Detection of Phishing Attacks: A Machine Learning Approach

Ram Basnet, Srinivas Mukkamala, and Andrew H. Sung

New Mexico Tech, New Mexico 87801, USA
{ram,srinivas,sung}@cs.nmt.edu

1 Introduction

Phishing is a form of identity theft that occurs when a malicious Web site impersonates a legitimate one in order to acquire sensitive information such as passwords, account details, or credit card numbers. Though there are several anti-phishing software and techniques for detecting potential phishing attempts in emails and detecting phishing contents on websites, phishers come up with new and hybrid techniques to circumvent the available software and techniques.

Phishing is a deception technique that utilizes a combination of social engineering and technology to gather sensitive and personal information, such as passwords and credit card details by masquerading as a trustworthy person or business in an electronic communication. Phishing makes use of spoofed emails that are made to look authentic and purported to be coming from legitimate sources like financial institutions, ecommerce sites etc., to lure users to visit fraudulent websites through links provided in the phishing email. The fraudulent websites are designed to mimic the look of a real company webpage.

The phishing attacker's trick users by employing different social engineering tactics such as threatening to suspend user accounts if they do not complete the account update process, provide other information to validate their accounts or some other reasons to get the users to visit their spoofed web pages.

Why is it important to tackle the problem of phishing? According to the Anti-Phishing Working Group, there were 18,480 unique phishing attacks and 9666 unique phishing sites reported in March 2006. Phishing attacks affect millions of internet users and are a huge cost burden for businesses and victims of phishing (Phishing 2006). Gartner research conducted in April 2004 found that information given to spoofed websites resulted in direct losses for U.S. banks and credit card issuers to the amount of $1.2 billion (Litan 2004). Phishing has become a significant threat to users and businesses alike.

Over the past few years, much attention has been paid to the issue of security and privacy. Existing literature dealing with the problem of phishing is scarce. Fette et al proposed a new method for detecting phishing emails by incorporating features specific to phishing (Fette et al. 2006).

We applied different methods for detecting phishing emails using known as well as new features. We employ a few novel input features that can assist in discovering phishing attacks with very limited a-prior knowledge about the adversary or the method used to launch a phishing attack. Our approach is to classify phishing emails by incorporating key structural features in phishing emails and employing different

B. Prasad (Ed.): Soft Computing Applications in Industry, STUDFUZZ 226, pp. 373–383, 2008.
springerlink.com

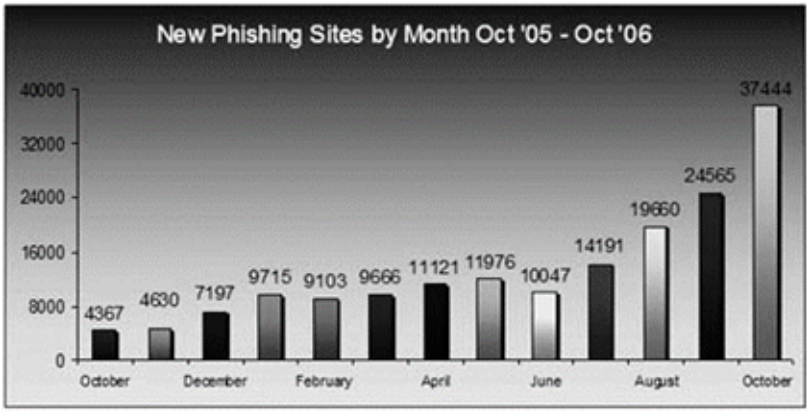

Fig. 1. Unique phishing site URLs rose 757 percent in one year

machine learning algorithms to our dataset for the classification process. The use of machine learning from a given training set is to learn labels of instances (phishing or legitimate emails). Our paper provides insights into the effectiveness of using different machine learning algorithms for the purpose of classification of phishing emails.

