# Spam Email Filtering
# Using Network-Level Properties

Paulo Cortez[1], André Correia[1], Pedro Sousa[3], Miguel Rocha[3], and Miguel Rio[2]

[1] Dep. of Information Systems/Algoritmi, University of Minho, 4800-058 Guimarães,
Portugal,
pcortez@dsi.uminho.pt
WWW home page: http://www3.dsi.uminho.pt/pcortez
[2] Dep. of Informatics, University of Minho, 4710-059 Braga, Portugal,
{pns, mrocha}@di.uminho.pt
[3] Department of Electronic and Electrical Engineering, University College London,
Torrington Place, WC1E 7JE, London, UK,
m.rio@ee.ucl.ac.uk

**Abstract.** Spam is serious problem that affects email users (e.g. phishing attacks, viruses and time spent reading unwanted messages). We propose a novel spam email filtering approach based on network-level attributes (e.g. the IP sender geographic coordinates) that are more persistent in time when compared to message content. This approach was tested using two classifiers, Naive Bayes (NB) and Support Vector Machines (SVM), and compared against bag-of-words models and eight blacklists. Several experiments were held with recent collected legitimate (ham) and non legitimate (spam) messages, in order to simulate distinct user profiles from two countries (USA and Portugal). Overall, the network-level based SVM model achieved the best discriminatory performance. Moreover, preliminary results suggests that such method is more robust to phishing attacks.

**Keywords:** Anti-Spam filtering, Text Mining, Naive Bayes, Support Vector Machines

## 1 Introduction

Email is a commonly used service for communication and information sharing. However, unsolicited e-mail (spam) emerged very quickly after email itself and currently accounts for 89% to 92% of all email messages sent [11]. The cost of sending these emails is very close to zero, since criminal organizations have access to millions of infected computers (known as botnets) [15]. Spam consumes resources, such as time spent reading unwanted messages, bandwidth, CPU and disk [6]. Also, spam is an intrusion of privacy and used to spread malicious content (e.g. phishing attacks, online fraud or viruses).

The majority of the current anti-spam solutions are based on [3]: Content-Based Filtering (CBF) and Collaborative Filtering (CF). CBF is the most popular anti-spam approach, using message features (e.g. word frequencies) and Data

Mining (DM) algorithms (e.g. Naive Bayes) to discriminate between legitimate (ham) and spam messages. CF works by sharing information about spam messages. One common CF variant is the DNS-based Blackhole List (DNSBL), also known as blacklist, which contains known IP addresses used by spammers. CF and CBF can also be combined. For example, a blacklist is often used at a server level to tag a large number of spam. The remaining spam can be detected later by using a personalized CBF at the client level (e.g. Thunderbird SpamBayes, http://www.entrian.com/sbwiki).

Spam content is very easy to forge in order to confuse CBF filters. For example, normal words can be mixed into spam messages and this heavily reduces the CBF performance [13]. In contrast, spammers have far less flexibility in changing network-level features. Yet, the majority of the spam research gives attention to content and the number of studies that address network-level properties is scarce. In 2005, Leiba et al. [9] proposed a reputation learning algorithm that is based on the network path (from sender to receiver) of the message. Such algorithm obtained a high accuracy when combined with a CBF bayesian filter. Ramachandran and Feamster [15] have shown that there are spam/ham differences for several network-level characteristics (e.g. IP address space), although the authors did not test these characteristics to filter spam using DM algorithms. More recently, transport-level properties (e.g. TCP packet stream) were used to classify spam messages, attaining a classification accuracy higher than 90% [1].

In this paper, we explore network-level characteristics to discriminate spam (see Section 2.1). We use some of the features suggested in [15] (e.g. operating system of sender) and we also propose new properties, such as the IP geographic coordinates of the sender, which have the advantage of aggregating several IPs. Moreover, in contrast with previous studies (e.g. [9, 1]), we collected emails from two countries (U.S. and Portugal) and tested two classifiers: Naive Bayes and Support Vector Machines (Section 2.2). Furthermore, our approach is compared with eight DNSBLs and CBF models (i.e. bag-of-words) and we show that our strategy is more robust to phishing attacks (Section 3).

## 2 Materials and Methods

### 2.1 Spam Telescope Data

To collect the data, we designed and developed the spam telescope repository. The aim of this repository is to perform a longitudinal and controlled study by gathering a significant slice of the world spam traffic. Spam was harvested by setting several spam traps; i.e. fake emails that were advertised through the Internet (e.g. Web pages). To collect ham, we created email addresses what were inscribed in moderated mailing lists with distinct topics. Figure 1 shows the proportions of mailing list topics that were used in our datasets. For both spam and ham collection, we tried to mimic real users from two countries: U.S. and Portugal (PT). For instance, we first registered a U.S. domain (.com) and then set the corresponding Domain Name System (DNS) Mail Exchange (MX) record. Next, the USA spam traps were advertised in USA popular Web sites and the

USA ham emails were inscribed in 12 USA mailing lists. A similar procedure was taken to harvest the Portuguese messages (e.g. .pt domain).

