

A Real-Life Study in Phishing Detection

André Bergholz
Gerhard Paaß
Fraunhofer IAIS
<firstname.lastname>
@iais.fraunhofer.de

Luigi D'Addona
Domenico Dato
Tiscali Italia
ldaddona@tiscali.com
ddato@tiscali.com

ABSTRACT

Phishing is a serious threat to global security and economy. Previously we have developed a phishing filtering system based on automatic classification. We perform statistical filtering of emails, where a classifier is trained on characteristic features of existing emails and subsequently is able to identify new phishing emails with different contents. In this work we test our developed system in a real-life environment at a commercial ISP. The system is applied to an unskewed real-life stream consisting of thousands of emails every day. We use active learning to keep the system's model up-to-date. The experiments show that the system performs very well as a filter even in the presence of many spam emails. We furthermore demonstrate that active learning is indeed useful and leads to better results than using a fixed model. Last, we integrate the output of another spam filter into the system and show that this combined filter leads to better results than either filter by itself.

1. INTRODUCTION

Phishing is a serious threat to global security and economy. Criminals are trying to convince unsuspecting online users to reveal sensitive information, e.g., passwords, account numbers, social security numbers or other personal information. Most phishing emails aim at withdrawing money from financial institutions or getting access to private data. There are a number of possible countermeasures to phishing. These range from communication-oriented approaches like authentication protocols over blacklisting to content-based filtering approaches.

The goal of the AntiPhish project is to develop reliable anticipatory phishing email filters [1]. In the scope of the project one goal is to test the developed AntiPhish filter system in a real-life industrial setting. Compared to the offline experiments performed during the first phases of the project the real-life deployment poses a number of challenges:

- The data comes from the present real-life stream.
- More non-English data is present as the experiments are conducted in Italy.
- The data is (almost) unskewed.
- All data is unlabeled, spam cannot easily be eliminated beforehand.

CEAS 2010 - Seventh annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference July 13-14, 2010, Redmond, Washington, US

- Privacy must be respected.

Here we report on the comprehensive study we performed in October and November 2008 at Tiscali Italia. Our main experimental setting was a continuous evaluation of AntiPhish filter system during a period of 20 days. We performed active learning and updated the model once per day. Our main goal was to separate ham email from non-ham emails, i.e., phishing and spam emails combined. We evaluated AntiPhish filter system both as a stand-alone classifier and as an add-on to commercial spam filters. We conducted experiments on the tuning of AntiPhish filter system, the effect of active learning, and separate phishing classification. As this is a real-life study, we can obviously not go into details about reasons for misclassification.

This report is organized as follows: After surveying related work in Section 2 we briefly summarize the AntiPhish filter system in Section 3. In 4 we describe the experimental setup and the used data. Section 5 gives the results of all performed experiments. We close with conclusions in Section 6.

2. RELATED WORK

The filtering of emails naturally can be defined as a classification task: first relevant features x (i.e., words, format or header information, etc.) are extracted from an email. Then a classifier function $f : x \rightarrow y$ is determined where the class variable y can take the values “phishing”, “spam”, or “legitimate”. A large number of experiments were reported where traditional classifiers were applied to this task [13, 6, 5, 2, 1]. Usually the support vector machine is used as classifier, which is able to process very large numbers of features. Using advanced features and careful parameter tuning these approaches are able to filter a given set of emails with an F-value close to 99%.

For a real-life application, however, the filtering system has to be able to adapt to new type of phishing emails. This requires the regular check of the filter performance using a small number of manually labeled emails. As classifiers are inductively trained from examples the classification is not correct with certainty but only with some probability. For many classifiers it is possible to estimate for each individually classified email the probability, that the proposed classification is correct.

In “active learning” these borderline cases with intermediate probability are labeled to increase the training set and improve the filter [8]. Classifiers can be adapted to emails with new characteristics with a minimum number of additional annotations [7]. Sculley presents an online version,

where a new emails from a stream can be selected for labeling just after it is classified, the objective being the maximization of the classification performance while minimizing the number of labels requested [10]. Three hyperplane-based classifiers, namely perceptron, perceptron with margins and relaxed online SVM were studied, along with three methods for deciding to request the label [11]. It is shown that online active learning can dramatically reduce labeling and training costs with negligible additional overhead while maintaining high levels of classification performance.

3. EXPERIMENTAL METHODOLOGY

In this section we briefly summarize the methodology used in this real-life deployment [2, 1]. We use a machine learning approach, i.e., we train an automatic classifier. In contrast to manually constructed filter rules this classifier automatically assesses the relevance of input features $x = (x_1, \dots, x_m)$ (e.g., email characteristics) and establishes a function to determine the desired classification y (e.g., phishing or non-phishing).

3.1 Features for Email Data

To train the classifier appropriate features must be extracted from the emails. We group the features into three groups: basic features, dynamic Markov chain features, and latent topic model features. After providing these features we furthermore perform feature processing (scaling and normalization) and feature selection (i.e., automatic elimination of the most irrelevant features).

3.1.1 Basic Features

Basic features are features that can directly be extracted from the email without much further processing. Note that we only use features that can be derived directly from the email itself. In particular, we do not use features that require information about specific linked-to websites. We use the following basic features:

- Structural features: Reflect the body part structure of an email, i.e., information about the number of body parts, discrete and composite body parts, and alternative body parts.
- Link features: Reflect various properties of links contained in an email, i.e., information about the total number of links, internal and external links, links with IP-numbers, deceptive links (links where the URL visible to the user is different from the URL the link is pointing to), links behind an image, etc.
- Element features: Reflect what kinds of web technologies are used in an email, i.e., information about whether HTML, scripting and in particular JavaScript, and forms are used.
- Word list features: We use a positive word list, i.e., a list of words hinting at the possibility of phishing.