Soft Computing techniques are increasingly being used to address a gamut of computational problems. Clustering is a type of unsupervised learning; unsupervised learning assumes that there is no previous knowledge about the class membership of the observations, i.e., class labels of data is unknown. The purpose of using unsupervised learning is to directly extract structure from a dataset without prior training. Although, supervised learning provides for a much better accuracy, unsupervised learning provides for a fast and reliable approach to derive knowledge from a dataset.

This paper is organized as follows: In section 2 we provide an overview of features used in our experiments. In section 3 we present the data used. In section 4 we describe and present the different experiments conducted and also present the performance results of various machine learning algorithms. Finally, in section 5 we provide concluding remarks.

2 Features Used

There exist a number of different structural features that allow for the detection of phishing emails. In our approach, we make use of sixteen relevant features. The features used in our approach are described below.

- HTML Email: HTML-formatted emails are mainly used for phishing attacks, because plaintext emails do not provide for the scale of tricks afforded with HTML-formatted emails. Hyperlinks are active and clickable only in html-formatted emails. Thus, a HTML-formatted email is flagged and is used as a binary feature.

- IP-based URL: One way to obscure a server's identity is achieved through the use of an IP address. Use of an IP address makes it difficult for users to know exactly where they are being directed to when they click the link. A legitimate website usually has a domain name for its identification. Phishers usually use some zombie systems to host phishing sites. When a link in an email contains a link whose host is an IP address (for example, *http://81.215.214.238/pp/*) we flag the email and is used as a binary feature.
- Age of Domain Name: The domain names (if any) used by fraudsters are usually used for a limited time frame to avoid being caught. We can thus use this feature to flag emails as phishing based on the fact that the domain is newly registered and set a criteria of being new if it is less than 30 days old. This can be achieved by performing a WHOIS query on the domain name in the link. A WHOIS query provides other information such as the name or person to which the domain is registered to, address, domain's creation and expiration dates etc. This feature is a binary.
- Number of Domains: We make use of the domain names in the links that we extract and do a count of the number of domains. Two or more domain names are used in an URL address to forward address from one domain to the other. *http://www.google.com/url?sa=t&ct=res&cd=3&url=http%3A%2F%2Fwww.antiphishing.org%2F&ei=-0qHRbWHK4z6oQLTm-BM&usg=uIZX_3aJvESkMveh4uItI5DDUzM=&sig2=AVrQFpFvihFnLjpnGHVsxQ* for instance has two domain names where google.com forwards the click to URL antihphishing.org domain name. The number of domains we count is considered a continuous feature.
- Number of Sub-domains: Fraudsters make use of sub domains to make the links look legitimate. Having sub domains means having an inordinately large number of dots in the URL. We can make use of this feature to flag emails as phishing emails. For instance, *https://login.personal.wamu.com/verification.asp?d=1* has 2 sub domains. This is a continuous feature.
- Presence of JavaScript: JavaScript is usually employed in phishing emails, because it allows for deception on the client side using scripts to hide information or activate changes in the browser. Whenever an email contains the string "JavaScript", we flag it as a phishing email and use it as a binary feature.
- Presence of Form Tag: HTML forms are one of the techniques used to gather information from users. An example below shows the use of form tag in an email. An email supposedly from Paypal may contain a form tag which has the action attribute actually sending the information to *http://www.paypal-site.com/profile.php* and not to *http://www.paypal.com*. The email used for collecting user's info has form tag *<FORM action=http://www.paypal-site.com/profile.php method=post>* for example.
- Number of Links: Most often phishing emails will exploit the use of links for redirection. The number of links in email is used as a feature. A link in an email is one that makes use of the "href" attribute of the anchor tag. This feature will be continuous.
- URL Based Image Source: To make the phishing emails look authentic, images and banner of real companies are used in the emails. Such images are usually linked from the real companies' web pages. Thus, if any of the emails make use of such URL based images we flag it as a phishing email. This feature is binary.