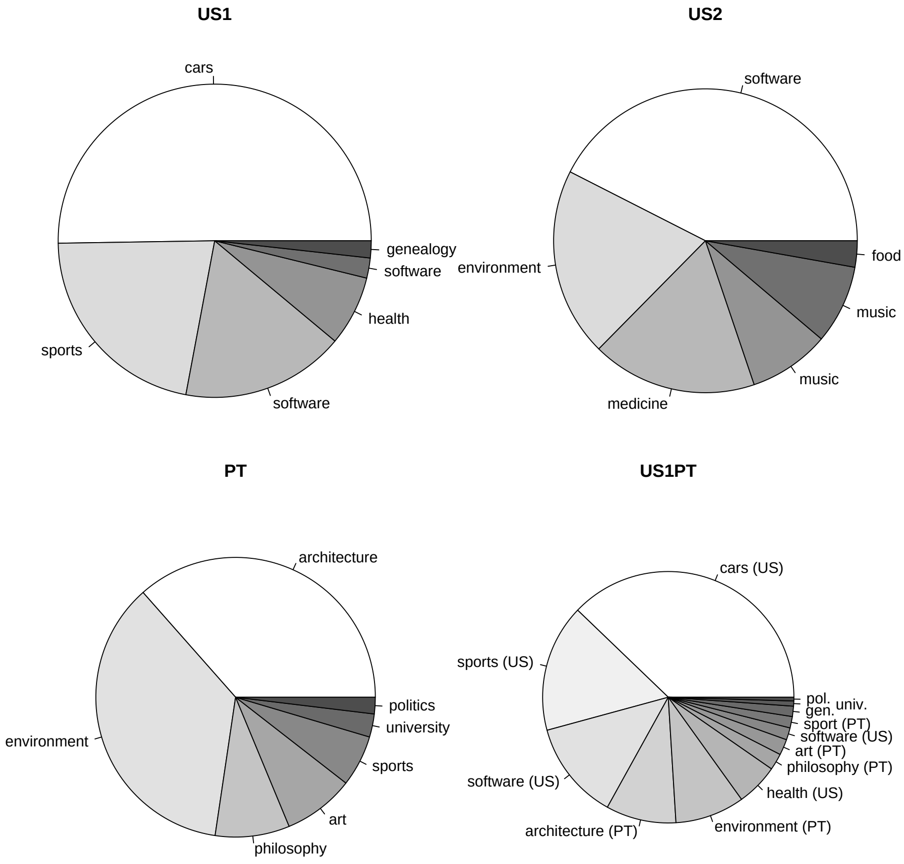

**Fig. 1.** Pie charts showing the distribution of mailing list topics for each dataset

All spam telescope messages were gathered at a specially crafted server. This server uses virtual hosting to redirect addresses from distinct Internet domains and runs a customized Simple Mail Transfer Protocol (SMTP) program called Mail Avenger (http://www.mailavenger.org). We set Mail Avenger to tag each received message with the following information:

- IP address of the sender and a traceroute to this IP;
- Operating System (OS) of the sender, as estimated from a passive p0f TCP fingerprint;

– lookup at eight DNSBLs: cbl.abuseat.org (B1), dnsbl.sorbs.net (B2), bl.-spamcop.net (B3), sbl-xbl.spamhaus.org (B4), dul.dnsbl.sorbs.net (B5), zen.-spamhaus.org (B6), psbl.surriel.com (B7) and blackholes.five-ten-sg.com (B8).

The four network-level properties used in this study are presented in Table 1. Instead of using direct IP addresses, we opt for geographic coordinates (i.e. latitude and longitude), as collected by querying the free http://ipinfodb.com database. The geographic features have the advantage of aggregating several IPs. Also, it it known that a large fraction of spam comes from specific regions (e.g. Asia) [15]. The NHOP is a distance measure that was computed using the traceroute command. The passive OS signatures were encoded into four classes: windows – if from the MS family (e.g. windows 2000); linux (if a linux kernel is used); other (e.g. Mac, freebsd, openbsd, solaris); and unknown (if not detected).

**Table 1.** Network-level attributes

| **Attribute** | **Domain** |
|---|---|
| NHOP – number of hops/routers to sender | {8,9,. . .,65} |
| Lat. – latitude of the IP of sender | [-42.92°,68.97°] |
| Long. – longitude of the IP of sender | [-168.10°,178.40°] |
| OS – operating system of sender | {windows,linux,other,unknown} |

In this study, we collected recent data, from April 21st April to November 9th 2009. Five datasets were created in order to mimic distinct and realistic user profiles (Table 2). The US1 set uses ham from 6 mailing lists whose members are mostly U.S. based, while US2 contains ham from different U.S. lists and that is more spread through the five continents (Figure 2). Regarding the spam, the data collected from the U.S. traps was added into US1 and US2, while PT includes only features extracted from Portuguese traps. The mixture of ham and spam was based on the time that each message was received (date field), which we believe is more realistic than the sampling procedure adopted in [12]. Given the large number of experiments addressed in this work, for US1, US2 and PT we opted to fix the global spam/ham ratio at 1. Yet, it should be noted that the spam/ham ratios fluctuate through time (right of Figure 4). The fourth set (US1PT) merges the data from US1 and PT, with the intention of representing a bilingual user (e.g. Portuguese but working in U.S.). Finally, the U.S. Without Blacklist Spam (USWBS) contains ham and spam from US2. The aim is to mimic a hybrid blacklist-filter scenario, thus all spam that was detected by any of the eight DNSBLs was removed from US2. For this last set, we set the spam/ham ratio to a realistic 0.2 value, since in such scenario most spam should be previously detected by the DNSBLs. Figures 2, 3 and 4 show several examples of ham/spam differences when analyzing the network-level attributes.