3.1.2 Dynamic Markov Chain Features

Dynamic Markov Chain (DMC) features are based on information theory and capture the likelihood of a message belonging to a specific class. We extract these likelihoods as well as class membership indicators as features for our classification system.

The dynamic Markov chain generation is a technique developed for arithmetic compression, the problem of compressing arbitrary binary sequences [4]. A sequence is thought to be generated by a random source. This source can be approximated by a dynamically constructed Markov chain. Cormack et al. developed a technique for the incremental construction of a Markov chain [4]. These dynamic Markov chains have been successfully applied to text classification problems, e.g., in [9]. Each class is considered as a different source, and each text belonging to the class is treated as a message emitted from the corresponding source. The source is approximated by incrementally enhancing the initial starting chain. Given a sufficiently large number of training examples the iterative approximation of the unknown source permits the accurate estimation of the likelihood that a given sequence originated from that source. By comparing these likelihoods for different sources the sequence may be classified.

For email classification we build two models, one for ham emails and one for phishing emails. We extract four DMC features: the scores for an email for each of the two models and two Boolean features indicating membership in either class.

3.1.3 Latent Topic Model Features

Semantic features are content-based indicators from email messages and are extracted in a data-driven way using latent topic models. These latent topics are clusters of words that tend to appear together in emails. We can expect that in a phishing email the words click and account often appear together, while in regular financial emails the words market, prices and plan may co-occur. Latent topic models exploit the co-occurrence of words in a training set of emails.

Common latent topic models do not take into account different classes of documents, e.g., phishing or nonphishing. We developed a new statistical model, the latent Class-Topic Model (CLTOM) [2]. CLTOM is an extension of latent Dirichlet allocation (LDA) in such a way that it incorporates category information of emails during the model inference [3]. It yields word clusters that are more focused to the distinction of phishing emails from legitimate emails.

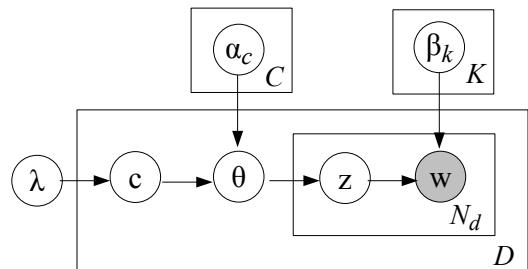


Figure 1: The Graphical Model of CLTOM

Figure 1 shows the graphical model of CLTOM. When a new message arrives, the semantic features for the message are derived through an inference in the trained CLTOM. Finally, the posterior mean value is provided as the semantic feature for the incoming message.

3.2 Initial Study

To prepare for this set of experiments an initial study was conducted in August and September 2008. This study

allowed to get an impression on how the AntiPhish filter system performs, which features are most relevant and what potential pitfalls may appear. The initial study led to many promising results. The three most important results were:

1. The inclusion of spam did not strongly hurt the performance.
2. The inclusion of many Italian-language emails did not pose a problem.
3. At a low false positive rate the AntiPhish filter system can filter out a large number of both spam and phishing emails and act as an add-on to existing spam filters.

Because the AntiPhish system was originally developed to detect phishing emails we expected the performance to be lower when spam emails are included. Of course, in a real-life environment spam emails cannot be completely eliminated beforehand. However, the study showed that the results were not much worse than our previous benchmark results as reported in [2]. Furthermore, the system was designed to deal primarily with English-language emails, so we feared that the inclusion of many Italian-language emails would hurt the performance. However, not too many of the used features are language-dependent. In fact, only the word list is. As for the Dynamic Markov chains we trained separate automaton for English, Italian, and other languages using an open-source language detection software.

4. EXPERIMENTAL SETUP

The AntiPhish filter system is deployed at Tiscali Italia as an actively learning classifier that recognizes the moving target of unwanted (non-ham) emails. The outline of the general deployment procedure is as follows:

1. In the beginning an initial model M_0 based on an initial labeled training set and a well-defined feature set is learned.
2. For the deployment period of n (in our case $n = 20$) days the following procedure is executed once for every day $t \in \{1, \dots, n\}$:
 - (a) A set of emails S_t is captured and sent in real-time through existing spam filters.
 - (b) The initial model M_0 is used to determine a test subset of $T_t \subseteq S_t$ that will be used for evaluation of the current model M_{t-1} . Note that the test set must always be determined by the exact same criterion and must not depend on the current model.
 - (c) The current model M_{t-1} , trained on the previous day and on the very first day set to M_0 , is used to determine a subset $A_t \subseteq S_t$ of emails that are difficult to classify, which we refer to as uncertain emails.
 - (d) The sets T_t and A_t are labeled as ham, phishing, or spam. Phishing and spam emails are grouped together to form the non-ham class. Statistics about the labels of these emails are stored.
 - (e) The current model M_{t-1} is evaluated on the set T_t and evaluation results are stored.

- (f) The set A_t is added to the training set. For the current deployment no deletion of emails from the training set is performed. The new larger training set is used to train the new model M_t based on the same set of features.