- Matching Domains (From & Body): We make use of the information from the header of the email and match it with the domains in the body of the email. Most phishing emails will have different domains in the header and the body part. We will thus flag emails that have mismatching domain information. For example: The 'From' information in the header part of the email will show the email originating from "someone@paypal-site.com", while the body will have actual (*"http://www.paypal.com"*) company's domain for an authentic look. This feature is binary.
- Keywords: Phishing emails contain number of frequently repeated keywords such as suspend, verify, username, etc. We use word frequency (Count of keyword divided by total number of words in an email) of a handful of most commonly used keywords by phishers. This feature is continuous.
- Some handful of keywords if present in emails are counted and normalized. Group of words with similar meaning or synonyms are used as a single feature. We use six groups of keywords as six separate features. Six groups of keywords that we have used as features are listed below:
  - Update, Confirm
  - User, Customer, Client
  - Suspend, Restrict, Hold
  - Verify, Account
  - Login, Username, Password
  - SSN, Social Security

3 Data Used

To implement and test our approach, we have used two publicly available datasets i.e., the ham corpora from the SpamAssassin project as legitimate emails and the emails from PhishingCorpus as phishing emails (Phishing 2006, Spam 2006). The total number of emails used in our approach is 4000. Out of which 973 are used as phishing emails and 3027 as legitimate (ham) emails. The entire dataset is divided into two parts for testing and training purpose. A total of 2000 emails are considered as training samples and the remaining are considered for testing purpose. The tabular form of different samples used:

Table 1. Data used for experiments

Total samples	4000
Total phishing emails	973
Total legitimate emails	3027
Total training samples	2000
Total testing samples	2000

We used Python and Java scripts to parse the phishing and legitimate (ham) emails and extract the features mentioned above in section 3. We have a total of sixteen attributes for each email relation.

4 Experiments

To evaluate our implementation, we used different machine learning methods and a clustering technique on our phishing dataset. We used Support Vector Machines (SVM, Biased SVM & Leave One Model Out), Neural Networks, Self Organizing Maps (SOMs) and K-Means on the dataset described in section 3.

4.1 Model Selection of Support Vector Machines (SVMs)

In any predictive learning task, such as classification, both a model and a parameter estimation method should be selected in order to achieve a high level of performance of the learning machine. Recent approaches allow a wide class of models of varying complexity to be chosen. Then the task of learning amounts to selecting the sought-after model of optimal complexity and estimating parameters from training data (Chapelle 1999, Cherkassy 2002, Lee 2000).

Within the SVMs approach, usually parameters to be chosen are (i) the penalty term C which determines the trade-off between the complexity of the decision function and the number of training examples misclassified; (ii) the mapping function Φ; and (iii) the kernel function such that

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \tag{1}$$

In the case of RBF kernel, the width, which implicitly defines the high dimensional feature space, is the other parameter to be selected (Chapelle 1999).

We performed a grid search using 5-fold cross validation for each of the faults in our data set. We achieved the search of parameters C and γ in a coarse scale.

4.2 Biased Support Vector Machines (BSVMs)

Biased support vector machine (BSVM), a decomposition method for support vector machines (SVM) for large classification problems (Chan 2004). BSVM uses a