**Table 2.** Summary of the Spam Telescope corpora

| **setup** | **ham main** | **#mailing** | **#ham** | **total** | **spam** | **time** |
|---|---|---|---|---|---|---|
| | **language** | **lists** | **senders** | **size** | **/ham** | **period** |
| US1 | English | 6 | 343 | 3184 | 1.0 | [23/Apr./09,9/Nov./09] |
| US2 | English | 6 | 506 | 3364 | 1.0 | [21/Apr./09,9/Nov./09] |
| PT | Portuguese | 7 | 230 | 1046 | 1.0 | [21/May/09,9/Nov./09] |
| US1PT | Eng./Port. | 13 | 573 | 4230 | 1.0 | [23/Apr./09,9/Nov./09] |
| USWBS | English | 6 | 257 | 612 | 0.2 | [21/Apr./09,9/Nov./09] |

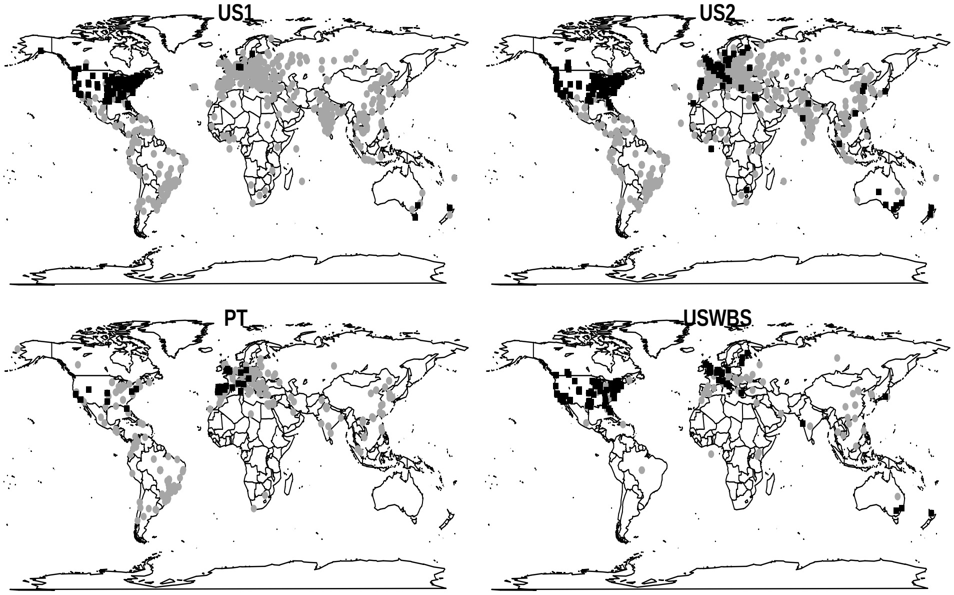

**Fig. 2.** Distribution of geographic IP of sender (black squares denote ham, gray circles show spam) for the used datasets

## 2.2 Spam Filtering Methods

We adopted two DM classifiers, Naive Bayes (NB) and Support Vector Machine (SVM), using the R statistical tool [14] (e1071 and kernlab packages) [14]. The NB algorithm is widely adopted by anti-spam filtering tools [3]. It computes the probability that an email message $j \in \{1, \ldots, N\}$ is spam (class $s$) for a filter trained over $\mathcal{D}$ data with $N$ examples:

$$p(s|\mathbf{x}_j) = \beta \cdot p(s) \prod_{i=1}^{m} p(x_i|s) \tag{1}$$

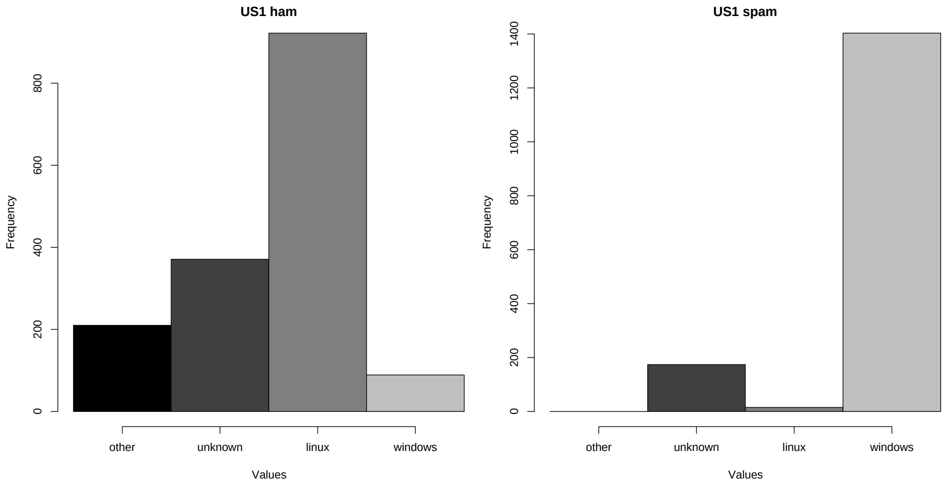

**Fig. 3.** Operating system histograms for the US1 dataset (left ham, right spam)

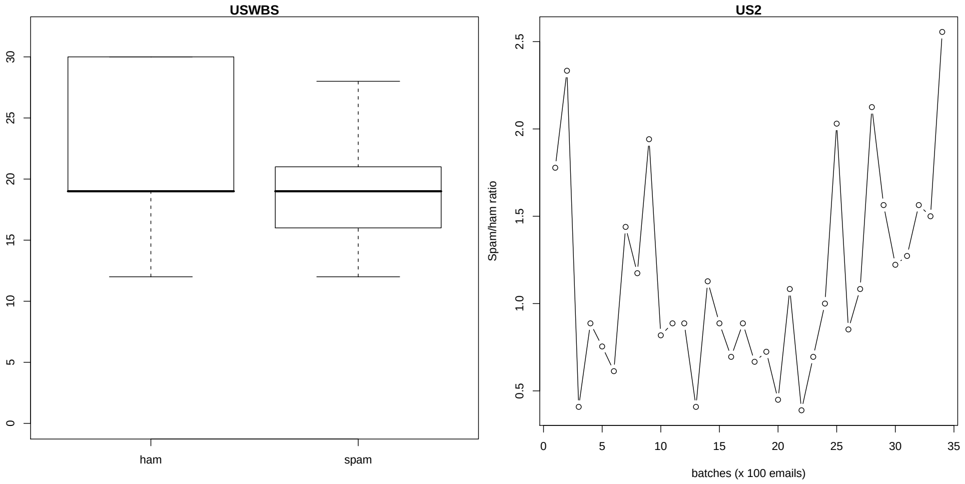

**Fig. 4.** NHOP ham/spam box plots for the USWBS dataset (minimum, median and maximum values, left) and spam/ham ratio evolution for the US2 dataset (right)

where $\beta$ is normalization constant that ensures that $p(s|\mathbf{x}) + p(\neg s|\mathbf{x}) = 1$, $p(s)$ is the spam frequency of dataset $\mathcal{D}$ and $x_i$ denotes the input feature $i \in \{1, \ldots, m\}$. The $p(x_i|s)$ estimation depends on the NB version. We used the multi-variate Gauss NB that is implemented in the R tool [12]:

$$p(x_i|c) = \frac{1}{\sigma_{i,c}\sqrt{2\pi}} \exp - \frac{(x_{ij} - \mu_{i,c})^2}{2\sigma_{i,c}^2} \qquad (2)$$

where it is assumed each attribute ($x_i$) follows a normal distribution for each $c = s$ or $c = \neg s$ categories and the mean ($\mu_{i,c}$) and typical deviation ($\sigma_{i,c}$) are estimated from $\mathcal{D}$.

The Support Vector Machine (SVM) is a more powerful and flexible learner, capable of complex nonlinear mappings and was recently considered one of the most influential DM algorithms [16]. The basic idea is transform the input $\mathbf{x}_j \in \Re^m$ into a high $f$-dimensional feature space by using a nonlinear mapping. Then, the SVM finds the best linear separating hyperplane, related to a set of support vector points, in the feature space. The transformation depends on a nonlinear mapping that does not need to be explicitly known but that depends of a kernel function. We opted for the popular gaussian kernel, which presents less parameters and numerical difficulties than other kernels (e.g. polynomial):

$$K(\mathbf{x}_j, \mathbf{x}'_j) = exp(-\gamma||\mathbf{x}_j - \mathbf{x}'_j||^2),\ \gamma > 0 \tag{3}$$

The probabilistic output SVM computes [10]:

$$\begin{array}{r}f(\mathbf{x}_j) = \sum_{p \in SV} y_p \alpha_p K(\mathbf{x}_p, \mathbf{x}_j) + b \\ p(s|\mathbf{x}_j) = 1/(1 + exp(Af(\mathbf{x}_j) + B))\end{array} \tag{4}$$

where $SV$ is the set of support vectors, $y_j \in \{-1, 1\}$ is the output for message $j$ (if spam $y_j$=1, else $y_j = -1$), $b$ and $\alpha_p$ are coefficients of the model, and $A$ and $B$ are determined by solving a regularized maximum likelihood problem. Under this setup, the SVM performance is affected by two parameters: $\gamma$, the parameter of the kernel, and $C$, a penalty parameter. Since the search space for these parameters is high, we heuristically set the least relevant parameter to $C = 3$ [4]. For NSV, $\gamma$ is set using a grid search (i.e. $\gamma \in \{2^{-15}, 2^{-13}, \ldots, 2^3\}$). During this search, the training data was further split into training (first 2/3 of $\mathcal{D}$) and validation sets (last 1/3). Then, the best $\gamma$ (i.e. with the highest AUC in the validation set) was selected and the model was retrained with all $\mathcal{D}$ data. Since the WSV model requires much more computation (with up to 3000 features when compared with the 4 NSV inputs), for this model we set $\gamma = 2^{-3}$.

