Improved Phishing Detection using Model-Based Features

André Bergholz, Gerhard Paaß,
Frank Reichartz, Siehyun Strobel
Fraunhofer IAIS
Schloß Birlinghoven
53754 St. Augustin, Germany
firstname.lastname@iais.fhg.de

Jeong-Ho Chang
Konan Technology
Mirim Tower 825-22
Yeoksam-dong, Gangnam-gu,
Seoul 135-080, Korea
jeongho.chang@gmail.com

Abstract

Phishing emails are a real threat to internet communication and web economy. Criminals are trying to convince unsuspecting online users to reveal passwords, account numbers, social security numbers or other personal information. Filtering approaches using blacklists are not completely effective as about every minute a new phishing scam is created. We investigate the statistical filtering of phishing emails, where a classifier is trained on characteristic features of existing emails and subsequently is able to identify new phishing emails with different contents. We propose advanced email features generated by adaptively trained Dynamic Markov Chains and by novel latent Class-Topic Models. On a publicly available test corpus classifiers using these features are able to reduce the number of misclassified emails by two thirds compared to previous work. Using a recently proposed more expressive evaluation method we show that these results are statistically significant. In addition we successfully tested our approach on a non-public email corpus with a real-life composition.

1 Introduction

In the last years email traffic has shown a rapid expansion of phishing, the practice of luring users to fraudulent websites. Criminals are trying to convince unsuspecting online users to reveal passwords, account numbers, social security numbers or other personal information. To this end they send faked messages disguised as coming from reputable online businesses, such as financial institutions.

Phishing has increased enormously over the last months and is a serious threat to global security and economy. A recent in-lab experiment shows that a up to 90% of the users are fooled by good phishing websites [10]. To avoid extensive losses different authors have proposed to determine characteristic features of phishing emails. These features serve as inputs to statistical classification techniques, which are then trained to identify phishing emails.

In this paper we evaluate features derived from emails with respect to their ability to detect phishing scams. We develop two new model-based features that outperform basic features proposed in previous approaches. In particular, this paper makes the following contributions:

- We apply the Dynamic Markov chain compression to the phishing detection problem and propose an adaptive training algorithm that reduces memory requirements by about two thirds.
- We propose a novel target-specific latent Dirichlet topic model that is an extension of the standard LDA technique and performs better for topic numbers of up to 100.
- We show that classifiers trained using features extracted with these two techniques clearly outperform the previous benchmark.
- We employ a recently proposed evaluation strategy that allows to determine the significance of improvements and show that our improvement is indeed statistically significant.

The paper is organized as follows. Section 2 describes the current state of the art. In Section 3 we introduce statistical approaches to phishing filtering. The following Section 4 is devoted to the description of features, especially the two new model-based features not yet used for phishing email detection. Section 5 describes our feature processing and selection approach. In Section 6 we introduce the test corpora and the evaluation strategy. Section 7 presents results of the

empirical evaluation. Finally we summarize the results and draw some conclusions in Section 8.

2 Current State

In a typical phishing attack the phisher sends out emails pretending to come from a reputable institution, e.g., a bank. In general, two approaches to phishing prevention are possible: website and email filtering.

2.1 Website Filtering

One approach of phishing prevention concentrates on filtering websites when they are rendered in a web browsers. In Mozilla Firefox, for instance, each web page requested by a user is checked against a blacklist of known phishing sites [20]. This list is automatically downloaded to the local machine and updated in regular intervals. It is well-known, however, that new phishing sites appear frequently. In October 2007 about 46 new phishing sites were detected per hour [2]. The average time for phishing sites to be online is only 3.1 days; many sites disappear within hours. Therefore the effectiveness of blacklisting is limited. Whitelist approaches, which maintain a list of "good" URLs, have also been implemented. However, it turns out that it is very difficult to register large numbers of variants of legitimate sites.

To capture new phishing sites content-based filtering approaches may be used. Internet Explorer 7, for instance, offers a built-in classifier that filters web pages based on their characteristics. The accuracy of detecting phishing sites by statistical methods leaves room for improvement. Zhang et al. [22] report that they are able to catch about 90% of phishing sites with 1% false positives. Miyamoto et al. [19] yield a true positive rate of 94% with a false negative rate of 0% based on a limited sample of 100 websites.

