



# Deliverable 6.3

## **Third Phase Prototype Report**

Fraunhofer IAIS  
Version 06  
11. August 2009



IST 2006 027600

AntiPhish

Anticipatory Learning for Reliable Phishing Prevention

Specific Targeted Research or Innovation Project

2.4.3 Towards a global dependability and security framework

## D 6.3 Third Phase Prototype Report

Due date of deliverable: M42 (30 June 2009)

Actual submission date: 11 August 2009

Start date of project: 01. Jan. 2006

Duration: 42 months

Lead Contractor for this Deliverable: Fraunhofer IAIS

Revision: 06

<b>Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)</b>		
<b>Dissemination Level</b>		
<b>PU</b>	Public	X
<b>PP</b>	Restricted to other programme participants (including the Commission Services)	
<b>RE</b>	Restricted to a group specified by the consortium (including the Commission Services)	
<b>CO</b>	Confidential, only for members of the consortium (including the Commission Services)	

## Revision history

Deliverable administration and summary		
Project acronym: AntiPhish		ID: IST-2006-027600
Document identifier:	AntiPhish-del-D63-ThirdPhasePrototypeReport-f-v06	
Leading partner: Fraunhofer IAIS		
Report version: v06		
Report preparation date: 11 August 2009		
Classification: Public		
Nature: Report		
Author(s) and contributors: André Bergholz, Gerhard Paass in collaboration with all partners		
Status:		Plan
		Draft
	X	Working
		Final
		Submitted
		Approved

The AntiPhish © Consortium has addressed all comments received, making changes as necessary. Changes to this document are detailed in the change log table below.

Date	Edited by	Status	Changes made
-	DoW	Plan	report template
15.07.09	André Bergholz	Working	Description
03.08.09	Gerhard Paaß	Working	Additions and editing
07.08.09	Patrick Horkan	Working	Edits, comments
11.08.09	Gerhard Paaß	Final	Finalization

Notice that other documents may supersede this document. A list of latest public AntiPhish deliverables can be found at the AntiPhish webpage at [www.AntiPhishResearch.org/publications](http://www.AntiPhishResearch.org/publications).

## Copyright

This report is © AntiPhish Consortium 2009. Its duplication is allowed only in the integral form for anyone's personal use for the purposes of research or education.

## Citation

André Bergholz, Gerhard Paaß (2009). Deliverable D6.3 Third Phase Prototype Report. Fraunhofer IAIS, AntiPhish Consortium , [www.antiphishresearch.org](http://www.antiphishresearch.org)

## Acknowledgements

The work presented in this document has been conducted in the context of the EU Framework Programme project IST 2006 027600 AntiPhish. AntiPhish is a 36-month project that started on January 1st, 2006 and is funded by the European Commission as well as by the industrial partners. Their support is appreciated.

The partners in the project are Fraunhofer Institute for Intelligent Analysis and Information Systems (FHG), Symantec LIRIC Limited (LIRIC), Symantec Ltd. (Symantec Ireland), TISCALI Services S.r.l. (Tiscali) and K. U. Leuven / ICRI-LIIR (K.U. Leuven). The content of this document is the result of extensive discussions within the AntiPhish© Consortium as a whole.

## More information

Public AntiPhish reports and other information pertaining to the project are available through AntiPhish public web site under [www.antiphishresearch.org](http://www.antiphishresearch.org).

## Table of contents

1	Introduction .....	7
2	Machine Learning Framework .....	9
2.1	Active Learning .....	9
2.2	Stratified Evaluation .....	11
2.3	Incorporation of New Features .....	12
2.4	Active Learning of Dynamic Markov Chains .....	13
2.5	Outlier Detection by Dynamic Markov Chains .....	15
3	Evaluations of the AntiPhish System .....	18
3.1	Evaluations on Benchmark Data .....	18
3.2	Evaluations in Field Experiments in an Industrial Setting .....	19
4	Conclusions .....	20
5	References .....	21

## Executive summary

This document describes and documents the third prototype of the AntiPhish Filter System (APS) which is the outcome of WP6. The report concentrates on the progress that has been made at the end of the second phase (month 25-29) until milestone M6.2 in May 2008 and in the third phase of the project until milestone M6.3 in June 2009. The third phase was devoted to transferring the APS to “the real-world” experiments.