DM models such as NB and SVM are harder to interpret when compared with simpler methods (e.g multiple regression). Still, it is possible to extract knowledge in terms of input relevance by using a sensitivity analysis procedure [5]. This procedure is applied after the training phase and analyzes the model responses when the inputs are changed. Let $p(s|\mathbf{x}(l))$ denote the output obtained by holding all input variables at their average values except $x_a$, which varies through its entire range with $l \in \{1, \ldots, L\}$ levels. If a given input variable ($x_a \in \{x_1, \ldots, x_m\}$) is relevant then it should produce a high variance ($V_a$). Thus, its relative importance ($R_a$) can be given by:

$$\begin{array}{l}V_a = \sum_{l=1}^{L} (p(s|\mathbf{x}(l)) - \overline{p(s|\mathbf{x}(l))})^2/(L-1) \\ R_a = V_a / \sum_{i=1}^{m} V_i \times 100\ (\%)\end{array} \tag{5}$$

In this work, we propose novel filters based on network-level inputs and compare these with bag-of-words models and blacklists. For the first two classes of filters, we tested both NB and SVM algorithms using either network based attributes or word frequencies. The complete set of models includes:

- NNB and NSV, NB and SVM classifiers using the four inputs from Table 1;
- WNB and WSV, NB and SVM using word frequencies;
- Eight blacklist based models (B1,...,B8), where spam probabilities are set to $p(s|\mathbf{x}_j) = 1$ if the IP is present in the corresponding DNSBL, else it is 0;
- finally, the All Blacklist (AB) method that outputs $p(s|\mathbf{x}_j) = 1$ if any of the eight DNSBLs was activated, otherwise it returns 0.

Regarding the bag-or-words models (WNB and WSV), we used the preprocessing adopted in [6]. First, all attachments are removed. In the case of ham, all mailing list signatures are also deleted. Then, word frequencies are extracted from the subject and body message (with the HTML tags previously removed). Next, we apply a feature selection that is based in ignoring any words whose frequency is lower than 5 in the training set ($\mathcal{D}$) and then selecting up to the 3000 most relevant words according to a mutual information criterion. Finally, we apply a TF-IDF and length normalization transform to the word frequencies. All preprocessing was performed using the perl [2] and R languages [14].

### 2.3 Evaluation

To access the predictive performances, we adopted the realistic incremental retraining evaluation procedure, where a mailbox is split into batches $b_1, \ldots, b_n$ of $k$ adjacent messages ($|b_n|$ may be less than $k$) [12]. For $i \in \{1, \ldots, n-1\}$, the filter is trained with $\mathcal{D} = b_1 \cup \ldots \cup b_i$ and tested with the messages from $b_{i+1}$.

For a given probabilistic filter, the predicted class is given by: $s$ if $p(s|\mathbf{x}_j) > D$, where $D \in [0.0, 1.0]$ is a decision threshold. For a given $D$ and test set, it is possible to compute the true ($TPR$) and false ($FPR$) positive rates:

$$\begin{array}{l} TPR = TP/(TP+FN) \\ FPR = FP/(FP+TN) \end{array} \quad (6)$$

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) [7]. 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 should be close to 1.0. With the incremental retraining procedure, one $ROC$ is computed for each $b_{i+1}$ batch and the overall results are presented by adopting the vertical averaging $ROC$ (i.e. according to the $FPR$ axis) algorithm presented in [7]. Statistical confidence is given by the t-student test [8].

## 3 Experiments and Results

We tested all methods from Section 2.2 in all datasets from Table 2 and using a retraining evaluation with a batch size of $k = 100$ (a reasonable value also adopted in [12]). The obtained results are summarized as the mean of all test sets ($b_{i+1}$, $i \in \{1, \ldots, n-1\}$), with the respective 95% confidence intervals and shown in Tables 3 and 4. To increase clarity, we only show the best blacklist (B6) in Table 3. In the tables, the best values are in **bold**, while underline denotes a statistical significance (i.e. p-value<0.05). In Table 3, the significance was computed for a paired t-test comparison of the network-level approach against AB and the corresponding bag-of-words method (e.g. NSV vs AB and WSV). In Table 4, the paired t-test is performed against the second best blacklist (B4).

Under the AUC metric and for all setups, the NSV method is the best choice and the obtained results are of high quality (from 95.3% to 99.8%). The NNB is the second best filter for the last three datasets. It is also interesting to notice that both NSV and NNB are robust to a geographic spread of the ham origin, since there is only a slight decrease (0.4 and 0.8 pp) when comparing US2 and US1 filtering performances. For WSV, the detection capability is higher when there is Portuguese ham (PT and US1PT). This was an expected behavior, since most spam is written in English. The bag-of-words performances decrease substantially for the last setup, showing that the spam that is not detected in blacklists is more difficult to classify based on content. However, our network-level based methods still obtained high AUC values, around 95%. When using the same inputs, the SVM algorithm is always better when compared with NB, with an average improvement of 1.2 pp for the network-level features and 9.7 pp for the bag-of-words attributes.