4.1 Deployment Details

The deployment took place over a period of $n = 20$ days. The initial training set consisted of 4489 emails captured and labeled during the six days of October 3, 7, 8, 9, 13, and 14. It consists of 1514 ham emails, 1342 phishing emails, and 1633 spam emails.

During the initial study the following features were identified as being the most useful: unigram scores, DMC scores, semantic topic scores with $k = 25$ topics, link, and lexical features.

Every day a total of $|T_t \cup A_t| = 750$ emails are added:

- The underlying base set S_t of emails that the AntiPhish filter system is applied to consists of 15000 to 30000 emails per day.
- The test set T_t , $|T_t| = 250$ is selected from all emails in S_t . T_t is a stratified sample of its underlying base set S_t . The underlying idea is to “better” represent “interesting” emails. This idea is based on the assumption is that emails that are difficult to classify come in a larger variety whereas emails that are easy to classify are more or less alike. Now the basic procedure is to oversample the difficult emails to ensure their variety is properly represented, but to give them a lower weight in the evaluation to ensure that the overall proportions are kept. Specifically, for $T_t \subseteq S_t$ the procedure is as follows: Using M_0 the base set S_t is divided into two sets of uncertain and certain emails $S_t = S_t^{(u)} \cup S_t^{(c)}$; we use a probability of $p_\theta = 0.95$ (for non-ham, $p_\theta = 0.05$ for ham) as certainty threshold. Let $s^{(u)}$ and $s^{(c)}$ be the respective sizes of the two sets, i.e., $s^{(u)} + s^{(c)} = |S_t|$. Now we want to sample k_1 emails from $S^{(u)}$ and k_2 emails from $S^{(c)}$, in our case $k_1 = k_2 = 125$. For the final evaluation, i.e., the confusion matrix, each of the k_1 emails from $S^{(u)}$ is weighted with

$$w_1 = \frac{s^{(u)}}{|S_t|} \cdot \frac{k_1 + k_2}{k_1}$$

and, analogously, each of the k_2 emails from $S^{(c)}$ with

$$w_2 = \frac{s^{(c)}}{|S_t|} \cdot \frac{k_1 + k_2}{k_2}$$

- The “active” set A_t , $|A_t| = 500$ consists of the 400 top-ranked emails from S_t having the lowest confidence in classification plus 100 emails randomly selected from the rest of S_t . Minimization of duplicates among the 400 uncertain emails is ensured by considering only distinct confidences.

As false positives, i.e., missed ham emails, are less tolerable than false negatives, we use a threshold for classification. The model returns a probability p with which an email is considered as non-ham. Instead of using $p = 0.5$ as classification threshold we use our “certainty threshold” of $p_\theta = 0.95$. That is to say that an email is classified as

unwanted if and only if it is considered with a probability of at least 95% to be non-ham, otherwise it is classified as ham.

4.2 Scenario and Data

The evaluation scenario is designed to evaluate the AntiPhish filter system as a stand-alone classifier. A random sample of all emails is used for initial training and for identifying “difficult” emails to be added during the active learning. The test set is constructed in the same manner. I.e., a random subset of the complete email stream is selected (using stratified sampling) for testing the AntiPhish filter system. Email data was gathered continuously at Tiscali from the real-life email stream. It was furthermore annotated in real-time with the output of various spam filters, both commercial and non-commercial.

4.2.1 Initial Training Data

The initial training data was collected on six days between October 3rd and October 14th, 2008. During every one of those six days 750 emails were randomly selected and annotated. A few emails that could not be labeled with certainty, such as emails in unknown languages, were eliminated. In total the initial training data consists of 4489 emails, 1514 ham emails (33.73%), 1342 phishing emails (29.90%), and 1633 spam emails (36.38%). Figure 2 shows the distribution of the emails over the six days.

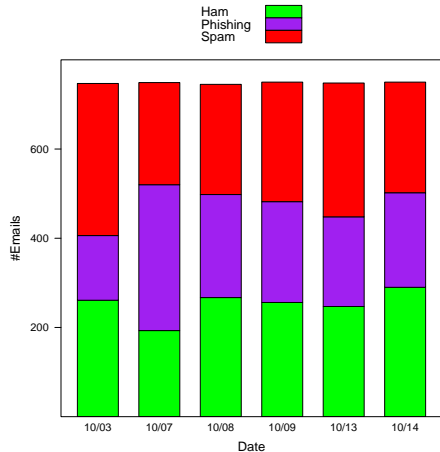


Figure 2: Composition of Initial Training Data

4.2.2 Active Learning Data

Every day 500 new emails are added to the training data, so that the final training data for day 20 consists of 13989 emails (4489 + 19 · 500). This final training data consists of 4390 ham emails (31.38%), 2587 phishing emails (18.49%), and 7012 spam emails (50.13%). The 500 emails that are added every day consist of the 400 emails that the then current model is most uncertain about and 100 emails selected randomly from the rest of the daily email stream. The daily distribution of the added training emails is shown in Figure 3. The figure shows that the composition of the added training data varies strongly from day to day.

