Adversarial Support Vector Machine Learning

Yan Zhou Murat Kantarcioglu Bhavani Thuraisingham
Computer Science Department
University of Texas at Dallas
Richardson, TX 75080
yan.zhou2@utdallas.edu, muratk@utdallas.edu, bxt043000@utdallas.edu

Bowei Xi
Department of Statistics
Purdue University
West Lafayette, IN 47907
xbw@stat.purdue.edu

ABSTRACT

Many learning tasks such as spam filtering and credit card fraud detection face an active adversary that tries to avoid detection. For learning problems that deal with an active adversary, it is important to model the adversary's attack strategy and develop robust learning models to mitigate the attack. These are the two objectives of this paper. We consider two attack models: a *free-range* attack model that permits arbitrary data corruption and a *restrained* attack model that anticipates more realistic attacks that a reasonable adversary would devise under penalties. We then develop optimal SVM learning strategies against the two attack models. The learning algorithms minimize the hinge loss while assuming the adversary is modifying data to maximize the loss. Experiments are performed on both artificial and real data sets. We demonstrate that optimal solutions may be overly pessimistic when the actual attacks are much weaker than expected. More important, we demonstrate that it is possible to develop a much more resilient SVM learning model while making loose assumptions on the data corruption models. When derived under the *restrained* attack model, our optimal SVM learning strategy provides more robust overall performance under a wide range of attack parameters.

Categories and Subject Descriptors

I.5.1 [**Computing Methodologies**]: Pattern Recognition—*Models*; I.2.6 [**Computing Methodologies**]: Artificial Intelligence —*Learning*

General Terms

Theory, Algorithms

KDD'12, August 12–16, 2012, Beijing, China.

Keywords

adversarial learning, attack models, robust SVM

1. INTRODUCTION

Many learning tasks, such as intrusion detection and spam filtering, face adversarial attacks. Adversarial exploits create additional challenges to existing learning paradigms. Generalization of a learning model over future data cannot be achieved under the assumption that current and future data share identical properties, which is essential to the traditional approaches. In the presence of active adversaries, data used for training in a learning system is unlikely to represent future data the system would observe. The difference is not just simple random noise which most learning algorithms have already taken into consideration when they are designed. What typically flunk these learning algorithms are targeted attacks that aim to make the learning system dysfunctional by disguising malicious data that otherwise would be detected. Existing learning algorithms cannot be easily tailored to counter this kind of attack because there is a great deal of uncertainty in terms of how much the attacks would affect the structure of the sample space. Despite the sample size and distribution of malicious data given at training time, we would need to make an educated guess about how much the malicious data would change, as sophisticated attackers adapt quickly to evade detection. Attack models, that foretell how far an adversary would go in order to breach the system, need to be incorporated into learning algorithms to build a robust decision surface. In this paper, we present two attack models that cover a *wide range of attacks* tailored to match the adversary's motives. Each attack model makes a simple and realistic assumption on what is known to the adversary. Optimal SVM learning strategies are then derived against the attack models.

Some earlier work lays important theoretical foundations for problems in adversarial learning [15, 6, 20]. However, earlier work often makes strong assumptions such as unlimited computing resource and both sides having a complete knowledge of their opponents. Some proposes attack models that may not permit changes made to arbitrary sets of features [20]. In security applications, some existing research mainly explores practical means of defeating learning algorithms used in a given application domain [25, 19, 22].

Meanwhile, various learning strategies are proposed to fix application-specific weaknesses in learning algorithms [24, 21, 17], but only to find new doors open for future attacks [10, 22]. The main challenge remains as attackers continually exploit unknown weaknesses of a learning system. Regardless of how well designed a learning system appears to be, there are always "blind" spots it fails to detect, leading to escalating threats as the technical strengths on both sides develop. Threats are often divided into two groups, with one group aiming to smuggle malicious content past learning based detection mechanism, while the other trying to undermine the credibility of a learning system by raising both false positive and false negative rates [3]. The grey area in between is scarcely researched. In this work, we set ourselves free from handling application-specific attacks and addressing specific weaknesses of a learning algorithm. Our main contributions lie in the following three aspects:

- We develop a learning strategy that solves a general convex optimization problem where the strength of the constraints is tied to the strength of attacks.
- We derive optimal support vector machine learning models against an adversary whose attack strategy is defined under a general and reasonable assumption.
- We investigate how the performance of the resulting optimal solutions change with different parameter values in two different attack models. The empirical results suggest our proposed adversarial SVM learning algorithms are quite robust against various degrees of attacks.

The rest of the paper is organized as follows. Section 2 presents the related work in the area of adversarial learning. Section 3 formally defines the problem. Section 4 presents the attack models and Section 5 derives the adversarial SVM models. Section 6 presents experimental results on both artificial and real data sets. Section 7 concludes our work and presents future directions.