**Table 3.** Comparison among the main filters (AUC test set results, in %)

| **setup** | **B6** | **AB** | **WNB** | **WSV** | **NNB** | **NSV** |
|---|---|---|---|---|---|---|
| US1 | 98.0±0.8 | 98.9±0.5 | 73.0±4.7 | 75.8±2.2 | 98.7±0.6 | **99.8**±0.2 |
| US2 | 98.1±0.7 | 98.9±0.6 | 65.4±2.7 | 77.0±2.3 | 97.9±0.8 | **99.4**±0.5 |
| PT | 83.9±4.5 | 89.0±3.4 | 71.4±7.5 | 82.1±5.1 | 95.6±1.5 | **97.3**±1.6 |
| US1PT | 94.5±0.9 | 96.3±0.8 | 68.4±3.4 | 78.2±2.0 | 98.2±0.5 | **99.2**±0.4 |
| USWBS | 50.0±0.0 | 50.0±0.0 | 50.1±0.3 | 63.6±7.6 | 94.7±3.4 | **95.3**±3.6 |

**Table 4.** Blacklist filter performances (AUC test set results, in %)

| **setup** | **B1** | **B2** | **B3** | **B4** | **B5** | **B6** | **B7** | **B8** |
|---|---|---|---|---|---|---|---|---|
| US1 | 87.6±1.2 | 80.2±1.8 | 80.8±1.6 | 87.7±1.2 | 58.7±1.1 | **98.0**±0.8 | 74.3±2.6 | 67.1±2.6 |
| US2 | 88.7±1.2 | 80.3±2.0 | 79.4±2.0 | 88.9±1.2 | 59.2±1.0 | **98.1**±0.7 | 74.0±2.6 | 67.5±3.0 |
| PT | 76.0±3.8 | 74.7±2.1 | 69.1±3.7 | 78.0±4.2 | 59.0±3.0 | **83.9**±4.5 | 65.5±3.0 | 63.0±4.5 |
| US1PT | 84.5±1.0 | 79.0±1.5 | 77.2±1.1 | 85.2±0.9 | 58.7±1.1 | **94.5**±0.9 | 72.2±1.8 | 66.2±2.0 |

Regarding the blacklist comparison (Table 4), B6 is clearly the best filter. Overall, the second best DNSBL is B4, followed by B1. For all setups, three blacklists (B5, B7 and B8) are outperformed by the WSV model. B5 is the worst filter, with no average AUC value above 60%. For all DNSBLs except B5, the worst performance is achieved for the Portuguese dataset (PT). This outcome was expected, since the tested blacklists are international and thus may fail in mapping more country specific spam.

The full ROC analysis is given in Figure 5. To increase clarity, we only selected the best and worst blacklists (B6 and B5). The ROC curve allows the definition of different filtering profiles, according to the user needs. In the studied datasets, the blacklists never output a false positive. Thus, for B6 and AB, the TPR values are high when FPR is zero. For the spam domain, this is an important point of the ROC curve, since often the cost of losing normal e-mail ($FP$) is much higher than receiving spam ($FN$). This is particularly true if the email client action is set to delete messages marked as spam. For this decision point, AB, followed by B6, are the best filters, except for US1 and USWBS, where NSV is the best option. For larger admissible values of FPR, NSV gives the best TPR values. It should be noted that for some users, this is an interesting scenario, as the cost of receiving spam can also be high, due to an higher vulnerability to phishing attacks, viruses or online fraud, while not all ham is important. Since often email clients move messages marked as spam to a different folder, false positives could still be read by the user.

The average network-level feature importances for NSV are plotted in Figure 6. The bar plots show the $\overline{R_a}$ values, while the whiskers denote the 95% confidence intervals. The US1 importance bars are not shown, since they are similar to US2. All four attributes contribute to the model, although their relative influences vary. For example, the operating system (OS) is the most relevant feature for the US datasets, although it is the least important input for PT. On the other hand, the length of the message path (NHOP) is most relevant attribute for PT and US1PT.

To study the filtering vulnerability to phishing email attacks, we searched within the datasets for spam messages asking for user password details (e.g. related to a bank online account). Five messages were found and the respective spam probability predictions ($p(s|\mathbf{x}_j)$) are shown in Table 5. The first column of the table shows the dataset that contained such messages. Although the number of examples is not enough for a more definitive conclusion, the results seem to favor the network-level based methods. For a decision threshold of $D = 0.5$, NSV detects all attacks, while NNB predicts four. The less robust methods are B6 and WSV.

## 4 Conclusions

In this work, we proposed a new spam filtering approach that is based on four network-level attributes: message path length in terms of number of routers (NHOP), geographic coordinates (i.e. latitude and longitude) and operating sys-

**Table 5.** Filter responses to phishing messages (values above 0.5 are in **bold**)

| setup | B6 | AB | WNB | WSV | NNB | NSV |
|---|---|---|---|---|---|---|
| US1 | 0.00 | **1.00** | 0.00 | **0.62** | **1.00** | **0.99** |
| US1 | 0.00 | **1.00** | 0.00 | 0.36 | **1.00** | **0.98** |
| US2 | **1.00** | **1.00** | **1.00** | 0.28 | **1.00** | **1.00** |
| PT | 0.00 | 0.00 | **1.00** | 0.35 | **1.00** | **0.91** |
| US1PT | 0.00 | 0.00 | **1.00** | 0.29 | 0.00 | **0.96** |

tem of the sender. We tested two data mining (DM) classifiers, Naive Bayes (NB) and Support Vector Machines (SVM) and also targeted two countries from different continents and with different main languages (i.e. U.S. and Portugal). Since our network-level properties are not currently monitored by filtering systems, we created and developed a new spam repository, called spam telescope. This repository includes real legitimate (ham) and non legitimate (spam) messages. The ham was collected from several mailing lists, while the spam was captured from email traps (fake addresses advertised through the Web). Several experiments were carried out, where a realistic mixture of spam and ham was used to simulate distinct user profiles.