4.2.3 The Test Set

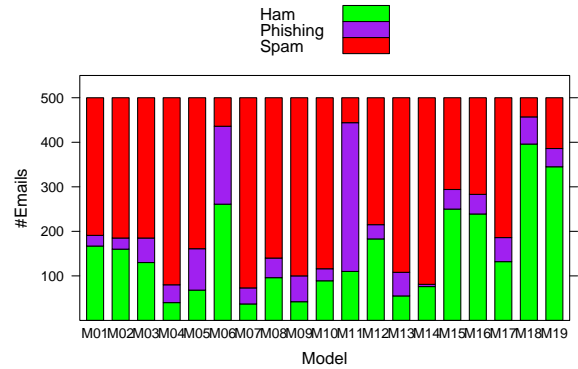


Figure 3: Composition of Added Training Data

Every day another 250 emails are labeled for the completely separate test set. These emails are selected using the previously described method of stratified sampling. In summary, the stratified sampling selects 125 emails that are difficult to classify according to the fixed initial model M_0 and 125 emails from the rest. These emails are then weighted to reflect the original proportions of “difficult” and “easy” emails in the daily emails stream. In reality this means that the “difficult” emails get a lower weight and the “easy” emails get a higher weight, because the real-life email stream contains many more “easy” emails than “difficult” email meaning that the “difficult” emails are grossly overrepresented in the test set. For very few days we were unable to select even 125 “difficult” emails, in these cases the test set then consists of fewer emails. Figure 4 shows the composition of the daily test set.

4.3 Evaluation Targets

We report standard measures for binary classification, such as precision, recall, and F-measure (or, more precisely, the F1-measure) as well as the false positive and the false negative rate. These measures are defined as:

$$\begin{aligned}
 precision &= \frac{|TP|}{|TP| + |FP|} & recall &= \frac{|TP|}{|TP| + |FN|} \\
 f &= \frac{2 \cdot precision \cdot recall}{precision + recall} \\
 fpr &= \frac{|FP|}{|FP| + |TN|} & fnr &= \frac{|FN|}{|TP| + |FN|}
 \end{aligned}$$

In email classification, errors are not of equal importance. A false positive is much more costly than a false negative. It is thus desirable to have a classifier with a low false positive rate. By choosing a classification threshold the one error can be traded for the other.

5. EXPERIMENTAL RESULTS

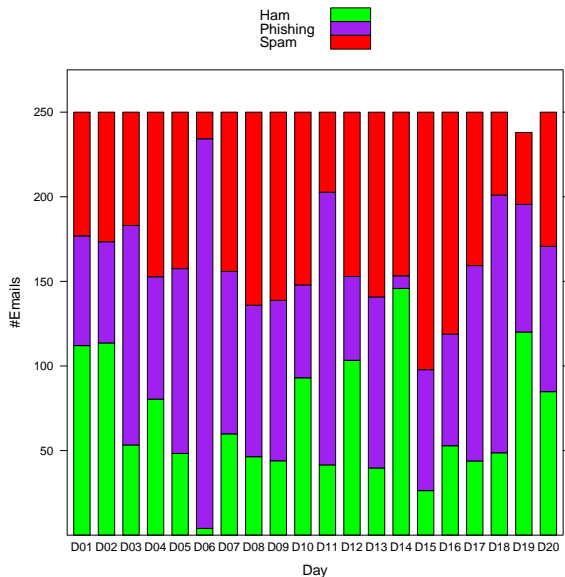


Figure 4: Composition of Daily Test Set

5.1 General Result

Here we present the false positive rate and false negative rate for each of the 20 days. Remember that the default decision threshold for deciding whether an email is unwanted is $\theta = 0.95$, i.e., only when the classifier is 95% sure that an email is unwanted it is classified as unwanted, i.e., as true. We achieve an average false positive rate of 0.34% and an average false negative rate of 7.09% (macro-averaged over the 20 days). Figure 5 and Table 1 give the detailed results.

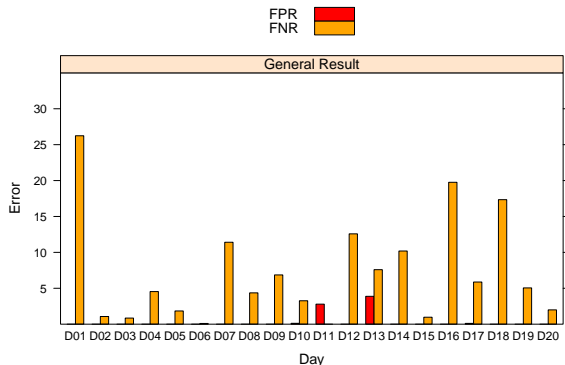


Figure 5: General Result

Only on four days the false positive rate is above zero. However, it is above 2.5% on two of those, day 11 and day 13, which we would like to refer to as the two “difficult days”.

Day	FPR	FNR	Day	FPR	FNR
1	0.00%	26.24%	11	2.79%	0.00%
2	0.00%	1.07%	12	0.00%	12.58%
3	0.00%	0.85%	13	3.88%	7.59%
4	0.00%	4.54%	14	0.00%	10.19%
5	0.00%	1.84%	15	0.00%	0.97%
6	0.00%	0.10%	16	0.00%	19.76%
7	0.00%	11.41%	17	0.11%	5.87%
8	0.00%	4.36%	18	0.00%	17.34%
9	0.00%	6.86%	19	0.00%	5.05%
10	0.12%	3.26%	20	0.00%	1.99%
Mean				0.34%	7.09%

Table 1: General Result

Unfortunately, the nature of this study does not permit us to investigate content details about those misclassified emails. During some days we observe relatively high false negative rates, which is low on other days. This variation probably is caused by a batch of similar emails of a new type. On the next day the filter adapts to these emails leading again to low false negative rates.