2. RELATED WORK

Kearns and Li [15] provide theoretical upper bounds on tolerable malicious error rates for learning in the presence of malicious errors. They assume the adversary has unbounded computational resource. In addition, they assume the adversary has the knowledge of the target concept, target distributions, and internal states of the learning algorithm. They demonstrate that error tolerance needs not come at the expense of efficiency or simplicity, and there are strong ties between learning with malicious errors and standard optimization problems.

Dalvi et al. [6] propose a game theoretic framework for learning problems where there is an optimal opponent. They define the problem as a game between two cost-sensitive opponents: a naive Bayes classifier and an adversary playing optimal strategies. They assume all parameters of both players are known to each other and the adversary knows the exact form of the classifier. Their adversary-aware algorithm makes predictions according to the class that maximizes the conditional utility. Finding optimal solutions remains to be computational intensive, which is typical in game theory.

Lowed and Meek [20] point out that assuming the adversary has perfect knowledge of the classifier is unrealistic. Instead they suggest the adversary can confirm the membership of an arbitrary instance by sending queries to the classifier. They also assume the adversary has available an adversarial cost function over the sample space that maps samples to cost values. This assumption essentially means the adversary needs to know the entire feature space to issue optimal attacks. They propose an adversarial classifier reverse engineering (ACRE) algorithm to learn vulnerabilities of given learning algorithms.

Adversarial learning problems are often modeled as games played between two opponents. Brückner and Scheffer model adversarial prediction problems as Stackelberg games [5]. To guarantee optimality, the model assumes adversaries behave rationally. However, it does not require a unique equilibrium. Kantarcioglu et al. [14] treat the problem as a sequential Stackelberg game. They assume the two players know each other's payoff function. They use simulated annealing and genetic algorithm to search for a Nash equilibrium. Later on such an equilibrium is used to choose optimal set of attributes that give good equilibrium performance. Improved models in which Nash strategies are played have also been proposed [4, 18].

Other game theoretic models play zero-sum minimax strategies. Globerson and Roweis [11] consider a problem where some features may be missing at testing time. This is related to adversarial learning in that the adversary may simply delete highly weighted features in malicious data to increase its chance to evade detection. They develop a game theoretic framework in which classifiers are constructed to be optimal in the worst case scenario. Their idea is to prevent assigning too much weight on any single feature. They use the support vector machine model which optimally minimizes the hinge loss when at most K features can be deleted. El Ghaoui et al [9] apply a minimax model to training data bounded by hyper-rectangles. Their model minimizes the worst-case loss over data in given intervals. Other robust learning algorithms for handling classification-time noise are also proposed [16, 23, 7, 8].

Our work differs from the existing ones in several respects. First of all, we do not make strong assumptions on what is known to either side of the players. Second, both wide-range attacks and targeted attacks are considered and incorporated into the SVM learning framework. Finally, the robustness of the minimax solutions against attacks over a wide range of parameters is investigated.

3. PROBLEM DEFINITION

Denote a sample set by $\{(x_i, y_i) \in (\mathcal{X}, \mathcal{Y})\}_{i=1}^{n}$, where x_i is the i^{th} sample and $y_i \in \{-1, 1\}$ is its label, $\mathcal{X} \subseteq \mathbb{R}^d$ is a d-dimensional feature space, n is the total number of samples. We consider an adversarial learning problem where the adversary modifies malicious data to avoid detection and hence achieves his planned goals. The adversary has the freedom to move only the malicious data ($y_i = 1$) in any direction by adding a non-zero displacement vector δ_i to $x_i|_{y_i=1}$. For example, in spam-filtering the adversary may add good words to spam e-mail to defeat spam filters. On the other hand, adversary will not be able to modify legitimate e-mail.

We make no specific assumptions on the adversary's knowledge of the learning system. Instead, we simply assume there is a trade-off or cost of changing malicious data. For example, a practical strategy often employed by an adversary is to move the malicious data in the feature space as close as possible to where the innocuous data is frequently observed. However, the adversary can only alter a malicious data point

so much that its malicious utility is not completely lost. If the adversary moves a data point too far away from its own class in the feature space, the adversary may have to sacrifice much of the malicious utility of the original data point. For example, in the problem of credit card fraud detection, an attacker may choose the "right" amount to spent with a stolen credit card to mimic a legitimate purchase. By doing so, the attacker will lose some potential profit.

4. ADVERSARIAL ATTACK MODELS

We present two attack models—*free-range* and *restrained*, each of which makes a simple and realistic assumption about how much is known to the adversary. The models differ in their implications for 1) the adversary's knowledge of the innocuous data, and 2) the loss of utility as a result of changing the malicious data. The *free-range* attack model assumes the adversary has the freedom to move data anywhere in the feature space. The *restrained* attack model is a more conservative attack model. The model is built under the intuition that the adversary would be reluctant to let a data point move far away from its original position in the feature space. The reason is that greater displacement often entails loss of malicious utility.