WholeSecurity's Web Caller-Id[1] technology claims to detect 98% of phishing websites. It intercepts a web page, which it tests for spoof-site risk factors, including the URL, the depth of the website, and whether the domain name was recently registered. Then the page gets scored, and if it fails, a report gets sent to toolbar customers.

2.2 Email Filtering

In this paper we use an alternative source of filtering information, the content of the original email. Phishing emails usually contain specific phrases asking users to submit information or to access the phishing website. In conjunction these phrases often may be used as indicators for phishing and can be detected by filtering the content of emails.

In recent years spam filters have been widely discussed [13]. However, the identification of phishing emails is different from spam classification. A spammer simply wants to contact a user and inform him about some product while the phisher has to deliver a message, which has an insuspicious look and pretends to come from some reputable institution. Therefore many techniques used in spamming, like deliberate typos to defeat spam filters, usually do not appear in phishing emails. Hence, different features and techniques have to be employed if phishing emails are to be detected.

Recently, classifiers have been applied to phishing email identification. Chandrasekaran et al. [6] start with the following features: (1) a number of style marker features for emails, (2) structural attributes, and (3) the frequency distribution of selected function words (e.g., "click"). They evaluate these features for a small corpus of 400 emails and achieve good results. In contrast to the approach of many machine learning studies the authors do not evaluate different splits between training and testing data, which makes the significance of the results difficult to assess.

Abu-Nihmeh et al. [1] investigate the performance of different popular classifiers used in text mining, e.g., logistic regression, random forests and support vector machines. A public collection of about 1700 phishing emails and 1700 legitimate emails from private mailboxes is used. For this setup the random forest classifier yields the best result with an F-measure of 90%.

Fette et al. [11] follow a similar approach but use a larger publicly available corpus of about 7000 legitimate (ham) emails and 860 phishing emails. They propose ten different features to identify phishing scams. Nine of these features can be extracted from the email itself, while the tenth feature, the age of linked-to domain names, has to be obtained by a WHOIS query at the time the email is received. Another feature the authors include is the score of a publicly available spam-filter, the SpamAssassin [2]. By 10-fold cross-validation Fette et al. arrive at a false positive rate of 0.13% and a false negative rate of 3.6%, which corresponds to an F-measure of 97.6%.

3 Machine Learning Approach

Machine learning techniques, in particular automatic classification, have become popular in email spam

[1] http://www.webcallerid.net

[2] http://spamassassin.apache.org

and phishing detection. In contrast to manually constructed filter rules they automatically assess the relevance of input features $\mathbf{x} = (x_1, \ldots, x_m)$ (e.g., email characteristics) and establish a function to determine the desired classification y (e.g., phishing or non-phishing)

$$y = f(\mathbf{x}, \gamma)$$

The vector of unknown parameter values γ is determined in a training phase in such a way that the relation between $\mathbf{x}$ and y in the observed data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_D, y_D)$ 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.

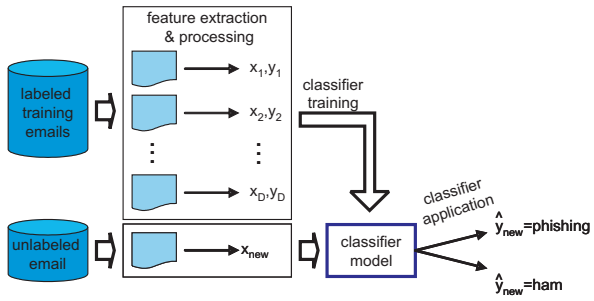

Figure 1: The machine learning approach

To assess a classifier's performance different quality metrics can be measured on an independent test set not used during the training phase.

4 Features for Email Data

In this section we present the features that serve as input to our classifiers. We start with basic features and continue with new advanced features based on dynamic Markov chains and latent topic models.

4.1 Basic Features

One main characteristic of phishing emails is that they aim at leading users to some website where they reveal private data. It is thus natural to look at the email structure and external links for the detection of phishing emails.

In this section we describe the basic features we use, which differ from the list proposed by Fette et al. [11] in two ways. First, our list is somewhat larger. We extract a total of 27 basic features. Second, 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 websites, such as the age of linked-to domains.