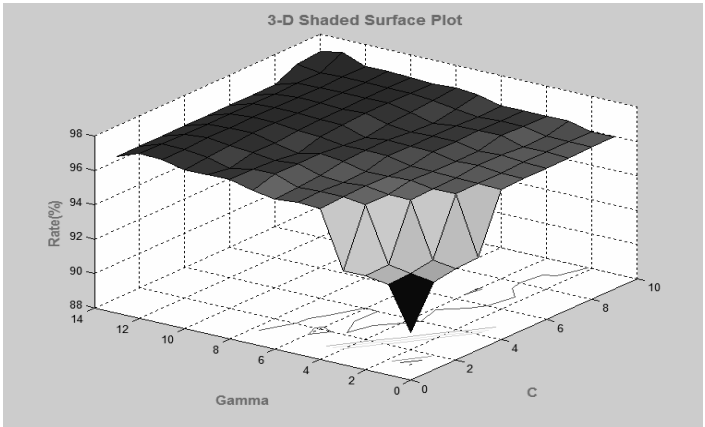

Fig. 2. 3-D view of accuracy for different Gamma and C pairs

decomposition method to solve a bound-constrained SVM formulation. BSVM Uses a simple working set selection which leads to faster convergences for difficult cases and a bounded SVM formulation and a projected gradient optimization solver which allow BSVM to quickly and stably identify support vectors.

Leave-one-out model selection for biased support vector machines (BSVM) is used for automatic model selection (Chan 2004). Model selection results BSVM using LOOMS are given in figure 2.

4.3 Neural Networks

Artificial neural network consists of a collection of processing elements that are highly interconnected and transform a set of inputs to a set of desired outputs. The result of the transformation is determined by the characteristics of the elements and the weights associated with the interconnections among them. A neural network conducts an analysis of the information and provides a probability estimate that it matches with the data it has been trained to recognize. The neural network gains the experience initially by training the system with both the input and output of the desired problem. The network configuration is refined until satisfactory results are obtained. The neural network gains experience over a period as it is being trained on the data related to the problem. Since a (multi-layer feedforward) ANN is capable of making multi-class classifications, a single ANN (Scaled Conjugate Gradient), is employed for classification, using the same training and testing sets.

4.3.1 Scaled Conjugate Gradient Algorithm

The scaled conjugate gradient algorithm is an implementation of avoiding the complicated line search procedure of conventional conjugate gradient algorithm (CGA). According to the SCGA, the Hessian matrix is approximated by

$$E''(w_k)\,p_k = \frac{E'(w_k + \sigma_k p_k) - E'(w_k)}{\sigma_k} + \lambda_k p_k \qquad (2)$$

where E' and E'' are the first and second derivative information of global error function $E(w_k)$. The other terms p_k, σ_k and λ_k represent the weights, search direction, parameter controlling the change in weight for second derivative approximation and parameter for regulating the indefiniteness of the Hessian. In order to get a good quadratic approximation of E, a mechanism to raise and lower λ_k is needed when the Hessian is positive definite (Moller 1993).

We ran experiments for 17 times with the same neural network settings. The first experiment was run using all the 16 features and we got 97.8% accuracy. Then for each experiment we removed one feature. So, each 16 experiment had one less feature (total 15) starting from feature no. 1, no. 2 so on and so forth. The following graph shows the result of accuracy for each experiment with one less feature. The motivation for doing so was to see which set of 15 features produces the highest accuracy which in turn might help us do some sort of feature selection. Results using different features are summarized in figure 3.

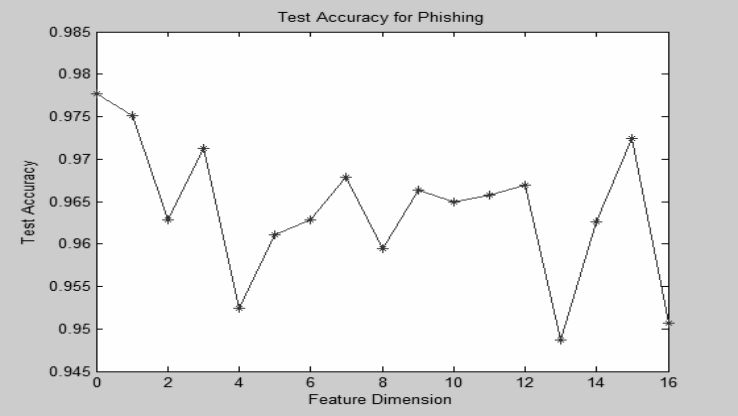

Fig. 3. Neural network results using different features

4.3.2 Self Organizing Maps (SOMs)

Self-Organizing Map (SOM) is an unsupervised algorithm that performs clustering of input data and maps it to a two-dimensional map. In the map, the similar data items will be mapped to nearby locations on the map (Vesanto 1999).