4.1 Free-Range Attack

The only knowledge the adversary needs is the valid range of each feature. Let $x_{.j}^{max}$ and $x_{.j}^{min}$ be the largest and the smallest values that the j^{th} feature of a data point x_i—x_{ij}—can take. For all practical purposes, we assume both $x_{.j}^{max}$ and $x_{.j}^{min}$ are bounded. For example, for a Gaussian distribution, they can be set to the 0.01 and 0.99 quantiles. The resulting range would cover most of the data points and discard a few extreme values. An attack is then bounded in the following form:

$$C_f(x_{.j}^{min} - x_{ij}) \leq \delta_{ij} \leq C_f(x_{.j}^{max} - x_{ij}), \forall j \in [1, d],$$

where $C_f \in [0, 1]$ controls the aggressiveness of attacks. $C_f = 0$ means no attacks, while $C_f = 1$ corresponds to the most aggressive attacks involving the widest range of permitted data movement.

The great advantage of this attack model is that it is sufficiently general to cover all possible attack scenarios as far as data modification is concerned. When paired with a learning model, the combination would produce good performance against the most severe attacks. However, when there are mild attacks, the learning model becomes too "paranoid" and its performance suffers accordingly. Next, we present a more realistic model for attacks where significant data alteration is penalized.

4.2 Restrained Attack

Let x_i be a malicious data point the adversary aims to alter. Let x_i^t, a d-dimensional vector, be a potential target to which the adversary would like to push x_i. The adversary chooses x_i^t according to his estimate of the innocuous data distribution. Ideally, the adversary would optimize x_i^t for each x_i to minimize the cost of changing it and maximize the goal it can achieve. Optimally choosing x_i^t is desired, but often requires a great deal of knowledge about the feature space and sometimes the inner working of a learning algorithm [6, 20]. More realistically, the adversary can set x_i^t to be the estimated centroid of innocuous data, a data point sampled from the observed innocuous data, or an artificial data point generated from the estimated innocuous data distribution. Note that x_i^t could be a rough guess if the adversary has a very limited knowledge of the innocuous data, or a very accurate one if the adversary knows the exact make up of the training data.

In most cases, the adversary cannot change x_i to x_i^t as desired since x_i may lose too much of its malicious utility. Therefore, for each attribute j in the d-dimensional feature space, we assume the adversary adds δ_{ij} to x_{ij} where

$$|\delta_{ij}| \leq |x_{ij}^t - x_{ij}|, \ \forall \ j \in d.$$

Furthermore, we place an upper bound on the amount of displacement for attribute j as follows:

$$0 \leq (x_{ij}^t - x_{ij})\delta_{ij} \leq \left(1 - C_\delta \frac{|x_{ij}^t - x_{ij}|}{|x_{ij}| + |x_{ij}^t|}\right)(x_{ij}^t - x_{ij})^2,$$

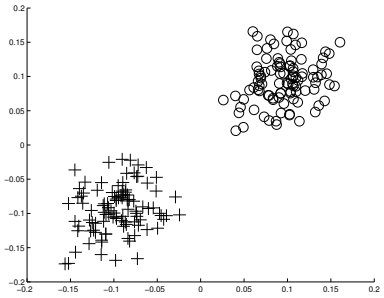
where $C_\delta \in [0, 1]$ is a constant modeling the loss of malicious utility as a result of the movement δ_{ij}. This attack model specifies how much the adversary can push x_{ij} towards x_{ij}^t based on how far apart they are from each other. The term $1 - C_\delta \frac{|x_{ij}^t - x_{ij}|}{|x_{ij}| + |x_{ij}^t|}$ is the percentage of $x_{ij}^t - x_{ij}$ that δ_{ij} is allowed to be at most. When C_δ is fixed, the closer x_{ij} is to x_{ij}^t, the more x_{ij} is allowed to move towards x_{ij}^t percentage wise. The opposite is also true. The farther apart x_{ij} and x_{ij}^t, the smaller $|\delta_{ij}|$ will be. For example, when x_{ij} and x_{ij}^t reside on different sides of the origin, that is, one is positive and the other is negative, then no movement is permitted (that is, $\delta_{ij} = 0$) when $C_\delta = 1$. This model balances between the needs of disguising maliciousness of data and retaining its malicious utility in the mean time. $(x_{ij}^t - x_{ij})\delta_{ij} \geq 0$ ensures δ_{ij} moves in the same direction as $x_{ij}^t - x_{ij}$. C_δ is related to the loss of malicious utility after the data has been modified. C_δ sets how much malicious utility the adversary is willing to sacrifice for breaking through the decision boundary. A larger C_δ means smaller loss of malicious utility, while a smaller C_δ models greater loss of malicious utility. Hence a larger C_δ leads to less aggressive attacks while a smaller C_δ leads to more aggress attacks.