Structural Features (4) Structural features reflect the body part structure of an email. The MIME standard defines a number of possible message formats. We record four features: the total number of body parts, the number of discrete and composite body parts and the number of alternative body parts, which are different representations of the same content.

Link Features (8) Link features reflect various properties of links contained in an email. We record eight features: the total number of links, the number of internal and external links, the number of links with IP-numbers, the number of deceptive links (links where the URL visible to the user is different from the URL the link is pointing to), the number of links behind an image, the maximum number of dots in a link, and a Boolean indicating whether there is a link whose text contains one of the following words: click, here, login, update.

Element Features (4) Element features reflect what kinds of web technologies are used in an email. We record four Boolean features of whether HTML, scripting and in particular JavaScript, and forms are used.

Spam Filter Features (2) Next, we use an untrained, off-line version of SpamAssassin to generate two features, the score and a Boolean of whether or not an email is considered as spam. A message is considered spam if its score is greater than 5.0. It is important to note that we consider only features intrinsic to an email; we do not use any kinds of black- or whitelists. In a real-life scenario, the inclusion of such information would probably lead to performance improvements.

Word List Features (9) Finally, we use a positive word list, i.e., a list of words hinting at the possibility of phishing. For each word in the list we record a Boolean feature of whether or not the word occurs in the email. The list contains a total of nine word stems: account, update, confirm, verify, secur, notif, log, click, inconvenien.

In the remainder of this section we propose two kinds of advanced features. Both of these can be viewed as classifiers themselves, because they are based on models. The outputs of these models serve as features in our global email classification process.

4.2 Dynamic Markov Chain Features

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 [8], 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 [8]. These dynamic Markov chains have been successfully applied to text classification problems in various domains [17, 12]. 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. [5] 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) $H(\mathbf{x}, M)$ between the message $\mathbf{x}$ and the source approximated by the model M is a measure for the likelihood that a message $\mathbf{x}$ with the binary representation $(b_1 \dots b_n)$ originated from that source. The cross-entropy [9] is defined as:

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

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

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

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 [16].

Let $M_c^{1,k}$ be the model that is generated after processing the training examples $\mathbf{x}_1, \dots, \mathbf{x}_k$ of a class c. During the incremental model generation we think of $H(\mathbf{x}_i, M_c^{1,i-1})$ as the expected cross entropy of a training message $\mathbf{x}_i$ 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

$$\hat{\sigma}(k) = \sqrt{\frac{1}{k-2} \cdot \sum_{i=1}^{k-1} \left(H(\mathbf{x}_i, M_c^{1,i-1}) - \overline{H(x, M_c^{1,i-1})} \right)^2}$$

where $\overline{H(x, M_c^{1,i-1})} = \frac{1}{i-1} \sum_{j=1}^{i-1} H(\mathbf{x}_j, M_c^{1,j-1})$. 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 $\mathbf{x}_i$ for which the following equation holds:

$$H(\mathbf{x}_i, M_c^{1,i-1}) - \overline{H(x, M_c^{1,i-1})} > \rho \cdot \hat{\sigma}(i) \qquad (1)$$

with a given adaptation rate ρ. Our adaptive training uses a fixed percentage τ of the training data for the generation of an initial model and then applies the heuristic Equation 1 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.

4.3 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 generate such features by exploiting the co-occurrence of words in a training set of emails.

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

We first have to train the class-topic model using a training collection of emails annotated with classes. The estimated model may be applied to a new, uncategorized email and assigns a latent topic to each word of the email. The proportions of the different topics in the email then serve as input features for the classifier.

