNB**	**Gain**	**NB**	**SNB**	**Gain**
kam	17	**91.5**	87.4	-5	47.5	**59.5**	25
far	16	86.6	**87.0**	0.4	44.6	**59.1**	33
bec	14	80.4	**87.6**	9	58.2	**65.8**	13
L=4			**AUC**			**NAUC**	
user	n	**NB**	**SNB**	**Gain**	**NB**	**SNB**	**Gain**
kam	39	95.4	**96.8**	1.5	75.0	**75.6**	0.8
far	32	88.7	**94.7**	6.8	53.1	**72.5**	37
bec	28	84.5	**96.9**	15	13.0	**78.6**	504
lok	29	73.1	**96.2**	32	6.1	**71.5**	1076
L=5			**AUC**			**NAUC**	
user	n	**NB**	**SNB**	**Gain**	**NB**	**SNB**	**Gain**
kam	15	**98.4**	98.0	-0.5	61.3	**78.2**	28
far	13	89.6	**95.5**	6.5	61.8	**71.9**	16
bec	8	85.2	**97.4**	14	1.8	**70.3**	3848
lok	9	63.6	**97.3**	53	0.7	**78.3**	10992
kit	15	74.6	**97.9**	31	18.2	**71.8**	295

to the previous filter) or using the RONI defense (Nelson et al., 2008), which rejects training examples that have a large negative impact in spam detection. Yet, focused assaults are more difficult to prevent and finding a defense is still an open problem (Nelson et al., 2008). We believe SF is an interesting solution due to the same rationale presented for the dictionary aggression, i.e. the combination of multiple filters should overcome the limitations of a single model contamination.

A new set of experiments was devised, using again scenario A and bec mailbox. During a given run, a legitimate message was randomly selected, from batches 6 to 9, as the target text. We assume that the attacker is confident about the

target content and thus can guess 50% of the target words. At batch 4, 10 spam emails were replaced by the contaminated messages. We repeated this procedure during 20 runs. The effect of this attack on spam is minimal and thus we will only show the effect on the target ham emails. Figure 9 plots the filter spam probability (y-axis) for each target message. The obtained probability for each run is plotted along the x-axis (total of 20 runs). Since all target messages are ham, a robust filter should present low spam probabilities, near the zero horizontal axis. The results show that local filter (NB) is much more vulnerable to focused attacks than the symbiotic strategy (SNB). The spam probability mean values of NB and SNB are 0.69 and 0.32 (the differences are statistically significant). For example, when using a decision threshold of $D = 0.5$, 14 (of 20) messages are classified by NB as spam, while this number lowers to 6 for SNB. Even if D is raised to 0.999, NB predicts 13 spam emails and SNB only detects 5.

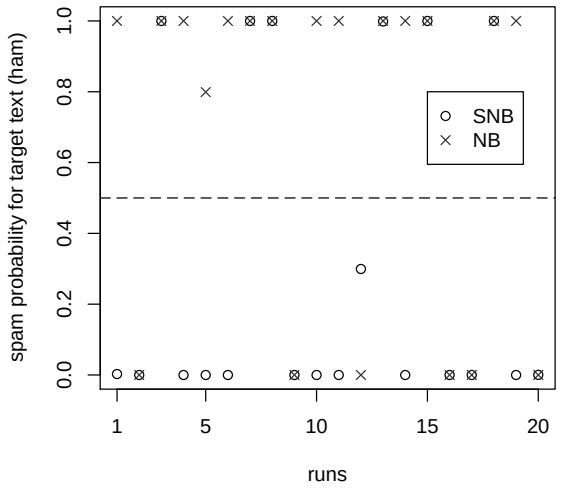

Fig. 9. The effect of the focused attack

5 Conclusions

Email has long been one of the most important and widely used Internet applications. However, spam emerged quickly after email itself and nowadays it accounts for the majority of the email traffic. Thus, several anti-spam techniques were developed. This paper proposes a novel Symbiotic Filtering (SF) approach, which combines several features from Collaborative Filtering (CF) and Content-Based filtering (CBF). It takes the advantage of use of social networks, where users with the same or related interests have the opportunity to form mutually beneficial alliances with the aim of enhancing spam detec-

tion techniques. Instead of sharing messages or digests, the idea is to share filters. To combine the individual probabilities, the proposed solution uses the concept of hierarchical learning, where a meta-learner is dynamically trained to improve its accuracy.

After describing the proposed model, we compared the effectiveness of the SF versus CBF, using for that purpose a realistic mixture of real spam and ham messages. Promising results were obtained by the SF, which outperformed the local filtering for a small number of users (from 3 to 5). Moreover, we have shown that SF is more robust to word attacks (e.g. dictionary or focused assaults). Furthermore, we proposed several deployment scenarios for SF, under a secure server or P2P settings.

There is a continuous race between spammers and anti-spammers and a local classifier that is currently perfect will be eventually defeated. We believe that a stronger protection is achieved by adopting a dynamic cooperation of filters from distinct users. In future work, we intent to apply SF to other personalized filtering scenarios, such as Web page blocking (e.g. sensitive content). Also, we will explore scalability issues. Under a large group, this could be achieved by adopting user selection algorithms (e.g. clustering user profiles).

Acknowledgments

This work is supported by FCT grant PTDC/EIA/64541/2006.

References

Alocilja, E. (2000). *Principles of Biosystems Engineering*. Erudition Books, Massachusetts, USA.

Beckermann, R., McCallum, A., and Huang, G. (2004). Automatic categorization of email into folders: benchmark experiments on Enron and SRI corpora. Ir-418, University of Massachusetts Amherst.