5.2 ROC Analysis

In our experiment we set the default decision threshold for deciding whether an email is unwanted as $\theta = 0.95$. However, by moving this threshold we can trade false positive errors for false negative ones and vice versa. A larger value for θ means that the classifier will be even more conservative, i.e., it will produce fewer false positives but more false negatives. A smaller value for θ leads to a more liberal classifier.

In this section we investigate, which choice of threshold would have been optimal. As a first step we depict in Figure 6 the optimal threshold for each individual day. We define “optimal” as the lowest possible threshold that leads to zero false positives. As the figure demonstrates this value varies quite substantially between 0.01 and 0.99. On most of the days it is below our selected default value. This large variance seems quite natural, because of the large variance in the test sets for each individual day.

For the next figure we use this optimal value for classification, the result is set in comparison to the general result in Figure 7. The false positive rate now is zero on every day, but we can observe a quite substantial increase in false negative rate.

Of course, it is impossible to know the optimal value for θ in advance for every day. Furthermore, we are interested in a conservative filter, because a false positive represents a lost email. In Figure 8 we show the results for different preselected thresholds. From the top to the bottom the false positive rate decreases whereas the false negative rate increases. In conclusion we can say that our original choice of $\theta = 0.95$ turned out to be a good one.

5.3 The Effect of Active Learning

We investigate the effect of active learning, i.e., the effect of regularly updating the model with new annotated emails, which were perceived as difficult to classify by the previous model. In this experiment we compare our previously obtained active learning results to results with a fixed model. Note that we do not compare the performances of an active

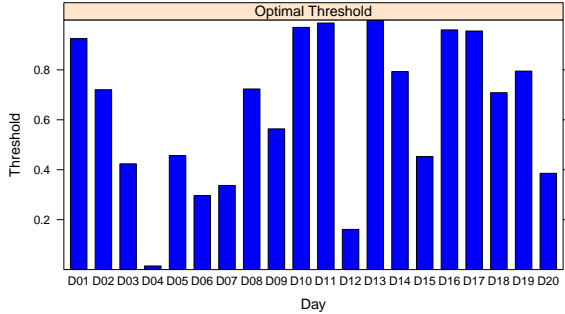


Figure 6: The optimal cutting threshold for every day of the evaluation

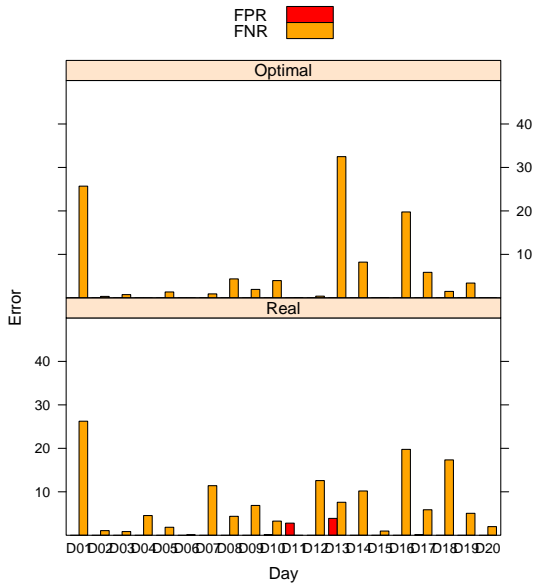


Figure 7: Result for the optimal cutting threshold

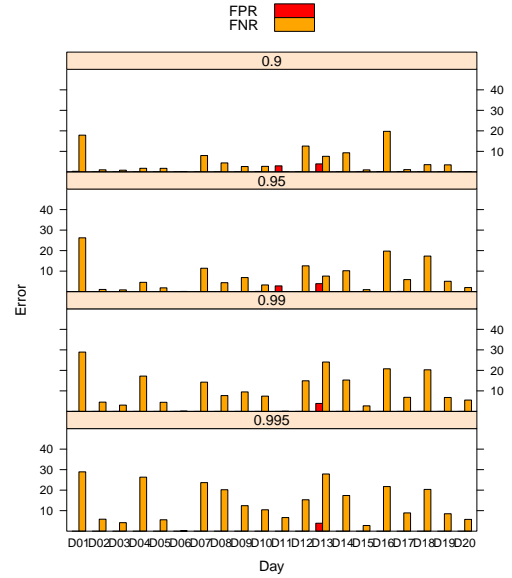


Figure 8: Result for selected cutting thresholds

learning model and a model completely based on random sampling. This difference is well established in the literature (e.g., in [12]). Here we want to know, if active learning is necessary to adapt to new features of emails. We use three different fixed models: our initial model M_0 obtained from the initial training data, the model M_5 obtained after five days of active learning, and the model M_{10} obtained after ten days of active learning. Of course, the latter two models are investigated respectively only from days six and eleven onwards. Likewise, on day six model M_5 of course gives the same result as the continuously updated active model; and on day eleven model M_{10} gives the same result as the active model. The main message from this experiment is that the active learning clearly turns out to be useful. In terms of false negative rate the active model performs best, followed in order by M_{10} , M_5 , and M_0 performs worst. For the false positive rate the results are not as clear-cut, however, we believe that the low performance of M_{10} here is mainly due to the fact that in its shorter evaluation period of ten days lie the two most difficult days, 11 and 13. Unfortunately, again, we cannot go into details about emails that were misclassified using a fixed model and then correctly classified using the active model. Figure 9 and Table 2 give the detailed results.