SOM-based analysis can be done using a visualization technique called the U-matrix, which shows the cluster structure of the map. High values of the U-matrix indicate a cluster border. While, uniform areas of low values indicate the clusters themselves. The component planes are also useful for visualizing the different components in the reference vectors. For visualizing the SOM of the phishing data, we took a sample of 200 emails (consisting of 50% legitimate and 50% phishing emails).

From the U-matrix we can see that the top rows of the SOM form a clear cluster. From the labels we can see that this corresponds to the legitimate emails (as represented by 'h'). The phishing emails form the other visible cluster on the lower part of the SOM. This is also indicated by the labels as phishing emails (as represented by 'p').

From the 16 component plane figures of Figure 4, we can visualize the clustering patterns for the different features. The highlights can be summarized as following:

1. Html Email: Almost all phishing emails are HTML-based
2. IP based URL: High values for phishing emails
3. Age of domain name: A small cluster concentrated on phishing emails
4. No of Domains: Very few of the emails (both phishing and legitimate had URL directed)
5. Max Sub Domains: This feature had prominence in phishing emails
6. Presence of JavaScript: Small cluster concentrated on phishing emails
7. Presence of Form Tag: Small cluster distributed evenly between phishing and legitimate emails
8. Number of Links: This feature had prominence in phishing emails
9. Image Source URL: This feature had prominence in phishing emails
10. Domain matching (From and Body): This feature had prominence in phishing emails.

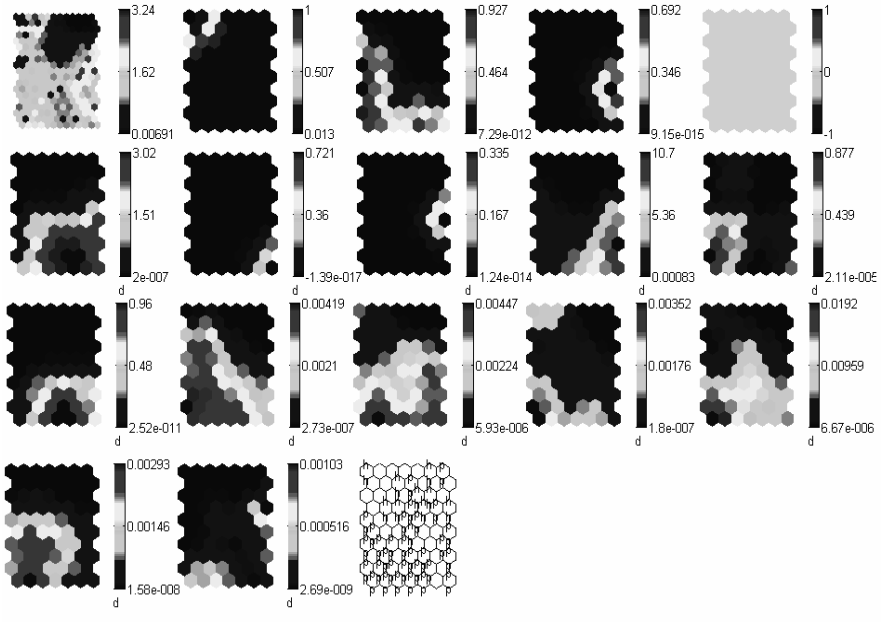

Fig. 4. Visualization of the SOMs for phishing data

11. Features 11 to 16 Keywords: The keywords features had very high prominence in phishing emails

The map unit labels also clearly show the clustering patterns. This complements the cluster pattern seen in the U-matrix. The U-matrix is shown on the top left. The other 16 figures following the U-matrix are the component planes. The bottom right figure is the map unit labels.