The attack model works great for well-separated data as shown in Figure 1(a). When data from both classes are near the separation boundary as shown in Figure 1(b), slightly changing attribute values would be sufficient to push the data across the boundary. In this case, even if C_δ is set to 1, the attack from the above model would still be too aggressive compared with what is needed. We could allow $C_\delta > 1$ to further reduce the aggressiveness of attacks, however, for simplicity and more straightforward control, we instead apply a discount factor C_ξ to $|x_{ij}^t - x_{ij}|$ directly to model the severeness of attacks:

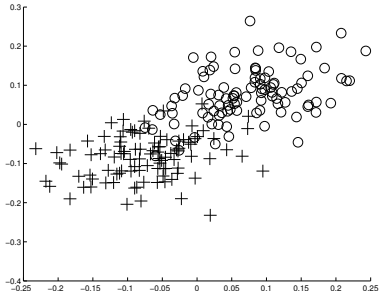
$$0 \leq (x_{ij}^t - x_{ij})\delta_{ij} \leq C_\xi \left(1 - \frac{|x_{ij}^t - x_{ij}|}{|x_{ij}| + |x_{ij}^t|}\right)(x_{ij}^t - x_{ij})^2,$$

where $C_\xi \in [0, 1]$. A large C_ξ gives rise to a greater amount of data movement, and a small C_ξ sets a narrower limit on data movement. Combining these two cases, the *restrained-attack* model is given as follows:

$$0 \leq (x_{ij}^t - x_{ij})\delta_{ij} \leq C_\xi \left(1 - C_\delta \frac{|x_{ij}^t - x_{ij}|}{|x_{ij}| + |x_{ij}^t|}\right)(x_{ij}^t - x_{ij})^2.$$



(a) Data well separated



(b) Data cluttered near boundary

Figure 1: Data well separated and data cluttered near separating boundary.

5. ADVERSARIAL SVM LEARNING

We now present an adversarial support vector machine model (AD-SVM) against each of the two attack models discussed in the previous section. We assume the adversary cannot modify the innocuous data. Note that this assumption can be relaxed to model cases where the innocuous data may also be altered.

5.1 AD-SVM against Free-range Attack Model

We first consider the free-range attack model. The hinge loss model is given as follows:

$$\begin{array}{lcll} h(w,b,x_i) & = & \begin{cases} \max\limits_{\delta_i} \lfloor 1-(w\cdot(x_i+\delta_i)+b)\rfloor_+ & \text{if } y_i=1 \\ \lfloor 1+(w\cdot x_i+b)\rfloor_+ & \text{if } y_i=-1 \end{cases} \\ s.t. & & \\ & & \delta_i \preceq C_f(x^{max}-x_i) \\ & & \delta_i \succeq C_f(x^{min}-x_i) \end{array}$$

where δ_i is the displacement vector for x_i, $\preceq$ and $\succeq$ denote component-wise inequality.

Following the standard SVM risk formulation, we have

$$\begin{array}{ll} \underset{w,b}{\text{argmin}} & \sum_{\{i|y_i=1\}} \max\limits_{\delta_i} \lfloor 1-(w\cdot(x_i+\delta_i)+b)\rfloor_+ \\ & +\sum_{\{i|y_i=-1\}} \lfloor 1+(w\cdot x_i+b)\rfloor_+ \\ & +\mu||w||^2 \end{array}$$

Combining cases for positive and negative instances, this is equivalent to:

$$\begin{array}{ll} \underset{w,b}{\text{argmin}} & \sum\limits_i \max\limits_{\delta_i} \lfloor 1-y_i(w\cdot x_i+b)-\frac{1}{2}(1+y_i)w\cdot\delta_i\rfloor_+ \\ & +\mu||w||^2 \end{array}$$

Note that the worst case hinge loss of x_i is obtained when δ_i is chosen to minimize its contribution to the margin, that is,

$$\begin{array}{lcl} f_i & = & \min\limits_{\delta_i} \frac{1}{2}(1+y_i)w\cdot\delta_i \\ s.t. & & \delta_i \preceq C_f(x^{max}-x_i) \\ & & \delta_i \succeq C_f(x^{min}-x_i) \end{array}$$

This is a disjoint bilinear problem with resect to w and δ_i. Here, we are interested in discovering optimal assignment to δ_i with a given w. We can reduce the bilinear problem to the following asymmetric dual problem over $u_i \in \mathbb{R}^d$, $v_i \in \mathbb{R}^d$ where d is the dimension of the feature space:

$$\begin{array}{lcl} g_i & = & \max -\sum_j C_f\left(v_{ij}(x_j^{max}-x_{ij})-u_{ij}(x_j^{min}-x_{ij})\right) \\ \text{or} & & \\ g_i & = & \min \sum_j C_f\left(v_{ij}(x_j^{max}-x_{ij})-u_{ij}(x_j^{min}-x_{ij})\right) \\ s.t. & & (u_i-v_i)=\frac{1}{2}(1+y_i)w \\ & & u_i \succeq 0 \\ & & v_i \succeq 0 \end{array}$$