5.4 Combining Spam Filters and the AntiPhish Filter System

One goal of this evaluation is to combine existing spam filters with the AntiPhish filter system. We use a commercial spam filter and integrate it into the AntiPhish filter system in two different manners. The first approach is the Spam Filter-AntiPhish pipe. Only emails that pass the spam filter are forwarded to the AntiPhish filter system for further filtering. The second approach we follow is to include the spam filter output as a binary feature in the AntiPhish filter system. This approach has the advantage that only one integrated model is trained. Our hypothesis is that one integrated optimized model performs better than two separately

Day	Model M_0		Model M_5		Model M_{10}		Active Model	
	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR
1	0.00%	26.24%					0.00%	26.24%
2	1.66%	8.66%					0.00%	1.07%
3	0.00%	4.79%					0.00%	0.85%
4	0.00%	13.29%					0.00%	4.54%
5	0.00%	14.01%					0.00%	1.84%
6	0.00%	7.32%	0.00%	0.10%			0.00%	0.10%
7	2.90%	33.30%	0.00%	13.90%			0.00%	11.41%
8	0.00%	17.30%	0.00%	5.74%			0.00%	4.36%
9	0.00%	39.50%	0.00%	2.91%			0.00%	6.86%
10	0.12%	3.26%	0.12%	12.74%			0.12%	3.26%
11	0.00%	2.34%	2.64%	2.27%	2.79%	0.00%	2.79%	0.00%
12	0.00%	21.81%	0.00%	19.83%	0.00%	13.36%	0.00%	12.58%
13	3.88%	26.75%	3.88%	22.09%	3.88%	10.74%	3.88%	7.59%
14	0.00%	19.95%	0.00%	21.04%	0.00%	16.84%	0.00%	10.19%
15	0.00%	25.43%	0.00%	24.46%	0.00%	24.16%	0.00%	0.97%
16	0.00%	29.21%	0.12%	29.82%	0.23%	28.78%	0.00%	19.76%
17	0.00%	5.34%	0.00%	4.37%	0.00%	5.27%	0.11%	5.87%
18	0.00%	2.61%	0.08%	19.95%	0.08%	1.29%	0.00%	17.34%
19	0.00%	4.06%	0.02%	30.77%	0.02%	3.43%	0.00%	5.05%
20	0.00%	4.23%	0.00%	6.31%	0.00%	3.17%	0.00%	1.99%
Mean	0.43%	15.47%	0.46%	14.42%	0.70%	10.70%	0.34%	7.09%

Table 2: The Effect of Active Learning

optimized local models arranged as a pipe.

5.4.1 The Spam Filter-AntiPhish Pipe

Our first approach is the Spam Filter-AntiPhish pipe. In this approach any email in the test set is filtered in two phases. First, it is sent through the spam filter. If the spam filter classifies the email as spam, it is filtered out. Otherwise it is forwarded to the AntiPhish filter system. Its output is the final classification for that email. One important consequence arises from this approach: The AntiPhish filter system cannot recover any false positive of the spam filter. Consequently, the false positive rate of the pipe is at least as high as the false positive rate of the spam filter itself. The goal of the AntiPhish filter system is to filter out more unwanted emails, i.e., to lower the false negative rate, without introducing more false positives. Figure 10 and Table 3 give the details. To sum up, the AntiPhish filter system substantially reduces the false-negative rate of the spam filter, on average from 6.10% to 0.81%. At the same time, only on three days (10, 11, and 13) additional false positives occur, altogether the false positive rate only increases from 5.38% to 5.72%.

5.4.2 Spam Filter Output as Feature in the AntiPhish Filter System

The second alternative is to include the spam filter output as a feature in the AntiPhish filter system. That is, in addition to the features we already use in the AntiPhish filter system (unigram, DMC, semantic topics with $k = 25$ topics, link, and lexical features) we include one Boolean feature indicating whether or not the spam filter classifies an email as spam.

We show the result of this integrated AntiPhish filter system and compare it to the original AntiPhish filter system. Figure 11 and Table 4 give the details. Through the inclusion of the spam filter feature the number of days with