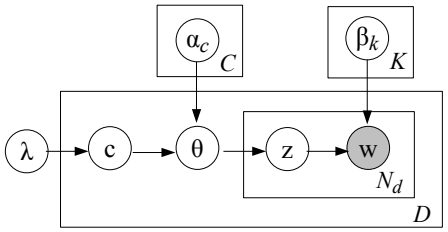

Figure 2: The graphical model of CLTOM

We now describe the class-topic model in more detail. The CLTOM assumes a probabilistic generative process for content of a message. We assume that there are C different email classes (e.g., phishing and non-phishing) and K different latent topics, where K is a parameter fixed in advance. First, a class indicator $c \in \{1, \ldots, C\}$ is sampled from a multinomial distribution parameterized by $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_C)$. Given class c a K-dimensional probability vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$ for the topic composition of the message is generated by a Dirichlet distribution parameterized by $\boldsymbol{\alpha}_c = \{\alpha_{c1}, \ldots, \alpha_{cK}\}$. Then, for each word position n in the message, a topic z_n is sampled from the multinomial $\boldsymbol{\theta}$. Finally a word w_n is randomly generated according to a topic-specific word distribution, which is a multinomial parameterized by $\boldsymbol{\beta}_{z_n}$ describing the probability $\beta_{z_n,w}$ for each possible word w in the set of all possible words $\mathcal{W}$. Under this assumption, the probability of the N words $\mathbf{w} = (w_1, \ldots, w_N)$ in an email is

$$p(\mathbf{w}|\lambda, \alpha, \beta) = p(c|\lambda) \int p(\theta|\boldsymbol{\alpha}, c) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \boldsymbol{\beta}) d\theta.$$

By the introduction of the class variable c and separate parameters $\boldsymbol{\alpha}_c$ for each class, the model is expected to capture more class-relevant semantic features compared with fully-unsupervised LDA. Note that the word order in a message content is not considered; a message is represented as a bag-of-words. Figure 2 shows the graphical model of CLTOM, which summarizes the stochastic process mentioned above.

The learning of the CLTOM is done using a variational EM algorithm [3]. For a corpus $\mathcal{D} = \{\mathbf{w}_1, \ldots, \mathbf{w}_D\}$, the bag-of-word representation of a collection of training examples $\mathbf{x}_1, \ldots, \mathbf{x}_D$, the log-likelihood is

$$\mathcal{L} = \log p(\mathcal{D}|\lambda, \alpha, \beta) = \sum_{d=1}^{D} \log p(\mathbf{w}_d|\lambda, \alpha, \beta). \qquad (2)$$

Here, the exact posterior inference for latent variables is intractable, so we employ a mean-field variational method for the inference of $\{(c_d, \theta_d, \mathbf{z}_d)\}_{d=1}^{D}$. For a message $\mathbf{w}_d$, the distribution of latent variables $(c_d, \theta_d, \mathbf{z}_d)$ is fully factorized as $q(c_d, \theta_d, \mathbf{z}_d) = q(c_d|\nu_d) q(\theta_d|\eta_d) \prod_{n=1}^{N_d} q(z_{dn}|\phi_{dn})$. Two kinds of variational parameters ν_d and ϕ_{dn} characterizes the posterior multinomial distribution for c_d and z_{dn}, and η_d is the variational parameter of the posterior Dirichlet $q(\theta_d|\eta_d)$. Then, the log-likelihood in (2) can be lower-bounded by

$$\begin{aligned}\mathcal{L} \geq & \sum_{d=1}^{D} \left(\mathbb{E}_{q_d} \left[\log(c_d|\lambda)\right] + \mathbb{E}_{q_d} \left[\log(\theta_d|c_d, \alpha)\right] \right. \\ & \left. + \mathbb{E}_{q_d} \left[\log(\mathbf{z}_d|\theta_d)\right] + \mathbb{E}_{q_d} \left[\log(\mathbf{w}_d|z_d, \beta)\right] + H(q_d) \right) \\ \equiv & \mathcal{F},\end{aligned}$$

where $q_d = q(c_d, \theta_d, \mathbf{z}_d)$ and $H(q_d)$ is the entropy term for the posterior distribution q_d.

We maximize the lower bound $\mathcal{F}$ based on the variational approximation. In the E-step of the variational EM algorithm, the distribution for hidden variables $q(c)$, $q(\theta)$, and $q(\mathbf{z})$ are estimated. Actually, $q(c_d)$ for training messages are fixed prior to learning, because the category information is already given for messages in training corpus. In the M-step, the model parameters λ, α, and β are determined. Two multinomial parameters λ and β can be calculated in closed form. The Dirichlet parameter α is estimated in an iterative way using a Newton-Raphson method [18], because there is no closed-form solution for the parameter. These steps are iteratively alternated until convergence.