The SVM risk minimization problem can be rewritten as follows:

$$\begin{array}{ll} \underset{w,b,t_i,u_i,v_i}{\text{argmin}} & \frac{1}{2}||w||^2 + C\sum_i \lfloor 1-y_i\cdot(w\cdot x_i+b)+t_i\rfloor_+ \\ s.t. & t_i \geq \sum_j C_f\left(v_{ij}(x_j^{max}-x_{ij})-u_{ij}(x_j^{min}-x_{ij})\right) \\ & u_i-v_i=\frac{1}{2}(1+y_i)w \\ & u_i \succeq 0 \\ & v_i \succeq 0 \end{array}$$

Adding a slack variable and linear constraints to remove the non-differentiality of the hinge loss, we can rewrite the problem as follows:

$$\begin{array}{ll} \underset{w,b,\xi_i,t_i,u_i,v_i}{\text{argmin}} & \frac{1}{2}||w||^2 + C\sum_i \xi_i \\ s.t. & \xi_i \geq 0 \\ & \xi_i \geq 1-y_i\cdot(w\cdot x_i+b)+t_i \\ & t_i \geq \sum_j C_f\left(v_{ij}(x_j^{max}-x_{ij})-u_{ij}(x_j^{min}-x_{ij})\right) \\ & u_i-v_i=\frac{1}{2}(1+y_i)w \\ & u_i \succeq 0 \\ & v_i \succeq 0 \end{array}$$

5.2 AD-SVM against Restrained Attack Model

With the restrained attack model, we modify the hinge loss model and solve the problem following the same steps:

$$\begin{array}{l} h(w,b,x_i) = \begin{cases} \max\limits_{\delta_i} \lfloor 1-(w\cdot(x_i+\delta_i)+b)\rfloor_+ & \text{if } y_i=1 \\ \lfloor 1+(w\cdot x_i+b)\rfloor_+ & \text{if } y_i=-1 \end{cases} \\ s.t. \\ (x_i^t-x_i)\circ\delta_i \preceq C_\xi\left(1-C_\delta\frac{|x_i^t-x_i|}{|x_i|+|x_i^t|}\right)\circ(x_i^t-x_i)^{\circ 2} \\ (x_i^t-x_i)\circ\delta_i \succeq 0 \end{array}$$

where δ_i denotes the modification to x_i, $\preceq$ is component-wise inequality, and $\circ$ denotes component-wise operations.

The worst case hinge loss is obtained by solving the following minimization problem:

$$\begin{array}{lcl} f_i & = & \min\limits_{\delta_i} \frac{1}{2}(1+y_i)w\cdot\delta_i \\ s.t. & & (x_i^t-x_i)\circ\delta_i \preceq C_\xi\left(1-C_\delta\frac{|x_i^t-x_i|}{|x_i|+|x_i^t|}\right)\circ(x_i^t-x_i)^{\circ 2} \\ & & (x_i^t-x_i)\circ\delta_i \succeq 0 \end{array}$$

Let

$$e_{ij} = C_\xi\left(1-C_\delta\frac{|x_{ij}^t-x_{ij}|}{|x_{ij}|+|x_{ij}^t|}\right)(x_{ij}^t-x_{ij})^2.$$

We reduce the bilinear problem to the following asymmetric dual problem over $u_i \in \mathbb{R}^d$, $v_i \in \mathbb{R}^d$ where d is the dimension of the feature space:

$$\begin{array}{lll} g_i & = & \max -\sum_j e_{ij}u_{ij}, \text{ or} \\ g_i & = & \min \sum_j e_{ij}u_{ij} \\ s.t. & & (-u_i + v_i) \circ (x_i^t - x_i) = \frac{1}{2}(1 + y_i)w \\ & & u_i \succeq 0 \\ & & v_i \succeq 0 \end{array}$$

The SVM risk minimization problem can be rewritten as follows:

$$\begin{array}{ll} \underset{w,b,t_i,u_i,v_i}{\text{argmin}} & \frac{1}{2}||w||^2 + C\sum_i \lfloor 1 - y_i \cdot (w \cdot x_i + b) + t_i \rfloor_+ \\ s.t. & t_i \geq \sum_j e_{ij}u_{ij} \\ & (-u_i + v_i) \circ (x_i^t - x_i) = \frac{1}{2}(1 + y_i)w \\ & u_i \succeq 0 \\ & v_i \succeq 0 \end{array}$$

After removing the non-differentiality of the hinge loss, we can rewrite the problem as follows:

$$\begin{array}{ll} \underset{w,b,\xi_i,t_i,u_i,v_i}{\text{argmin}} & \frac{1}{2}||w||^2 + C\sum_i \xi_i \\ s.t. & \xi_i \geq 0 \\ & \xi_i \geq 1 - y_i \cdot (w \cdot x_i + b) + t_i \\ & t_i \geq \sum_j e_{ij}u_{ij} \\ & (-u_i + v_i) \circ (x_i^t - x_i) = \frac{1}{2}(1 + y_i)w \\ & u_i \succeq 0 \\ & v_i \succeq 0 \end{array}$$

6. EXPERIMENT

We test the AD-SVM models on both artificial and real data sets. In our experiments, we investigate the robustness of the AD-SVM models as we increase the severeness of the attacks. We let x_i^t be the centroid of the innocuous data in our AD-SVM model against restrained attacks. We also tried setting x_i^t to a random innocuous data point in the training or test set, and the results are similar. Due to space limitations, we do not report the results in the latter cases.