4.4 K-Means

K-means clustering is an unsupervised non-hierarchal clustering. This attempts to improve the estimate of the mean of each cluster and re-classifies each sample to the cluster with nearest mean. Practical approaches to clustering use an iterative procedure, which converges to one of numerous local points. These iterative techniques are sensitive to initial starting conditions. The refined initial starting condition allows the iterative algorithm to converge to a "better" local point. The procedure is being used in k-means clustering algorithm which being used for both discrete and continuous data points. Let us consider a n example feature vectors $\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_n$ all from the same class, and we know that they fall into k compact clusters, k < n. Let $\mathbf{m}_i$ be the mean of the vectors in Cluster I. If the clusters are well separated, we can use a minimum-distance classifier to separate them. That is, we can say that $\mathbf{x}$ is in Cluster i if $\| \mathbf{x} - \mathbf{m}_i \|$ is the minimum of all the k distances (Witten 2005).

Each cluster then creates a centroid frequency distribution. Each instance is then iteratively reassigned to the cluster with the closest centroid. When instances stop moving between clusters, the iteration process also stops.

K-Means aims at minimizing the objective function below (Anderberg 1973):

$$J = \sum_{j=1}^{k} \sum_{i=1}^{x} \| x_i^{(j)} - C_j \|^2 \quad (3)$$

where $\| x_i^{(j)} - C_j \|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ Cluster center C_j, is an indicator of the distance of the *n* data points from Ftheir respective cluster centers.

Table 2. Accuracies of K-Means on phishing dataset

Clustering Technique	Incorrectly Clustered Instances	Incorrectly Clustered Instances %	Accuracy %
K-means	635.0	9.2109 %	90.7891%

5 ROC Curves (SVMs)

ROC is a graphical plot between the sensitivity and specificity. The ROC is used to represent the plotting of the fraction of true positives (TP) versus the fraction of false positives (FP).

The point (0,1) is the perfect classifier, since it classifies all positive cases and negative cases correctly. Thus an ideal system will initiate by identifying all the

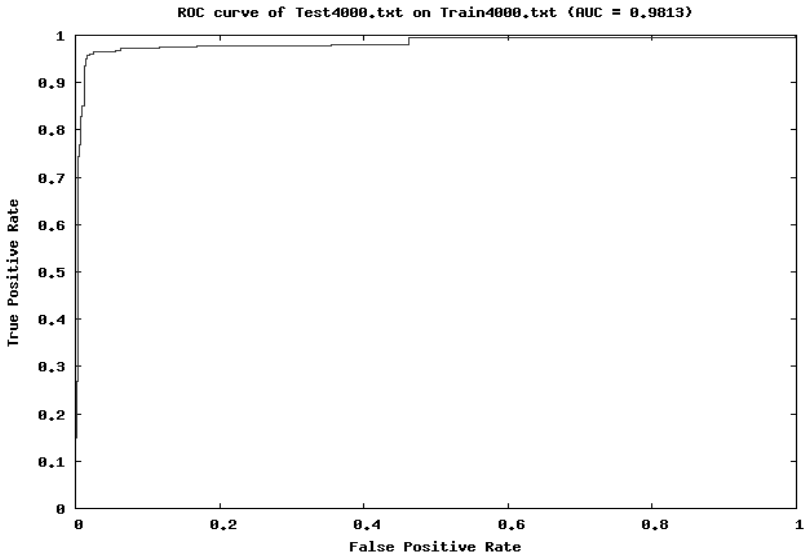

Fig. 5. Phishing attack detection accuracy using SVMs

positive examples and so the curve will rise to (0,1) immediately, having a zero rate of false positives, and then continue along to (1,1). Detection rates and false alarms are evaluated for the phishing data set and the obtained results are used to form the ROC curves. In each of these ROC plots, the x-axis is the false alarm rate, calculated as the percentage of normal emails considered as phishing attacks; the y-axis is the detection rate, calculated as the percentage of phishing attacks detected. A data point in the upper left corner corresponds to optimal high performance, i.e, high detection rate with low false alarm rate (Egan 1975).