When a new message $\mathbf{w}_m$ arrives, the semantic features for the message are derived through an inference in the trained CLTOM. In this application phase to a new message, unlike the training phase, $q(c|\nu_m)$ is also estimated along with $q(\theta|\eta_m)$ and $q(\mathbf{z}|\boldsymbol{\phi}_m)$, because the category information is not known for the newly arriving message. Finally, the posterior mean value

$\bar{\theta}_m = (\bar{\theta}_{m1}, \bar{\theta}_{m2}, \ldots, \bar{\theta}_{mK})$, $\bar{\theta}_{mk} = \eta_{mk} / \sum_l \eta_{ml}$, is calculated from the Dirichlet $q(\theta|\eta_m)$, and is provided as the *semantic feature* for the incoming message.

5 Feature Processing and Feature Selection

The features used in our system come from a variety of different sources. It is thus necessary to postprocess them to supply the classifiers with unified inputs.

5.1 Feature Processing

It is not advisable to directly use the unmodified feature values as input for a classifier. We perform scaling and normalization. Scaling guarantees that all features have values within the same range. Many classifiers are based on distance measures, such as the Euclidean distance, which overemphasizes features with large values. We perform a Z-transformation to ensure that all features have an empirical mean of 0 and an empirical standard deviation of 1. Additionally, we normalize the length of the feature vectors to one, which is adequate for inner-product based classifiers.

5.2 Feature Selection

In practice, machine learning algorithms tend to degrade in performance when faced with many features that are not necessary for predicting the correct label [14, 21]. The problem of selecting a subset of relevant features, while ignoring the rest, is a challenge that all learning schemes are faced with. Feature selection in a supervised learning setting can be understood as a search in a state space. In this space every state represents a feature subset. Operators that add and eliminate single features determine the connection between the states. Because there are $2^n - 1$ different non-empty subsets of n features, a complete search through the state space is impractical and heuristics need to be employed.

A common and widely used technique is the wrapper approach proposed by Kohavi et al. [15]: a search algorithm uses the classifier itself as part of the evaluation function. The classifier operates on an independent validation set; the search algorithm systematically adds and subtracts features to a current subset. In our experiments, we apply the so-called best-first search strategy, which expands the current node (i.e., the current subset), evaluates its children and moves to the child node with the highest estimated performance. To overcome local maxima, the most promising nodes that have not been expanded are maintained in a list and are considered if no improvement can be found. If in k expansions no improved node can be found, the search will be terminated. To avoid overfitting a node is considered an improvement if its estimation exceeds the estimation of the best node found so far by a factor of at least ϵ. In addition, we combine the best-first search engine with compound operators [15], which dynamically combine the set of best-performing children. The underlying idea is to shorten the search for each iteration by considering not only the information of the "best" child node (as described in the greedy approach above) but also the other evaluated children. More formally, by ranking the operators with respect to the estimated performance of the children a compound operator c_i can be defined to be the combination of the best $i+1$ operators.

6 Evaluation Criteria and Data

In this section we outline the evaluation method we used for testing our approach and we give details about the used data.

6.1 Evaluation Method

We use 10-fold cross-validation as our evaluation method and report a variety of evaluation measures. For each email four different scenarios are possible: true positive (TP, correctly classified phishing email), true negative (TN, correctly classified ham email), false positive (FP, ham email wrongly classified as phishing), and false negative (FN, phishing email wrongly classified as ham). For comparison reasons we report accuracy, i.e., the fraction of correctly classified emails. This measure is only of limited interest in a scenario where the different classes are very unevenly distributed.

More importantly, we report standard measures, such as precision, recall, and F-measure as well as the false positive and the false negative rate. These measures are defined as:

$$precision = \frac{|TP|}{|TP| + |FP|} \qquad recall = \frac{|TP|}{|TP| + |FN|}$$

$$f = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

$$fpr = \frac{|FP|}{|FP| + |TN|} \qquad fnr = \frac{|FN|}{|TP| + |FN|}$$

Note that 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.