Attacks on the test data used in the experiments are simulated using the following model:

$$\delta_{ij} = f_{attack}(x_{ij}^- - x_{ij})$$

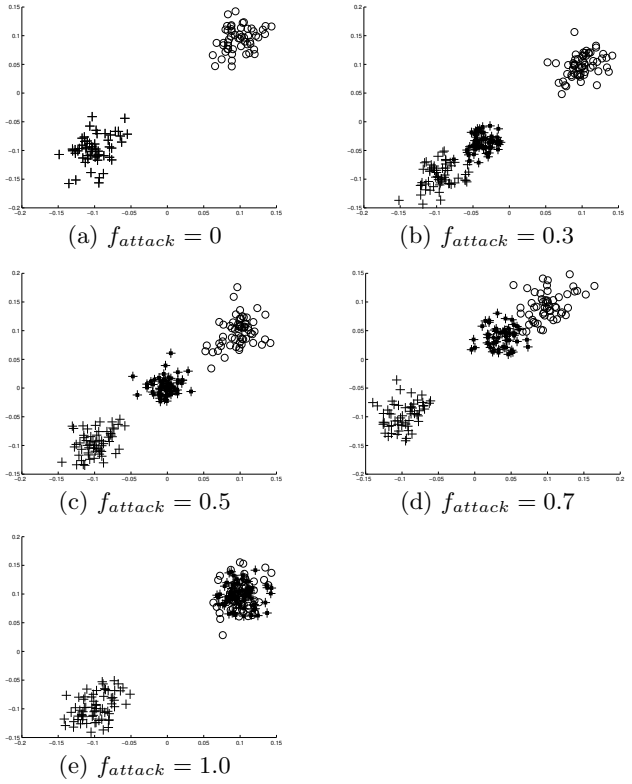
where x_i^- is an innocuous data point randomly chosen from the test set, and $f_{attack} > 0$ sets a limit for the adversary to move the test data toward the target innocuous data points. By controlling the value of f_{attack}, we can dictate the severity of attacks in the simulation. The actual attacks on the test data are intentionally designed not to match the attack models in AD-SVM so that the results are not biased. For each parameter C_f, C_δ and C_ξ in the attack models considered in AD-SVM, we tried different values as f_{attack} increases. This allows us to test the robustness of our AD-SVM model in all cases where there are no attacks and attacks that are much more severe than the model has anticipated. We compare our AD-SVM model to the standard SVM and one-class SVM models. We implemented our AD-SVM algorithms in CVX—a package for specifying and solving convex programs [12]. Experiments using SVM and one-class SVM are implemented using Weka [13].

6.1 Experiments on Artificial Dataset

We generate two artificial data sets from bivariate normal distributions with specified means and covariance matrices. Data in the first data set is well separated. The second data set consists of data more cluttered near the separating boundary. All results are averaged over 100 random runs.

6.1.1 Data Points Well Separated

Figure 2 illustrates the data distributions when different levels of distortion are applied to the malicious data by setting f_{attack} to 0 (original distribution), 0.3, 0.5, 0.7, and 1.0. As can be observed, as f_{attack} increases, the malicious data points are moved more aggressively towards innocuous data.



(a) $f_{attack} = 0$ (b) $f_{attack} = 0.3$ (c) $f_{attack} = 0.5$ (d) $f_{attack} = 0.7$ (e) $f_{attack} = 1.0$

Figure 2: Data distributions of the *first* data set after attacks. f_{attack} varies from 0 (no attack) to 1.0 (most aggressive). Plain "+" marks the original positive data points, "+" with a central black square marks positive data points after alteration, and "○" represents negative data.

Table 1 lists the predictive accuracy of our AD-SVM algorithm with the free-range attack model, the standard SVM algorithm, and the one-class SVM algorithm. AD-SVM clearly outperforms both SVM and one-class SVM when it assumes reasonable adversity ($C_f \in [0.1, 0.5]$). When there is mild attack or no attack at all, AD-SVM with more aggressive free-range assumptions ($C_f \in [0.5, 0.9]$) suffers great performance loss as we expect from such pessimistic model.