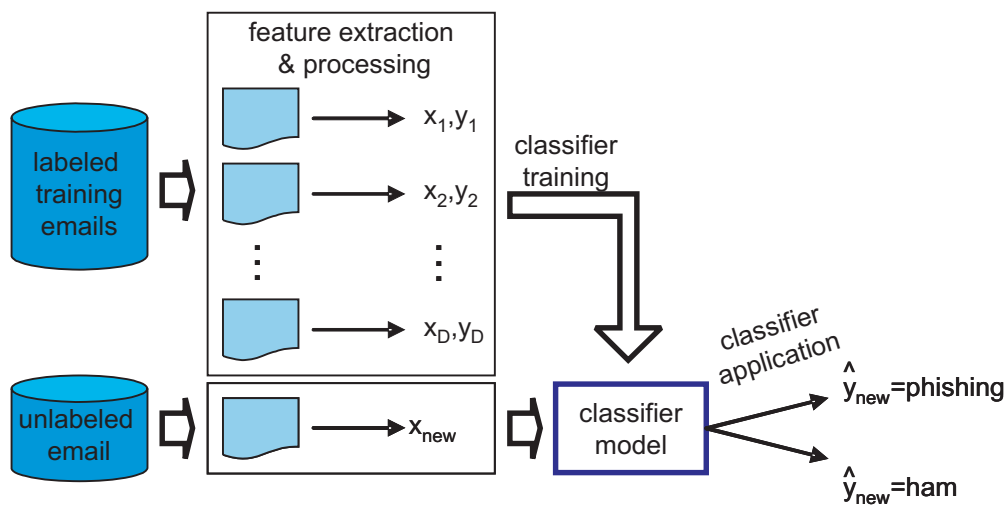
# 1 Introduction

Machine learning techniques, in particular automatic classification, have become popular in email and phishing detection. In contrast to manually constructed filter rules they automatically assess the relevance of input features  $x = (x_1, \dots, x_m)$  (e.g., email characteristics) and establish a function to determine the desired classification  $y$  (e.g., phishing or non-phishing)

$$y = f(x, \gamma)$$

The vector of unknown parameter values  $\gamma$  is determined in a training phase in such a way that the relation between  $x$  and  $y$  in the observed data  $(x_1, y_1), \dots, (x_D, y_D)$  is reproduced according to some optimization criterion. In the application phase the same features are extracted from a new incoming email. Based on these features and the model the classifier produces a classification of the email. The overall machine learning approach is summarized in Figure 1.

**Figure 1: Machine Learning Approach for Email Filtering**



Arising from this research is the AntiPhish prototype which enables the various research findings to be tested on actual emails. It contains a large number of modules generating features for classification, e.g. email structure features or topic models. These features then are fed to a classifier, which is trained to get reliable classifications. Through-out this document we refer to the prototype as the APS – short for AntiPhish Filter System.

The third phase of the AntiPhish project sought to provide a solution capable of real-world application. To this end the APS was deployed on a real-life normal email stream and a real-life general spam stream. Specifically APS was deployed to:

1. Filter a subset of a “normal email stream”, i.e. one containing legitimate emails (ham), spam and phishing emails. The task here was to filter out phishing and spam emails as well as possible.

2. Filter a subset of a “general spam stream” to separate phishing from regular-spam. This is an important task within the context of an anti-Spam operation centre that gives priority in responding to spam attacks of a phishing/fraudulent nature.

To support these field experiments the APS was extended and improved. Both experiments required the implementation of new workflows with complex evaluations. The necessary methodological enhancements are described in this document as well as in the feature extraction report D5.4. The details of the experiments and the results are given in deliverable D3.3.

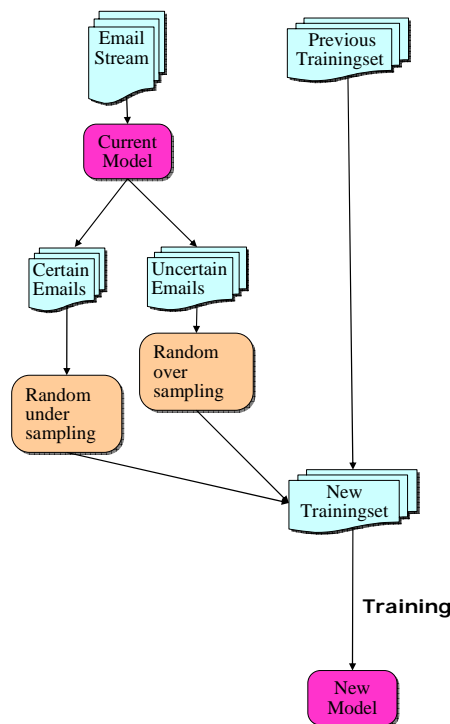
## 2 Machine Learning Framework

The AntiPhish machine learning framework (APS – AntiPhish Filter System) as described in deliverables 6.1 and 6.2 has been extended and improved to be used in a real-life environment during the third year.

- We have extended our work on active learning and provide workflows that implement this technique.
- For evaluating classifier performance we have extended the existing work to cover stratified evaluation.
- Furthermore the work on feature extraction has been consolidated and extended by new features based on PCA.
- We have reduced the size of dynamic Markov chain models (DMC) by a specific active sampling technique.
- Last but not least we have developed a new technique for outlier detection based on our previous work on dynamic Markov chains.