For some of the experiments we additionally perform repeated (10 times) 10-fold cross-validation. It corrects the statistical dependency of samples in normal

cross-validation. This leads to a nearly unbiased estimate of the variability of the final performance measures and in addition to more precise average values as argued in [4]. In particular, they propose to use a corrected t-statistic. For an r-times k-fold cross-validation let x_{ij} denote the observed difference of evaluation in the j-th fold of the i-th repetition and let

$$\hat{\mu} = \frac{1}{r \cdot k} \sum_{i=1}^{r} \sum_{j=1}^{k} x_{ij}$$

be the empirical mean and

$$\hat{\sigma}^2 = \frac{1}{r \cdot k - 1} \sum_{i=1}^{r} \sum_{j=1}^{k} (x_{ij} - \hat{\mu})^2$$

be the empirical variance, then

$$t = \frac{\hat{\mu}}{\sqrt{(\frac{1}{r \cdot k} + \frac{n_2}{n_1})\sigma^2}}$$

gives the corrected t-statistic with $df = r \cdot k - 1$ degrees of freedom. In the formula n_1 denotes the number of instances in the training set and n_2 the number of instances in the test set.

6.2 Test Corpora

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.

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%)

Table 1: Summary of the used corpora

The first corpus `Base07` is the same corpus that was used by Fette et al. [11]. The phishing emails were collected by Nazario and made publicly available on his website[3]. The ham emails were taken from the publicly available SpamAssassin corpus[4].

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. 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 labeled into ham and phishing. This dataset follows the expected distribution of ham and phishing emails in the mailbox of a common user.

7 Empirical Results

In our experiments we have used different classifiers, mainly the Support Vector Machine (SVM) classifier implemented in the libSVM-library [7]. The RBF kernel with parameters $C = 10$ and $\gamma = 0.1$ turned out to be most accurate and stable. We have also run experiments using other classifiers, e.g., maximum entropy and decision trees. The difference in most cases were negligible.

7.1 Added Value of Advanced Features

Table 2 shows the basic classification results using 10-fold cross-validation and the complete set of features for the `Base07` corpus and results for different subsets of features. The complete set of features consists of 81 features, 27 basic features plus $K = 50$ topic features plus four DMC features.

We can see that the error in terms of F-measure, i.e., $1 - f$, is reduced by almost 70% when compared to the work of Fette et al. The very low false positive rate indicates that virtually no legitimate email is lost. In addition, the more statistically founded evaluation approach using 10-times 10-fold cross-validation gives an average F-measure of 99.22% for our approach with an empirical variance of $5.34 \cdot 10^{-5}$. This leads to a corrected t-statistic of 6.221. If we assume that the result of Fette et al. comes from a distribution with the same variance, then we arrive at paired t-statistic of 3.111, which indicates that our results, for 99 degrees of freedom, are statistically significant with a probability of 99.88%.

Our system performs worse than the original system by Fette et al. if we restrict ourselves to our basic set of features. We believe the reason is that we do not use extrinsic features, such as the age of the linked-to domain. Such features are undoubtedly useful, but

[3] http://monkey.org/~jose/wiki/doku.php?id=phishingcorpus
[4] http://spamassassin.apache.org/publiccorpus/

Features	Accuracy	FP-Rate	FN-Rate	Precision	Recall	F-Measure	Error Red.
[Fette et al.]	99.49%	0.13%	3.62%	98.92%	96.38%	97.64%	–
All features	99.85%	0.01%	1.30%	99.88%	98.70%	99.29%	69.92%
Basic features	99.13%	0.20%	6.39%	98.26%	93.61%	95.88%	-74.58%
DMC features	99.56%	0.00%	4.02%	100.00%	95.98%	97.95%	13.14%
Topic features ($K = 50$)	99.53%	0.20%	2.72%	98.33%	97.28%	97.80%	6.78%
DMC + Topic features	99.77%	0.03%	1.89%	99.76%	98.11%	98.93%	54.66%
Feature selection	99.88%	0.00%	1.07%	100.00%	98.93%	99.46%	77.12%

Table 2: Classification results for different feature sets on corpus `Base07`

make repetition of experiments hard if not impossible. In addition we performed a simple experiment out of curiosity. We trained a classifier on one basic feature only: the number of deceptive links. This feature intuitively seems to be the most characteristic, however, the results were disappointing. While the false positive rate was surprisingly low at 1.37% the false negative rate was 67.46% indicating that many phishing emails were missed.