Compared to the free-range attack model, the restrained attack model works much more consistently across the entire spectrum of the learning and attack parameters. Here C_δ reflects the aggressiveness of attacks in our AD-SVM learning algorithm. Table 2 shows the classification results as C_δ decreases, from less aggressive ($C_\delta = 0.9$) to very aggressive ($C_\delta = 0.1$). Clearly, the most impressive results are lined up along the diagonal when the assumptions on the attacks

Table 1: Accuracy of *free-range* AD-SVM, SVM, and one-class SVM under data distributions shown in Figure 2(a), 2(b), 2(c), 2(d), and 2(e). C_f *increases* as the learning model assumes more aggressive attacks.

		$f_{attack}=0$	$f_{attack}=0.3$	$f_{attack}=0.5$	$f_{attack}=0.7$	$f_{attack}=1.0$
AD-SVM	$C_f=0.1$	1.000	**1.000**	**0.887**	**0.512**	0.500
	$C_f=0.3$	1.000	**1.000**	**0.997**	**0.641**	0.500
	$C_f=0.5$	0.996	0.996	**0.996**	**0.930**	0.500
	$C_f=0.7$	0.882	0.886	**0.890**	**0.891**	0.500
	$C_f=0.9$	0.500	0.500	0.500	0.500	0.500
SVM		1.000	0.999	0.751	0.502	0.500
One-class SVM		1.000	0.873	0.500	0.500	0.500

Table 2: Accuracy of *restrained* AD-SVM, SVM, and one-class SVM under data distributions shown in Figure 2(a), 2(b), 2(c), 2(d), and 2(e). C_δ *decreases* as the learning model assumes more aggressive attacks.

		$f_{attack}=0$	$f_{attack}=0.3$	$f_{attack}=0.5$	$f_{attack}=0.7$	$f_{attack}=1.0$
AD-SVM ($C_\xi=1$)	$C_\delta=0.9$	1.000	**1.000**	**0.856**	**0.505**	0.500
	$C_\delta=0.7$	1.000	**1.000**	**0.975**	**0.567**	0.500
	$C_\delta=0.5$	1.000	**1.000**	**0.999**	**0.758**	0.500
	$C_\delta=0.3$	0.994	0.994	**0.994**	**0.954**	0.500
	$C_\delta=0.1$	0.878	0.876	**0.878**	**0.878**	0.500
SVM		1.000	0.998	0.748	0.501	0.500
One-class SVM		1.000	0.873	0.500	0.500	0.500

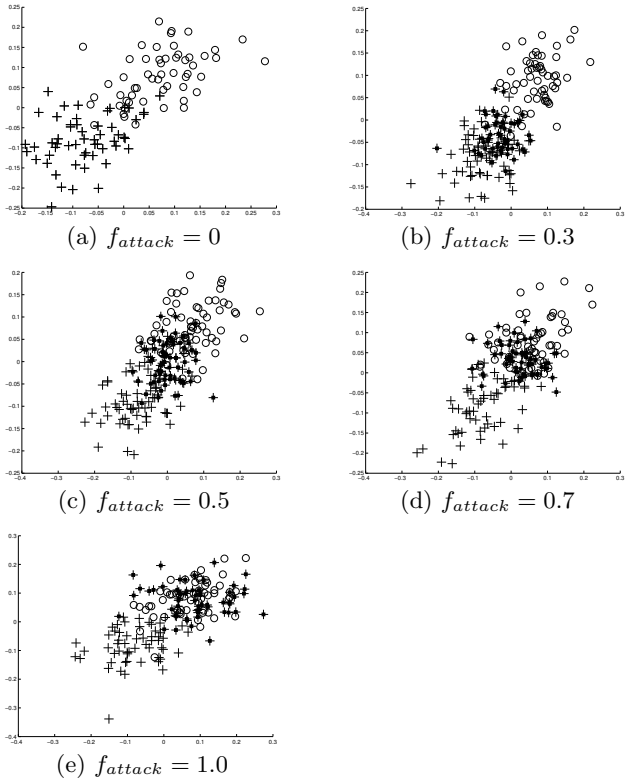
made in the learning model match the real attacks. The results of our AD-SVM in the rest of the experiments are mostly superior to both SVM and one-class SVM too. This relax the requirement of finding the best C_δ. Regardless of what C_δ value is chosen, our model delivers solid performance.

6.1.2 Data Cluttered Near Separating Boundary

Figure 3 illustrates the distributions of our second artificial data set under different levels of attacks. Malicious data points can be pushed across the boundary with little modification. We again consider both the free-range and the restrained attack models. Similar conclusions can be drawn: restrained AD-SVM is more robust than free-range AD-SVM; AD-SVMs in general cope much better with mild adversarial attacks than standard SVM and one-class SVM models.

Table 3 lists the predictive accuracy of our AD-SVM algorithm with the free-range attack model on the second data set. The results of the standard SVM algorithm and the one-class SVM algorithm are also listed. The free-range model is overly pessimistic in many cases, which overshadows its resilience against the most severe attacks. For the restrained attack model, since the two classes are not well separated originally, C_ξ is used (not combined with C_δ) to reflect the aggressiveness of attacks in AD-SVM. A larger C_ξ is more aggressive while a smaller C_ξ assumes mild attacks. Table 4 shows the classification results as C_ξ increases, from less aggressive ($C_\xi=0.1$) to very aggressive ($C_\xi=0.9$).