When comparing with Content-Based filters (CBF), i.e. bag-of-words, and eight DNS-based Blackhole Lists (DNSBL), the NSV method (SVM fed with the four network-level features) obtained the best discriminatory performance, with high quality results (from 95.3% to 99.8%). The NSV method requires much less computation than the respective bag-of-words filter. Also, in contrast with the blacklist methods, it does not require communication with other servers, since the free geographic IP database that we used can be installed locally. Moreover, preliminary results suggest that NSV is more robust to phishing email attacks.

Based on the achieved results, we advise the use of the NSV filter, which provides a high true positive rate (i.e. detects most of the spam). To reduce false positives (i.e. ham marked as spam), this method could be used after a first phase blacklist filtering. Yet, for an effective blacklisting, it should be considered a careful DNSBL server selection or (even better) use of multiple DNSBLs.

Spammers and anti-spammers are in a continuous struggle. The research community has devoted a large attention to improve CBF. Yet, as argued in [15], spammers can easily change content to confuse CBF filters but network-level properties are more persistent in time. For example, a large portion of current spam comes from botnets. Most spammers are greedy and want a massive distribution of spam, thus they do not care about the location of a given controlled machine. Furthermore, some operating systems (e.g. Windows) are more vulnerable to botnet control by malicious software. Hence, we believe it is more difficult for spammers to surpass network-level based filters. As future work, we intend to enlarge the experiments to other countries (e.g. Spain). Also, we wish to deploy the proposed models in real email clients (e.g. Thunderbird) to gather more feedback.

## Acknowledgments

This work is supported by FCT grant PTDC/EIA/64541/2006. We also wish to thank David Manzieres for supporting Mail Avenger.

## References

1. R. Beverly and K. Sollins. Exploiting transport-level characteristics of spam. In *5th Conference on Email and Anti-Spam (CEAS)*, 2008.
2. R. Bilisoly. *Practical text mining with Perl.* Wiley Publishing, 2008.
3. E. Blanzieri and A. Bryl. A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review*, 29(1):63–92, 2008.
4. V. Cherkassy and Y. Ma. Practical Selection of SVM Parameters and Noise Estimation for SVM Regression. *Neural Networks*, 17(1):113–126, 2004.
5. P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.
6. P. Cortez, C. Lopes, P. Sousa, M. Rocha, and M. Rio. Symbiotic Data Mining for Personalized Spam Filtering. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI-09)*, pages 149–156. IEEE, 2009.
7. T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27:861–874, 2006.
8. A. Flexer. 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, 1996.
9. B. Leiba, J. Ossher, VT Rajan, R. Segal, and M. Wegman. SMTP path analysis. In *Proceedings of the Second Conference on E-mail and Anti-Spam (CEAS)*, 2005.
10. H.T. Lin, C.J. Lin, and R.C. Weng. A note on Platts probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276, 2007.
11. MAAWG. Email Metrics Program: The Network Operators' Perspective. Report #10 – third and fourth quarter 2008, Messaging Anti-Abuse Working Group, S. Francisco CA, USA, March 2009.
12. V. Metsis, I. Androutsopoulos, and G. Paliouras. Spam Filtering with Naive Bayes – Which Naive Bayes? In *Third Conference on Email and Anti-Spam (CEAS)*, 2006.
13. B. Nelson, M. Barreno, F. Chi, A. Joseph, B. Rubinstein, U. Saini, C. Sutton, J. Tygar, and K. Xia. Exploiting Machine Learning to Subvert Your Spam Filter. In *1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, pages 1–9. ACM Press, 2008.
14. R Development Core Team. *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, 2009.
15. A. Ramachandran and N. Feamster. Understanding the Network-Level Behavior of Spammers. In ACM, editor, *SIGCOMM'06*, pages 291–302, 2006.
16. X. Wu, V. Kumar, J. Quinlan, J. Gosh, Q. Yang, H. Motoda, G. MacLachlan, A. Ng, B. Liu, P. Yu, Z. Zhou, M. Steinbach, D. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