The table also indicates that each of the two new feature groups alone surpasses the performance of the Fette et al. system by a small margin. Both of the new feature groups combined already clearly outperform the previous system. In particular, the false positive rate is already close to perfect. In other words, the addition of the basic features improves only the false negative rate. The basic features capture some of the missed phishing emails. We believe the reason for that observation is that the advanced features are based purely on the textual content of the email. Structural aspects helping to identify some of the phishing emails are added through the basic features. This argument is supported by our observations from the feature selection experiments.

7.2 Feature Selection

We performed feature selection as described in Section 5.2 and set the parameters $k = 5$ and $\epsilon = 0.1\%$ as suggested in [15]. In the cross-validation experiments we used feature selection on each individual fold to make the result more comparable to related work. Note that we no longer can use 90% of the data for training in each individual fold, because we reserve 20% of the training data for the feature selection process, i.e., for the validation of individual models based on different feature sets. As shown also in Table 2 we achieved an even better result using fewer features and less training data.

We also observed which features were selected in the process. Besides our DMC and topic features some structural features (total number of body parts, number of discrete and composite body parts), some link features (total number of links, number of deceptive links, number of image links) and the two SpamAssassin features placed among the selected ones. This seems reasonable as the structural features are complementary to the purely content-based advanced features. In addition, the rule-based SpamAssassin algorithm also covers a lot of different aspects of an email.

7.3 Different Test Corpora

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. Another typical false negative is the eBay item request, in which the phishing link is hidden behind some "Click here to answer the request."-statement. The false positives were often financial newsletters or account confirmation emails.

7.4 Further Investigation of Advanced Features

We conducted experiments to evaluate the space savings achieved using the adaptation approach for dynamic Markov models. We tested our approach for different adaptation rates ρ 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 model decreases by a large amount whereas the classification quality remains nearly constant. We conclude that the utilization of training examples that can be predicted sufficiently well is in-

Corpus	Features	Accuracy	FP-Rate	FN-Rate	Precision	Recall	F-Measure
Base07	[Fette et al.]	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%

Table 3: Classification results using all features

deed 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. Further experiments have to be conducted to show if our novel approach could also be applied to other text classification problems.

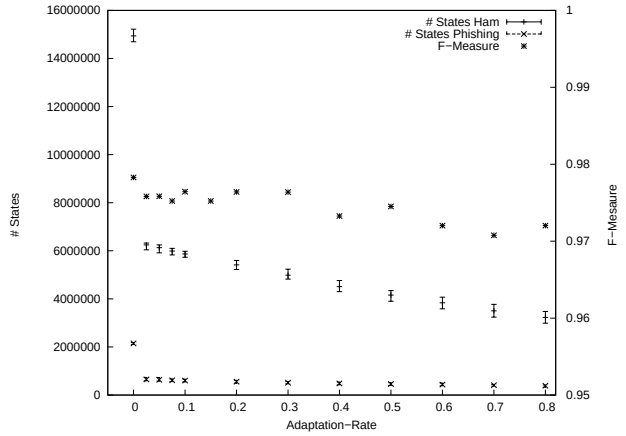

Figure 3: Results for the adaptive DMC approach. The F-measure is derived from the classification based solely on DMC features.

We also conducted experiments on the latent topic features to detect the influence of the parameter K indicating the number of topics and to compare our new CLTOM technique to the well-established LDA. The results depicted in Table 4 indicate that an increase in the number of topics leads to an improved overall classification performance. More phishing-specific topics can be extracted by the increase of the total number of topics in the model. One can also see that CLTOM outperforms LDA for lower topic numbers, whereas LDA catches up for larger topic numbers. The reason for that, we believe, is that LDA only estimates one Dirichlet parameter whereas CLTOM estimates K times the number of classes Dirichlet parameters. Hence, for very large topic numbers CLTOM is prone to overfitting. For reasonable topic numbers, however, the benefits of including class-relevant information are noticeable. In addition, Table 5 shows some of the phishing-related topics, represented by their ten most probable words.