### 2.1 Active Learning

Active learning is a technique to deal with the problem of too few labelled data. This is an important scenario in the case of email classification as it is impossible to manually label hundreds of thousands of emails. Active learning identifies emails where the confidence of classification is low. These emails are manually labelled and added to the training set.



For the field testing we did a thorough investigation of current active learning technology for text classification [Boulal (2009)]. Main point of the active learning algorithm is the selection of new data points for labelling. The efficiency of the algorithm depends on the information content of the new instances for the classification problem.

For classification we utilize the Support Vector Machine. It determines an optimal hyperplane between the instance sets of the two cases. Penalty terms are used if the classes are not linearly separable. [Tong and Koller (2000)] have shown that an unlabeled instance with a minimum distance to the current hyperplane leads to the largest reduction of prediction error. However this distance criterion is only sensible in the case of a single new data point to be labelled. If several instances have to be selected during an active learning iteration we have to induce the diversity of the selected new data points.

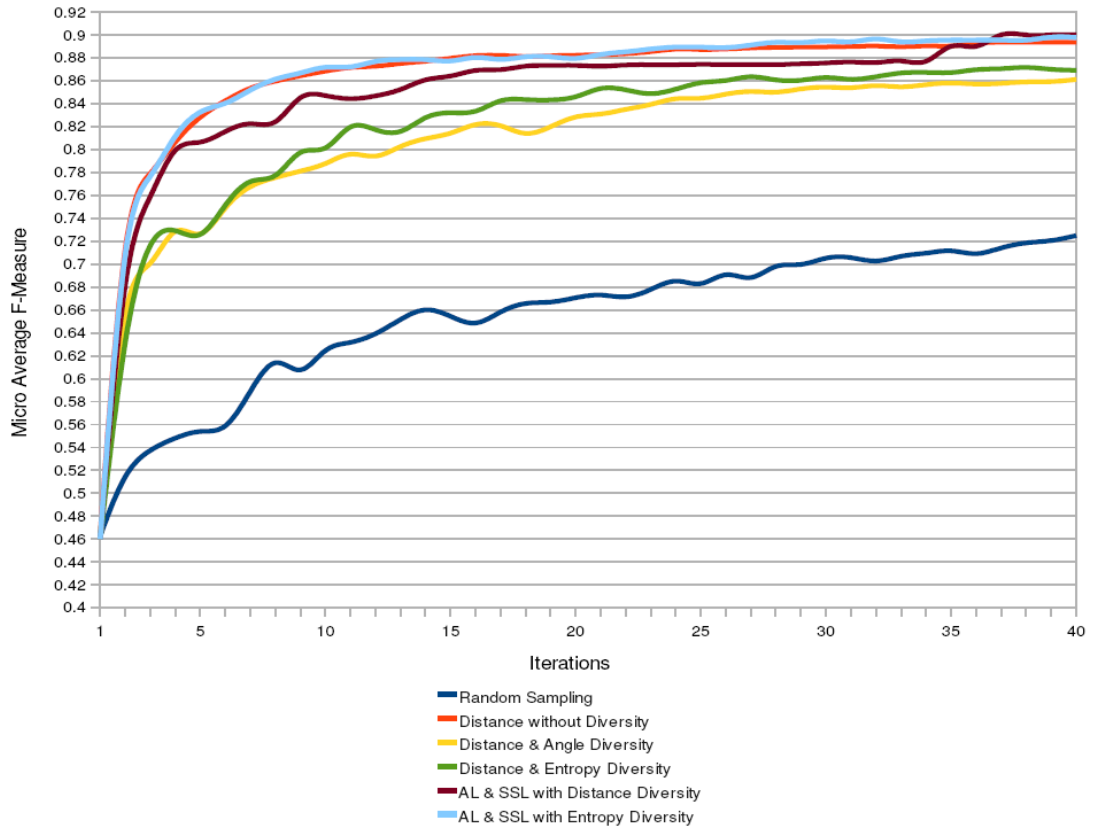
We have evaluated the following selection strategies for active learning:

1. **Random sampling.** Here we simple select new data points randomly for labelling. This strategy is evaluated for comparison purposes.
2. **Distance without Diversity.** Here the unlabeled documents are selected according to their distance to the SVM hyperplane.
3. **Distance & Angle Diversity.** Here the weighted sum of the distance to the hyperplane and the inverse angle between the feature functions  $\Phi(x_i)$  and  $\Phi(x_j)$  of two new data points  $x_i$  and  $x_j$  is minimized. The weight  $\alpha$  is determined empirically. This ensures the diversity of new data points.
4. **Distance & Entropy Diversity.** For this criterion a weighted sum of the distance to the hyperplane and the inverse empirical entropy is minimized.
5. **AL & SSL with Distance Diversity.** In this case the new elements are selected with two different criteria. Part of the data points are selected according to the Distance & Angle Diversity criterion. These uncertain documents are manually labelled. The rest of the documents are selected by semi-supervised learning. Here the documents with the most confident classifications are selected and automatically labelled with that classification. In this way manual labelling effort is saved.
6. **AL & SSL with Entropy Diversity.** As above the new elements are selected with two different criteria. Part of the data points are selected according to the Distance & Entropy Diversity criterion and manually labelled. As before the rest of the new elements selected according to semi-supervised learning, i.e. according to their classification confidences and labelled automatically.