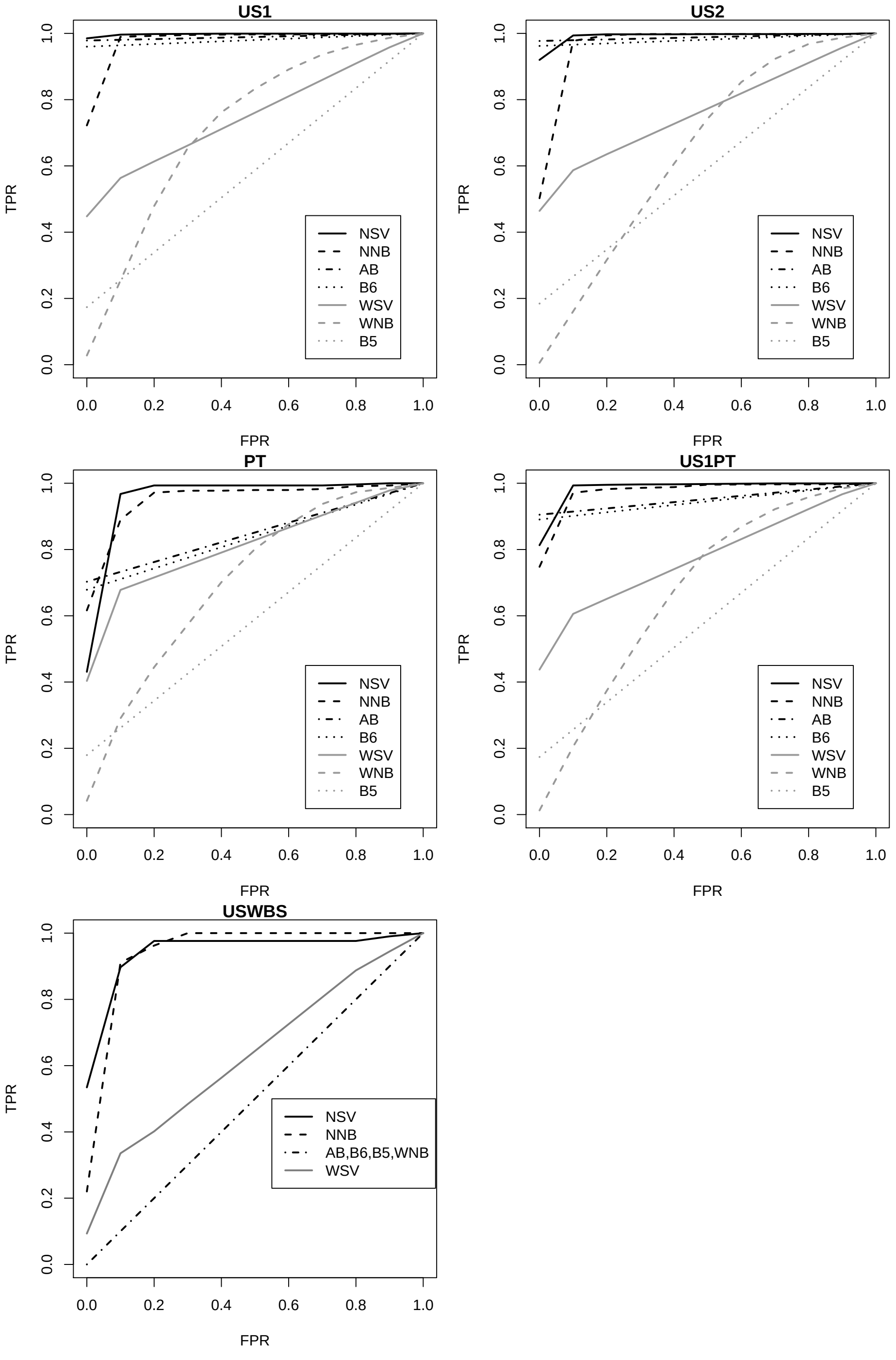

**Fig. 5.** Average test set ROC curves

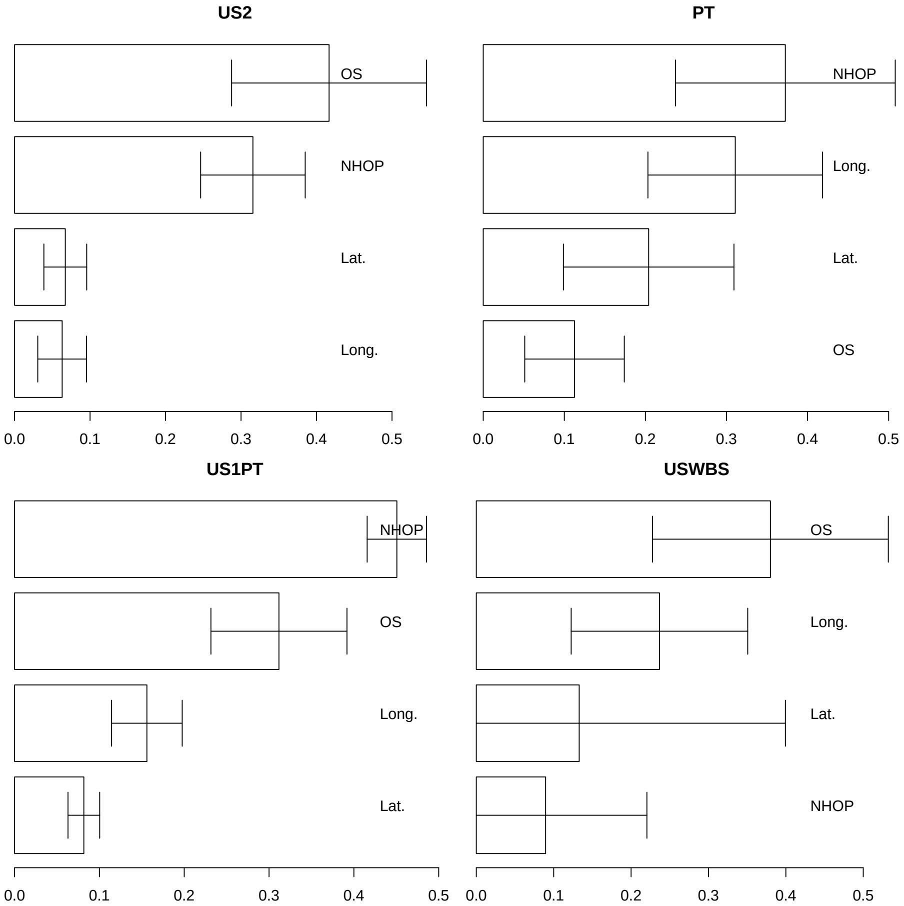

**Fig. 6.** Average input importances for the NSV model