K	Prec.	Recall	F	F (LDA)
5	94.59%	95.27%	94.93%	92.00%
10	96.82%	97.16%	96.99%	94.71%
25	96.94%	97.63%	97.29%	96.67%
50	98.33%	97.28%	97.80%	97.08%
100	99.16%	97.75%	98.45%	97.98%

Table 4: CLTOM-results and LDA comparison for different topic numbers

$\alpha = 1.5430$	$\alpha = 1.0192$	$\alpha = 0.9542$	$\alpha = 0.7230$
please	security	information	ebay
account	paypal	help	policy
update	secure	link	updated
click	e-mail	access	records
online	protect	following	agreement
card	accounts	provide	privacy
receive	account	limited	user
thank	password	personal	department
bank	team	protection	trademarks
customer	protecting	complete	suspended

Table 5: Example of phishing-related topics extracted by CLTOM with K=50. Higher values of the Dirichlet parameter α indicate higher relevance for the phishing class.

8 Conclusion

We employ statistical classification methods to classify emails as legitimate (ham) or phishing emails. We introduce two new types of features generated by adaptive Dynamic Markov Chains (DMC) and by latent Class-Topic Models (CLTOM). Our adaptive DMC approach reduces the memory requirements compared to the standard DMC approach by two thirds almost

without any loss in performance. Our CLTOM approach, which incorporates class-specific information into the topic model, outperforms the standard LDA approach for topic numbers of up to 100. Classifiers incorporating these features as input are able to substantially outperform previous approaches on publicly available benchmark corpora.

Due to privacy regulations it is extremely difficult to obtain a representative set of legitimate and phishing emails from real users. We tested our approach on another non-public email corpus with a more realistic composition. Here the level of accuracy turned out to be a bit lower. We plan to obtain additional features by analyzing attached images and by classifying the content of the linked-to websites.

References

[1] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair. A comparison of machine learning techniques for phishing detection. In *Proceedings of the eCrime Researchers Summit*, 2007.

[2] Anti-Phishing Working Group. Phishing activity trends - report for the month of October 2007, 2008. http://www.antiphishing.org/reports/apwg_report_oct_2007.pdf, accessed on 25.01.08.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] R. R. Bouckaert and E. Frank. Evaluating the replicability of significance tests for comparing learning algorithms. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 3–12, 2004.

[5] A. Bratko, G. V. Cormack, B. Filipic, T. R. Lynam, and B. Zupan. Spam filtering using statistical data compression models. *Journal of Machine Learning Research*, 6:2673–2698, 2006.

[6] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya. Phishing email detection based on structural properties. In *Proceedings of the NYS Cyber Security Conference*, 2006.

[7] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[8] G. V. Cormack and R. N. Horspool. Data compression using dynamic markov modelling. *The Computer Journal*, 30(6):541–550, 1987.

[9] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, 1991.

[10] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 581–590, 2006.

[11] 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, 2007.

[12] E. Frank, C. Chui, and I. H. Witten. Text categorization using compression models. In *Proceedings of the IEEE Data Compression Conference (DCC)*, pages 200–209, 2000.

[13] J. Goodman, G. V. Cormack, and D. Heckerman. Spam and the ongoing battle for the inbox. *Communications of the ACM*, 50:25–33, 2007.

[14] G. H. John. *Enhancements to the Data Mining Process*. PhD thesis, Stanford University, CA, USA, 1997.

[15] R. Kohavi and G. John. Wrappers for feature subset selection. *AI Journal*, 97(1-2):273–324, 1997.

[16] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. pages 3–12, Dublin, Ireland, July 1994.

[17] 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, 2005.

[18] T. P. Minka. Estimating a Dirichlet distribution. Technical report, Microsoft Research, 2003.

[19] D. Miyamoto, H. Hazeyama, and Y. Kadobayashi. A proposal of the AdaBoost-based detection of phishing sites. In *Proceedings of the Joint Workshop on Information Security*, 2007.

[20] Mozilla. Phishing protection, 2007. http://www.mozilla.com/en-US/firefox/phishing-protection/; accessed on 23.10.07.

[21] S. Thrun et al. The MONK's problems: A performance comparison of different learning algorithms. Technical Report CMU-CS-91-197, Carnegie Mellon University, Pittsburgh, PA, 1991.

[22] 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)*, 2007.