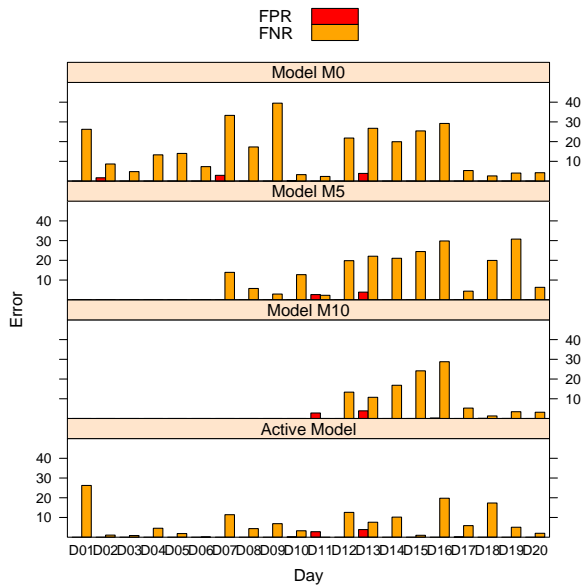


Figure 9: The Effect of Active Learning

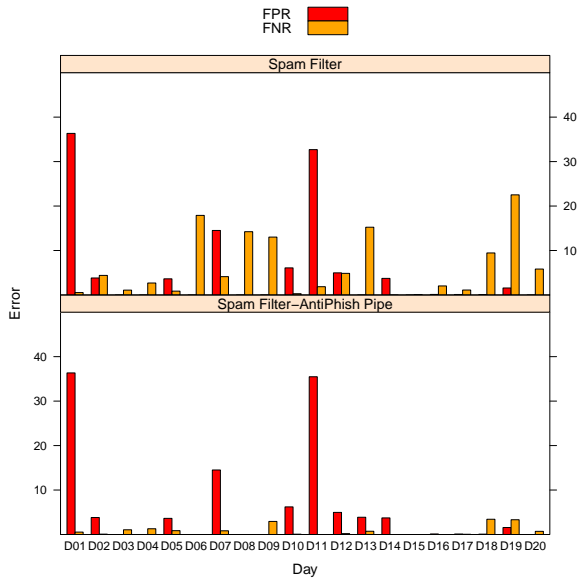


Figure 10: Result for the Spam Filter-AntiPhish pipe

Day	Spam Filter		Pipe	
	FPR	FNR	FPR	FNR
1	36.34%	0.54%	36.34%	0.54%
2	3.82%	4.40%	3.82%	0.08%
3	0.00%	1.09%	0.00%	1.06%
4	0.00%	2.69%	0.00%	1.29%
5	3.63%	0.86%	3.63%	0.86%
6	0.00%	17.90%	0.00%	0.00%
7	14.51%	4.12%	14.51%	0.83%
8	0.00%	14.23%	0.00%	0.00%
9	0.00%	13.02%	0.00%	2.94%
10	6.09%	0.28%	6.21%	0.07%
11	32.69%	1.86%	35.48%	0.00%
12	4.97%	4.86%	4.97%	0.20%
13	0.00%	15.24%	3.88%	0.73%
14	3.73%	0.00%	3.73%	0.00%
15	0.00%	0.07%	0.00%	0.00%
16	0.12%	2.03%	0.12%	0.00%
17	0.11%	1.11%	0.11%	0.05%
18	0.08%	9.45%	0.08%	3.43%
19	1.57%	22.52%	1.57%	3.32%
20	0.00%	5.83%	0.00%	0.71%
Mean	5.38%	6.10%	5.72%	0.81%

Table 3: Result for the Spam Filter-AntiPhish pipe

non-zero false positive rate is reduced from four to two. On only one of the 20 days the false negative rate is increased, on 17 days it is decreased, sometimes substantially so. Comparing this result to the result of the previous section, this integrated classifier clearly outperforms the pipe, the main reason is that the integrated classifier can correct false positives of the spam filter. This integrated classifier is the best-performing classifier of the complete series of experiments.

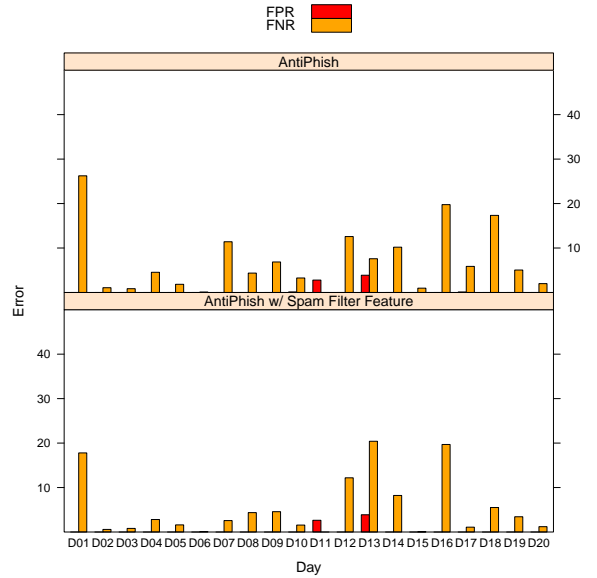


Figure 11: Result for AntiPhish Filter System with Spam Filter Output as Feature

5.5 Phishing Classification

The original goal of the AntiPhish project is of course to detect phishing emails. In this experiment we consider the subset of our data that consists only of ham and phishing emails, and we ignore the spam data. This holds for all data including the test data and the additional training data obtained through active learning. Specifically, our initial training set now consists of 1514 ham emails and 1342 phishing emails. We continue to use the default decision threshold for deciding whether an email is phishing $\theta = 0.95$. Figure 12 and Table 5 give detailed results. In the table we list only the false negative rate, as the false positive rate is zero on every single day. The false negative rate varies strongly, though the average is with 7.62% about the same as in the general experiment. This variation is probably caused by a set of similar emails of a new type, which appear at some day. The filter is able to adapt to these emails in the next round.