The following Figure 2 shows the micro f-values for a text mining classification task with multiple classes. It is evident that all approaches show a marked improvement over the Random Sampling baseline. It turns out, that the simple distance-based approach yields very good results. The diversity approaches using angle and entropy diversity were somewhat inferior. Very competitive results were also achieved with the AL & SSL with Entropy Diversity approach.

Based on these results we decided to use mainly the distance criterion for our active learning field experiments. Active learning is prone to getting stuck in a local optimum as adding emails selected by one sole criterion can move the classification model in one specific direction with every iteration. To deal with this problem we extended our

previous mechanism by adding a certain proportion (20%) of randomly selected emails to ensure a level of diversity of the training set.



**Figure 2 Micro F-Values for Active Learning in a Multiclass Classification Task using different Active Learning Selection Criteria (see Text). The x-Axis denotes the Number of Active Learning Iterations.**

In the third phase prototype APS we extended our previous first workflows to be able to cope with a real-life situation of tens of thousands of emails per day.

## 2.2 Stratified Evaluation

The third phase prototype APS includes workflows for stratified evaluation of a classifier. So far we have always treated every email in the test set in the same way, i.e., with equal importance. In stratified evaluation one can separate the test set according to some fixed criterion, for example whether or not an email has a financial background. One can then over-sample one part of the test set and under-sample another one to better simulate the variance of the interesting part. Of course, to respect the original proportions of the test set weights have to be assigned to the emails. The over-sampled part gets a lower weight whereas the under-sampled part gets a higher weight.

Stratified evaluation has been used during the field experiments conducted on a normal email stream. It is described in more detail in the separate report.

In the context of the AntiPhish project we use stratified evaluation in the following manner. Our split criterion is whether or not an email is easy to classify. Our reasoning is that emails that are difficult to classify are more “interesting” than emails that are easy

to classify. In particular, we hypothesize that emails that are difficult to classify are more diverse. To get a better picture of the performance of our system we would like to capture this diversity by over-sampling these “interesting” emails.

Specifically, to select a stratified sample  $T_t \subseteq S_t$  the procedure is as follows:

- Using the some model  $M_0$  the base set  $S_t$  of emails 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  $|S_t^{(u)}|$  and  $|S_t^{(c)}|$  be the respective sizes of the two sets. Now we randomly sample  $k_1$  emails from  $S_t^{(u)}$  and  $k_2$  emails from  $S_t^{(c)}$ .
- The sampled emails are manually labelled with the class labels ham, spam, and phishing.
- For the final evaluation, i.e., the confusion matrix, each of the  $k_1$  emails from

$S_t^{(u)}$  is weighted with  $w_1 = \frac{|S_t^{(u)}|}{|S_t|} \frac{k_1 + k_2}{k_1}$  and, analogously, each of the

$k_2$  emails from  $S_t^{(c)}$  with  $w_2 = \frac{|S_t^{(c)}|}{|S_t|} \frac{k_1 + k_2}{k_2}$ .

It has been proven that stratification will achieve a greater precision of results, if subsets for a population with larger variance of the quantity of interest (here: correctness of classification) are over-sampled. Stratified evaluation has been used during the field experiments on a normal email stream (described in more detail in the separate report D3.3a).

## 2.3 Incorporation of New Features

The APS incorporates more and better features. A large effort was put into making existing features perform correctly on all encountered emails, because emails that cannot be parsed can bypass our filters. This became all the more urgent, because the APS was tested in a real-life environment with tens of thousands of emails per day.

The performance of the basic text and link features as well as the salting features was significantly improved through message preprocessing. During this step the emails are cleaned and inconsistencies removed.

More specifically, work has been done to improve the performance of the following features:

- the parsing of HTML-encoded emails to produce the correct message text and link information even for emails containing errors
- the extraction of links from text-based emails
- the detection of sentences and paragraphs

More importantly, the APS includes the newly developed features based on Principal Component Analysis (PCA). PCA is a method for dimensionality reduction. Much like for topic models the idea is to automatically reduce the many different words in emails to a smaller, relevant combination. PCA has been successfully applied in many domains. The PCA approach is described in more detail in deliverable D5.4.

For the third phase prototype Fraunhofer IAIS and KU Leuven integrated the PCA modelling system into the feature provider subsystem of the APS. Prior to a classification a PCA-model has to be trained. We trained a model with and without PCA on a dataset containing spam and phishing messages. The training data comprised 4452 emails collected in one month, the test data covered 33431 emails from the subsequent five months.