Chang, M., Yih, W., and Meek, C. (2008). Partitioned Logistic Regression for Spam Filtering. In *14th ACM SIGKDD int. conference on Knowledge discovery and data mining*, pages 97–105.

Cheng, V. and Li, C. (2006). Personalized Spam Filtering with Semi-supervised Classifier Ensemble. In *IEEE/WIC/ACM International Conference on Web Intelligence*.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874.

Fdez-Riverola, F., Iglesias, E. L., Díaz, F., Méndez, J. R., and Corchado, J. M.

(2007). Applying lazy learning algorithms to tackle concept drift in spam filtering. *Expert Systems with Applications*, 33(1):36–48.

Feinerer, I., Hornik, K., and Meyer, D. (2008). Text Mining Infrastructure in R. *Journal of Statistical Software*, 25(1-54).

Flexer, A. (1996). Statistical Evaluation of Neural Networks Experiments: Minimum Requirements and Current Practice. In *Proceedings of the 13th European Meeting on Cybernetics and Systems Research*, volume 2, pages 1005–1008, Vienna, Austria.

Garg, A., Battiti, R., and Cascella, R. (2006). May I borrow your filter? Exchanging filters to combat spam in a community. In *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on*, volume 2.

Garriss, S., Kaminsky, M., Freedman, M., Karp, B., Mazières, D., and Yu, H. (2006). RE: reliable email. In *Proceedings of the 3rd conference on Networked Systems Design and Implementation (NSDI)*, pages 297–310, San Jose, CA. USENIX Association Berkeley, CA, USA.

Gray, A. and Haahr, M. (2004). Personalised, Collaborative Spam Filtering. In *1st Conference on E-Mail and Anti-Spam CEAS*.

Grossman, R., Hornick, M., and Meyer, G. (2002). Data Mining Standards Initiatives. *Communications of ACM*, 45(8):59–61.

Guzella, T. and Caminhas, W. (2009). A review of machine learning approaches to Spam filtering. *Expert Systems with Applications*, 36:10206–10222.

Hershkop, S. and Stolfo, S. (2005). Combining Email Models for False Positive Reduction. In *11th ACM SIGKDD int. conference on Knowledge discovery and data mining*, pages 21–24.

Kanich, C., Kreibich, C., Levchenko, K., Enright, B., Voelker, G., Paxson, V., and Savage, S. (2008). Spamalytics: An Empirical Analysis of Spam Marketing Conversion. In *Computer and Communications Security Conference (CCS'08)*, pages 27–31. ACM.

Kewley, R., Embrechts, M., and Breneman, C. (2000). Data Strip Mining for the Virtual Design of Pharmaceuticals with Neural Networks. *IEEE Trans Neural Networks*, 11(3):668–679.

Kong, J., Boykin, P., Rezaei, B., Sarshar, N., Roychowdhury, V., Rothenstein, B., Damian, I., Paramonov, P., Lyuksyutov, S., Barrat, A., et al. (2005). Let your cyberalter ego share information and manage spam. *eprint arXiv: physics/0504026*.

Kosmopoulos, A., Paliouras, G., and Androutsopoulos, I. (2008). Adaptive Spam Filtering Using Only Naive Bayes Text Classifiers. In *CEAS 2008 - Fifth Conference on Email and Anti-Spam*.

Lai, G., Chen, C., Laih, C., and Chen, T. (2009). A collaborative anti-spam system. *Expert Systems with Applications*, 36:6645–6653.

Loder, T., Van Alstyne, M., and Wash, R. (2004). An economic answer to unsolicited communication. In *proceedings of the 5th ACM conference on Electronic Commerce*, pages 40–50. ACM New York, NY, USA.

MAAWG (2009). Email Metrics Program: The Network Operators' Perspective. Report #10 – third and fourth quarter 2008, Messaging Anti-Abuse Working Group, S. Francisco CA, USA.

Méndez, J., Cid, I., Glez-Peña, D., Rocha, M., and Fdez-Riverola, F. (2008). A Comparative Impact Study of Attribute Selection Techniques on Naïve Bayes Spam Filters. In Springer, editor, *8th Industrial Conference on Data Mining*, volume LNAI 5077, pages 213–227.

Metsis, V., Androutsopoulos, I., and Paliouras, G. (2006). Spam Filtering with Naive Bayes – Which Naive Bayes? In *Third Conference on Email and Anti-Spam (CEAS)*, pages 125–134.

Nelson, B., Barreno, M., Chi, F., Joseph, A., Rubinstein, B., Saini, U., Sutton, C., Tygar, J., and Xia, K. (2008). Exploiting Machine Learning to Subvert Your Spam Filter. In *1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, pages 1–9. ACM Press.

R Development Core Team (2008). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-00-3, http://www.R-project.org.

Ramachandran, A. and Feamster, N. (2006). Understanding the Network-Level Behavior of Spammers. In ACM, editor, *SIGCOMM'06*, pages 291–302.

Walfish, M., Zamfirescu, J., Balakrishnan, H., Karger, D., and Shenker, S. (2006). Distributed Quota Enforcement for Spam Control. In *Proceedings of the 3rd conf. on Networked Systems Design and Implementation (NSDI)*, pages 281–296, San Jose, CA. USENIX Association Berkeley.

Wong, M. (2005). Sender authentication: What to do. *White Paper, July*.

Yu, K., Schwaighofer, A., Tresp, V., Ma, W., and Zhang, H. (2003). Collaborative Ensemble Learning: Combining Collaborative and Content-Based Information Filtering via Hierarchical Bayes. In *19th Int. Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 353–360. ACM.

Zhong, Z., Ramaswamy, L., and Li, K. (2008). ALPACAS: A Large-scale Privacy-Aware Collaborative Anti-spam System. In *IEEE INFOCOM*, pages 556–564.