6. CONCLUSIONS

Phishing is a multi-billion dollar activity and represents a serious threat to global security and economy. In the AntiPhish project we aim for phishing prevention through content-based email filtering. We have provided advanced text mining features for phishing detection, such as dynamic Markov chain scores or latent topic model scores. These

Day	AntiPhish		W/ Spam Filter	
	FPR	FNR	FPR	FNR
1	0.00%	26.24%	0.00%	17.79%
2	0.00%	1.07%	0.00%	0.58%
3	0.00%	0.85%	0.00%	0.81%
4	0.00%	4.54%	0.00%	2.81%
5	0.00%	1.84%	0.00%	1.60%
6	0.00%	0.10%	0.00%	0.10%
7	0.00%	11.41%	0.00%	2.57%
8	0.00%	4.36%	0.00%	4.36%
9	0.00%	6.86%	0.00%	4.57%
10	0.12%	3.26%	0.00%	1.56%
11	2.79%	0.00%	2.64%	0.00%
12	0.00%	12.58%	0.00%	12.19%
13	3.88%	7.59%	3.88%	20.41%
14	0.00%	10.19%	0.00%	8.22%
15	0.00%	0.97%	0.00%	0.07%
16	0.00%	19.76%	0.00%	19.69%
17	0.11%	5.87%	0.00%	1.09%
18	0.00%	17.34%	0.00%	5.51%
19	0.00%	5.05%	0.00%	3.43%
20	0.00%	1.99%	0.00%	1.20%
Mean	0.34%	7.09%	0.33%	5.43%

Table 4: Result for AntiPhish Filter System with Spam Filter Output as Feature

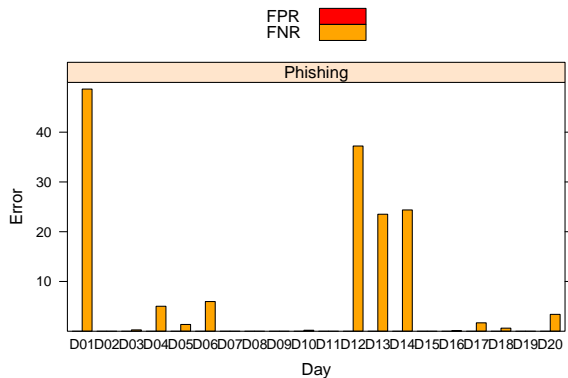


Figure 12: Result for phishing vs. ham classification

Day	FNR	Day	FNR
1	48.66%	11	0.00%
2	0.00%	12	37.22%
3	0.27%	13	23.51%
4	5.02%	14	24.37%
5	1.36%	15	0.00%
6	5.97%	16	0.09%
7	0.00%	17	1.69%
8	0.00%	18	0.62%
9	0.00%	19	0.00%
10	0.20%	20	3.39%
Mean			7.62%

Table 5: Result for phishing vs. ham classification

features are not handcrafted but are themselves statistical models.

In this work we have deployed our system in real-life and achieved encouraging results. Our system can be tuned to perform well in a real-life setting where spam emails are part of the data. In fact, the system performs very well as a spam filter even though it was designed with the phishing target in mind. A notable result is that we showed that active learning is useful in real-life and enables the classifier to remain up-to-date. Furthermore, other spam filters can be integrated into the system to improve overall performance. Last but not least the system was applied to approximately half a million emails during the course of the deployment, which demonstrates that it is applicable in a real-life setting.

The very nature of a real-life study means that it has some limitations. First, it is not reproducible for the reader. Second, we cannot investigate details about misclassifications of emails and adapt the system accordingly.

7. ACKNOWLEDGMENTS

This paper is based upon work performed within the FP6-027600 project AntiPhish (<http://www.antiphishresearch.org/>). The authors would like to thank the European Commission for partially funding the AntiPhish project as well as all the AntiPhish project partners (Symantec, Nortel, and Tiscali Italia) for their interest, support, and collaboration in this initiative.

8. REFERENCES

- [1] A. Bergholz, J. De Beer, S. Glahn, M. F. Moens, G. Paass, and S. Strobel. New filtering approaches for phishing email. *Journal of Computer Security*, 18:7–35, 2010.
- [2] A. Bergholz, G. Paass, F. Reichartz, S. Strobel, and J. H. Chang. Improved phishing detection using model-based features. In *Proceedings of the Conference on Email and Anti-Spam (CEAS)*, Mountain View, CA, USA, August 2008.
- [3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
- [4] G. V. Cormack and R. N. Horspool. Data compression using dynamic markov modelling. *The Computer Journal*, 30(6):541–550, 1987.
- [5] I. Fette, N. Sadeh, and A. Tomasic. Learning to detect phishing emails. In *Proceedings of the International*

- World Wide Web Conference (WWW)*, pages 649–656, Banff, Canada, May 2007.
- [6] J. Goodman, G. Cormack, and D. Heckerman. Spam and the ongoing battle for the inbox. *Communications of the ACM*, 50(2):24–33, 2007.
 - [7] S. C. H. Hoi, R. Jin, and M. R. Lyu. Large-scale text categorization by batch mode active learning. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 633–642, Edinburgh, Scotland, UK, May 2006.
 - [8] M. Li and I. K. Sethi. Confidence-based active learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1251–1261, 2006.
 - [9] Y. Marton, N. Wu, and L. Hellerstein. On compression-based text classification. In *Proceedings of the European Colloquium on IR Research (ECIR)*, pages 300–314, Santiago de Compostela, Spain, March 2005.
 - [10] D. Sculley. Online active learning methods for fast label-efficient spam filtering. In *Proceedings of the Conference on Email and Anti-Spam (CEAS)*, Mountain View, CA, USA, August 2007.
 - [11] D. Sculley and G. M. Wachman. Relaxed online svms for spam filtering. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 415–422, Amsterdam, The Netherlands, July 2007.
 - [12] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2002.
 - [13] Y. Zhang, J. Hong, and L. Cranor. Cantina: A content-based approach to detecting phishing web sites. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 639–648, Banff, Canada, May 2007.