**Table 1: Classification of Phishing vs. Spam with the PCA and other features.**

Features	% Precision	% Recall	% F-Value
topic models, DMC, unigram, link und lexical	85.7	95.6	90.4
PCA	81.7	89.0	85.2
PCA & topic models, DMC, unigram, link und lexical	71.9	95.7	82.1

Because of the relative short training period and the long test period the performance level of all results is relatively low. It shows that PCA is itself a valuable feature. In conjunction with the topic models and DMC there seems to be some overfitting. This may be reduced by using larger training sets.

## 2.4 Active Learning of Dynamic Markov Chains

Dynamic Markov Chain features are based in 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. 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 [Cormack and Horspool (1987)]. These dynamic Markov chains have been successfully applied to text classification problems in various domains [Marton et al.(2005)] [Frank et al.(2000)]. 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.

[Bratko et al.(2006)] achieve good results for the classification of spam emails using the dynamic Markov chain method. They point out that one limitation of the method is the high memory requirement. In contrast to their work we convert emails into plain text meaning that all headers etc. are removed and file attachments are discarded. The reason

for the exclusion of header information is that we feel they might make synthetic test data trivially separable through the inclusion of domain names or IP addresses.

The Markov chain classification can be summarized as follows. The cross-entropy (CE)  $H(x, M)$  between the message  $x$  and the source approximated by the model  $M$  is a measure for the likelihood that a message  $x$  with the binary representation  $(b_1, \dots, b_n)$  originated from that source. The cross-entropy is defined as:

$$H(x, M) = -\frac{1}{n} \log \prod_{i=1}^n p(b_i | b_1^{i-1}, M)$$

where  $p(b_i | b_1^{i-1}, M)$  is the probability of seeing bit  $b_i$  based on the previous bits  $(b_1, \dots, b_{n-1})$  of the message. The class to which message  $x$  has the lowest cross entropy is the one it most likely originated from. Therefore the classification of  $x$  can be formulated based on the minimal cross entropy to all classes  $C$  as:

$$f(x) = \arg \min_{c \in C} H(x, M_c)$$

where  $M_c$  is the model for class  $c \in C$ .

Our intention is to reduce the size of the Markov models to overcome the limitations of the approach. To this end we reduce the number of training examples by using an efficient heuristic to judge the value of each specific example in terms of impact on the classification accuracy, similar to uncertainty sampling in active learning [Lewis and Gale (1994)].

Let  $M_c^{1,k}$  be the model that is generated after processing the training examples  $x_1, \dots, x_k$  of a class  $c$ . During the incremental model generation we think of  $H(x_i, M_c^{1,i-1})$  as the expected cross entropy of a training message  $x_i$  and the model  $M_c^{1,i-1}$ . Let the empirical standard deviation of the expected cross entropies  $\hat{\sigma}(k)$  of  $M_c^{1,k}$  be

$$\hat{\sigma}(k) = \sqrt{\frac{1}{k-2} \sum_{i=1}^{k-1} (H(x_i, M_c^{1,i-1}) - \bar{H}(x_i, M_c^{1,i-1}))^2}$$

where  $\bar{H}(x_i, M_c^{1,i-1}) = \frac{1}{i-1} \sum_{j=1}^{i-1} H(x_j, M_c^{1,j-1})$ . To choose an example for training based on

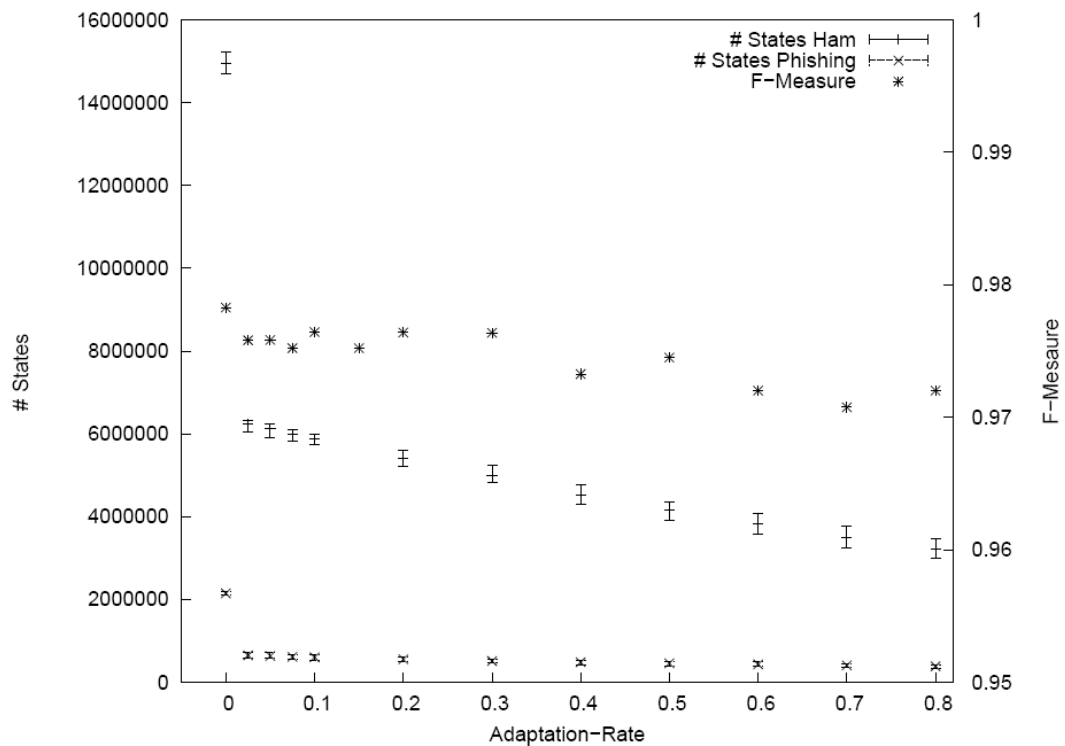
its usefulness for the classification accuracy we compare its cross entropy with the average and the standard deviation from the previous training steps. In other words, we only want to use training examples that the model cannot already classify well enough. We skip sequences that are most likely in the set of typical sequences for a source and use only training examples  $x_i$  for which the following equation holds:

$$H(x_i, M_c^{1,i-1}) - \bar{H}(x_i, M_c^{1,i-1}) > \rho * \hat{\sigma}(i)$$

with a given adaptation rate  $\rho$ . Our adaptive training uses a fixed percentage  $\tau$  of the training data for the generation of an initial model and then applies the heuristic bound to automatically adapt the training process. As will be empirically shown our heuristic adaptation technique limits the amount of space needed for each model by about two

thirds. For email classification we build two models, one for ham emails and one for phishing emails. We extract four DMC features: the  $H(x,M)$ -values for each of the two models and two Boolean features indicating membership in either class.

We conducted experiments to evaluate the space savings achieved using the active learning approach for dynamic Markov models. We tested our approach for different adaptation rates  $\rho$  with a 10-fold cross-validation on the corpus Base07 and achieve good results over a wide range of rates as shown in Figure 3. The size of the models (i.e. the number of states) for phishing and ham decreases by a large amount whereas the classification quality (F-measure indicated by "\*") remains nearly constant. We conclude that the utilization of training examples that can be predicted sufficiently well is indeed not expedient. Even when using a small adaptation rate the model size decreases already by about two thirds. As the adaptation rate increases further the model size decreases only by a small amount. This indicates that any atypical message is usually very different from the model learned thus far and hence is used even for high adaptation rates. We think that our adaptive generation process comes close to heuristically estimating a good training set for the phishing email classification problem.



**Figure 3 Results for the Active Learning DMC approach. The left side shows the number of states of the DMC models for different adaption rates  $\rho$ . The corresponding F-measures indicated by "\*" is derived from the classification based solely on DMC features.**

## 2.5 Outlier Detection by Dynamic Markov Chains

Within the scope of the AntiPhish project we are interested in emails that are in some way different from emails we have encountered so far. Recognizing these outliers helps in several ways:

- It permits us to learn about new scams.
- It enables us to update filter software in a shorter time frame.
- It can point us to existing weaknesses of our filters.

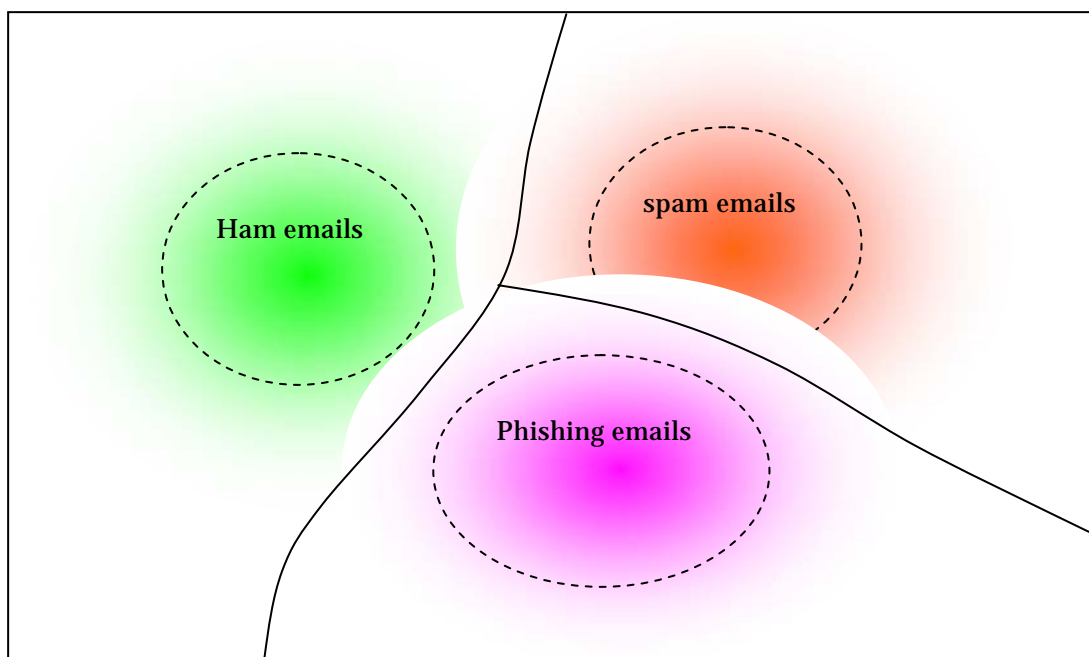
Many techniques exist to detect outliers. Using clustering one can identify items that are far from cluster centres. Using classifiers one can identify items that are far from the classification boundary.