The accuracy of the test depends on how well the test classifies the group being tested into 0 or 1. Accuracy is measured by the area under the ROC curve (AUC). An Area of 1 represents a perfect test and an area of .5 represents a worthless test. In our experiment, we got an **AUC of 0.9813** as shown in Figure 5.

6 Summary and Future Work

Although the performance of six different machine learning methods used is comparable, we found that Support Vector Machine (LIBSVM) achieved consistently the best results. Biased Support Vector Machine (BSVM) and Artificial Neural Networks gave the same accuracy of 97.99%.

We have added new features to what researchers have published in literature. The classifiers used in this paper showed comparable or better performance in some cases when compared to the ones reported in the literature using the same datasets. Our results demonstrate the potential of using learning machines in detecting and classifying phishing emails. As a future work we plan to use more machine learning algorithms to compare accuracy rates. We also plan to do a thorough feature ranking and selection on the same data set to come up with the set of features that produces the best accuracy consistently by all the classifiers.

Acknowledgements

Support for this research received from ICASA (Institute for Complex Additive Systems Analysis, a division of New Mexico Tech), and DOD IASP Capacity Building grant is gratefully acknowledged.

References

Anti-Phishing Working Group (2006) Phishing Activity Trends Report. http://www.antiphishing.org/reports/apwg_report_mar_06.pdf

Litan A (2004) Phishing Attack Victims Likely Targets for Identity Theft. Gartner Research

Fette I, Sadeh N, Tomasic A (2006) Learning to Detect Phishing Emails. Technical Report CMU-ISRI-06-112. Institute for Software Research International, Carnegie Mellon University

Phishing Corpus (2006) http://monkey.org/~jose/wiki/doku.php?id=PhishingCorpus

Spam Assassin (2006) http://spamassassin.apache.org/

Anti-Phishing Working Group, Phishing Activity Trends Report (2006), http://www.antiphishing.org/reports/apwg_report_mar_06.pdf

Litan, A.: Phishing Attack Victims Likely Targets for Identity Theft. Gartner Research (2004)

Fette, I., Sadeh, N., Tomasic, A.: Learning to Detect Phishing Emails. Technical Report CMU-ISRI-06-112. Institute for Software Research International, Carnegie Mellon University (2006)

Phishing Corpus (2006), http://monkey.org/~jose/wiki/doku.php?id=PhishingCorpus

Spam Assassin (2006) http://spamassassin.apache.org/

Chapelle, O., Vapnik, V.: Model Selection for Support Vector Machines. Advances in Neural Information Processing Systems 12

Cherkassy, V.: Model Complexity Control and Statistical Learning Theory. Journal of Natural Computing 1, 109–133 (2002)

Lee, J.H., Lin, C.J.: Automatic Model Selection for Support Vector Machines. Technical report, Department of Computer Science and Information Engineering, National Taiwan University (2000)

Chang, C.C., Lin, C.J.: LIBSVM: A Library for Support Vector Machines. Department of Computer Science and Information Engineering, National Taiwan University (2001)

Chan, C.H., King, I.: Using Biased Support Vector Machine to Improve Retrieval Result in Image Retrieval with Self-organizing Map. In: Proceedings of International Conference on Neural Information Processing, pp. 714–719. Springer, Heidelberg (2004)

Moller, A.F.: A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Neural Networks 6, 525–533 (1993)

Vesanto, J., et al.: Self Organizing Map (SOM) Toolbox. In: Proceedings of Mat lab DSP Conference, Finland, pp. 35–40 (1999)

Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Anderberg, M.: Cluster Analysis for Applications. Academic Press, London (1973)

Egan, J.P.: Signal Detection Theory and ROC Analysis. Academic Press, New York (1975)