The previously developed dynamic Markov chains are very suited for outlier detection. In this technique automata are automatically learnt to model the language of the different classes; ham, spam, and phishing. They operate on the bit level representation of the email. Then for a new incoming email it is detected to which automaton it fits best. Now if an email does not really fit to any automaton it can very well be called an outlier. Another advantage of this technique is that one can use different weights for the different classes, so that the definition of an outlier can be adapted.

Two problems arise when we use dynamic Markov chains for outlier detection. First, the scores of different automata are not directly comparable. Second, the scores for the emails from which the automata were constructed are much better than scores for arbitrary incoming emails, so that automatic identification of outlier score thresholds is difficult.

To overcome these problems we use a validation set of emails in addition to the training set. This reserved set of emails is used to compute a range of scores for each automaton. These ranges can then be used to better evaluate the emails in the test set with respect to their characteristics as outliers. Instead of the numeric scores produced by the automata we alternatively use the ranks of the emails to detect outliers. We can also easily associate weights with each individual automaton so that outliers can be weighted with respect to different classes.

Figure 4 schematically shows the different regions which are determined by the outlier procedure. Within the dashed regions the observed emails of each type are located. Outside of these regions we have outliers. The type of the outlier probably corresponds to the closest class.



**Figure 4: Schematic representing outliers of different kind. Within the dashed regions the observed emails are located. Outside the regions there are outliers, which in turn probably to the closest class.**

We applied this technique to two datasets, the public dataset used in the work of [Fette et al.(2007)] and a proprietary dataset from November 2006 (see section 3.1). Quantitative evaluation of this approach is difficult as we do not have emails labelled as outliers. Instead, we manually looked at the emails that were proposed as outliers by this technique. These emails fall into three categories:

- “binary” emails, e.g., images as inline binary sequences
- very short ham emails
- regular phishing emails

In general, all of those three categories fit the description of outliers if we assume outliers to be in some way not normal.

## 3 Evaluations of the AntiPhish System

### 3.1 Evaluations on Benchmark Data

For the evaluation of phishing filtering it is especially difficult to provide standardized publicly available data. Due to privacy regulations it is virtually impossible to obtain a representative corpus containing legitimate emails. We evaluate our system and the usefulness of our new features on a number of different corpora. The summary of the key figures of each used corpora is given in Table 1.

**Table 2 Characteristics of Benchmark Corpora**

Corpus	Size	Ham	Phishing
Base07	7808	6951 (89%)	857 (11%)
JNNEW	10653	6951 (65%)	3702 (35%)
JNFULL	11510	6951 (61%)	4559 (39%)
Nov06	3472	3156 (91%)	316 (9%)

The first corpus Base07 is the same corpus that was used by [Fette et al.(2007)]. The phishing emails were collected by Nazario and made publicly available on his website<sup>3</sup>. The ham emails were taken from the publicly available SpamAssassin corpus<sup>4</sup>. Recently, Nazario made more phishing emails available. This permits the construction of more comprehensive corpora. Nazario provides two new collections of phishing emails gathered between November 2005 and August 2007 and containing a total of 3702 emails. Using these additional phishing emails and the previously used SpamAssassin ham emails we constructed two new corpora JNNEW and JNFULL. For the corpus JNNEW we use the only the new phishing email collections, whereas the JNFULL contains all phishing emails publicly available from Nazario, including the ones used in the Base07 corpus.

[Fette et al.(2007)] point out that it would be desirable to perform experiments on data from a real-world mailbox. To this end, we use the dataset Nov06, which contains emails that were received during the month of November 2006 and manually labelled into ham and phishing. This dataset follows the expected distribution of ham and phishing emails in the mailbox of a common user.

Table 3 shows the basic classification results using 10- fold cross-validation and the complete set of features for our different corpora. The results for the extended corpora JNNEW and JNFULL are similar to the results for Base07. On the other hand we can observe that our results on the Nov06 corpus are somewhat inferior. We believe that the cause for this observation lies in the fact that the public corpora are somewhat artificial in that they are collected from diverse sources and even cover different time periods. Our real-life corpus contains ham and phishing emails from the same month and seems to provide a harder challenge. We looked into the misclassified emails. Among the false negatives there are many emails consisting of just an image, where the link to the

phishing website is behind the complete image. The false positives were often financial newsletters or account confirmation emails.

**Table 3 Classification Results for Different Benchmark Data Using All Features**

Corpus	Features	Accuracy	FP-Rate	FN-Rate	Precision	Recall	F-Measure
Base07	[Fette et al.(2007)]	99.49%	0.13%	3.62%	98.92%	96.38%	97.64%
Base07	All features	99.85%	0.01%	1.30%	99.88%	98.70%	99.29%
JNNEW	All features	99.61%	0.07%	0.99%	99.86%	99.01%	99.44%
JNFULL	All features	99.52%	0.07%	1.11%	99.89%	98.89%	99.39%
Nov06	All features	99.19%	0.16%	7.28%	98.32%	92.72%	95.44%

The results were published in different papers [Bergholz et al.(2008a)], [Bergholz et al.(2008b)], [Bergholz et al.(2009)] and were presented to different audiences [Paaß et al.(2008)] [Bergholz (2009)].

### 3.2 Evaluations in Field Experiments in an Industrial Setting

The third phase prototype APS has been intensively tested and deployed in differing real-world field experiments. One experiment evaluated APS on a normal email stream where the goal was to separate legitimate from unwanted emails. The other concerned the separation of phishing and regular-spam. A detailed description of these experiments and their results is provided in a separate report.

## 4 Conclusions

During the third phase the AntiPhish Filter System APS was extensively used in benchmark trials and field experiments. The software architecture proved to be very flexible and allowed easy specification of workflows involving many loops for training, testing, feature selection, active learning, etc. A number of new features and techniques were implemented and extensively tested.

The field experiments required many adaptations and trials, which easily could be done with the APS. The results of the experiments are very promising and may lead to an actual implementation of the APS in production workflows. As all loops and modules may be parallelised with the ProActive framework it is easily possible to extend the APS to a large throughput required in real-world applications.

## 5 References

- [Bergholz *et al.*(2008a)] A. Bergholz, J.-H. Chang, G. Paaß, F. Reichartz, and S. Strobel. Improved phishing detection using model-based features. In *Conference on Email and Antispam CEAS2008*, 2008.
- [Bergholz *et al.*(2008b)] A. Bergholz, G. Paaß, F. Reichartz, S. Strobel, M.-F. Moens, and B. Witten. Detecting known and new salting tricks in unwanted emails. In *Conference on Email and Antispam CEAS 2008*, 2008.
- [Bergholz *et al.*(2009)] Andre Bergholz, Jan De Beer, Sebastian Glahn, Marie-Francine Moens, Gerhard Paaß, and Siehyun Strobel. New filtering approaches for phishing email. *Journal of Computer Security*, page accepted for publication, 2009.
- [Bergholz (2009)] Andre Bergholz. AntiPhish - lessons learnt. Invited talk at the ACM SIGKDD Workshop on CyberSecurity and Intelligence Informatics (CSI-KDD 2009) at the KDD conference, June 2009.
- [Boulal (2009)] Anouar Boulal. Diversity in active learning with SVMs for multilabel text classification. Master's thesis, Computer Science Department, University of Bonn, 2009.
- [Bratko *et al.*(2006)] Andrej Bratko, Gordon V. Cormack, Bogdan Filipic, Thomas R. Lynam, and Blaz Zupan. Spam filtering using statistical data compression models. *Journal of Machine Learning Research*, 6:2673–2698, 2006.
- [Cormack and Horspool (1987)] Gordon V. Cormack and R. Nigel Horspool. Data compression using dynamic markov modelling. *The Computer Journal*, 30(6):541–550, 1987.
- [Fette *et al.*(2007)] Ian Fette, Norman Sadeh, and Anthony Tomasic. Learning to detect phishing emails. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 649–656, 2007.
- [Frank *et al.*(2000)] Eibe Frank, Chang Chui, and Ian H. Witten. Text categorization using compression models. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 200–209, 2000.
- [Lewis and Gale (1994)] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. pages 3–12, Dublin, Ireland, July 1994.
- [Marton *et al.*(2005)] Yuval Marton, Ning Wu, and Lisa Hellerstein. On compression-based text classification. In *Proceedings of the European Colloquium on IR Research (ECIR)*, pages 300–314, 2005.
- [Paaß *et al.*(2008)] Gerhard Paaß, Marc Dacier, and Domenico Dato. Panel on spam, phishing - a global perspective. Annual Computer Security Applications Conference (ACSAC) 2008, Dec 2008.
- [Tong and Koller (2000)] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, pages 999–1006, 2000.