The restrained AD-SVM model still manages to improve the predictive accuracy compared to SVM and one-class SVM, although the improvement is much less impressive. This is understandable since the data set is generated to make it harder to differentiate between malicious and innocuous data, with or without attacks. The model suffers no performance loss when there are no attacks.



(a) $f_{attack}=0$ (b) $f_{attack}=0.3$ (c) $f_{attack}=0.5$ (d) $f_{attack}=0.7$ (e) $f_{attack}=1.0$

Figure 3: Data distributions of the *second* data set after attacks. f_{attack} varies from 0 (none) to 1.0 (most aggressive). Plain "+" marks the original positive data points, "+" with a central black square is for positive data points after alteration, and "○" represents negative data.

Table 3: Accuracy of *free-range* AD-SVM, SVM, and one-class SVM under data distributions shown in Figure 3(a), 3(b), 3(c), 3(d), and 3(e). C_f *increases* as the learning model assumes more aggressive attacks.

		$f_{attack}=0$	$f_{attack}=0.3$	$f_{attack}=0.5$	$f_{attack}=0.7$	$f_{attack}=1.0$
AD-SVM	$C_f=0.1$	0.928	**0.884**	**0.771**	**0.609**	0.500
	$C_f=0.3$	0.859	0.848	**0.807**	**0.687**	0.500
	$C_f=0.5$	0.654	0.649	0.658	**0.638**	0.500
	$C_f=0.7$	0.500	0.500	0.500	0.500	0.500
	$C_f=0.9$	0.500	0.500	0.500	0.500	0.500
SVM		0.932	0.859	0.715	0.575	0.500
One-class SVM		0.936	0.758	0.611	0.527	0.500

Table 4: Accuracy of *restrained* AD-SVM, SVM, and one-class SVM under data distributions shown in Figure 3(a), 3(b), 3(c), 3(d), and 3(e). C_ξ *increases* as the learning model assumes more aggressive attacks.

		$f_{attack}=0$	$f_{attack}=0.3$	$f_{attack}=0.5$	$f_{attack}=0.7$	$f_{attack}=1.0$
AD-SVM ($C_\delta=1$)	$C_\xi=0.9$	0.932	**0.860**	**0.719**	**0.575**	0.500
	$C_\xi=0.7$	0.930	**0.858**	**0.717**	**0.576**	0.500
	$C_\xi=0.5$	**0.935**	**0.860**	**0.721**	**0.578**	0.500
	$C_\xi=0.3$	0.931	0.855	**0.718**	**0.577**	0.500
	$C_\xi=0.1$	0.933	**0.858**	**0.718**	**0.575**	0.500
SVM		0.930	0.856	0.714	0.574	0.500
One-class SVM		0.933	0.772	0.605	0.525	0.500

6.2 Experiments on Real Datasets

We also test our AD-SVM model on two real datasets: *spam base* taken from the UCI data repository [2], and *web spam* taken from the LibSVM website [1].

In the *spam base* data set, the spam concept includes advertisements, make money fast scams, chain letters, etc. The spam collection came from the postmaster and individuals who had filed spam. The non-spam e-mail collection came from filed work and personal e-mails [2]. The dataset consists of 4601 total number of instances, among which 39.4% is spam. There are 57 attributes and one class label. We divide the data sets into equal halves, with one half T_r for training and the other half T_s for test only. Learning models are built from 10% of random samples selected from T_r. The results are averaged over 10 random runs.

We took the second data set from the LibSVM website [1]. According to the website, the *web spam* data is the subset used in the Pascal Large Scale Learning Challenge. All positive examples were kept in the data set while the negative examples were created by randomly traversing the Internet starting at well known web-sites. They treat continuous n bytes as a word and use word count as the feature value and normalize each instance to unit length. We use their unigram data set in which the number of features is 254. The total number of instances is 350,000. We again divide the data set into equal halves for training and test. We use 2% of the samples in the training set to build the learning models and report the results averaged over 10 random runs.

Table 5 and Table 6 show the results on the *spam base* data set. AD-SVM, with both the free-range and the restrained attack models, achieved solid improvement on this data set. C_δ alone is used in the restrained learning model. Except for the most pessimistic cases, AD-SVM suffers no performance loss when there are no attacks. On the other hand, it achieved much more superior classification accuracy than SVM and one-class SVM when there are attacks.

Table 7 and Table 8 illustrate the results on the *web spam* data set. Unlike the *spam base* data set where data is well separated, *web spam* data is more like the second artificial data set. The AD-SVM model exhibits similar classification performance as on the second artificial data set. The free-range model is too pessimistic when there are no attacks, while the restrained model performs consistently better than SVM and one-class SVM and, more importantly, suffers no loss when there are no attacks. We use C_ξ alone in our learning model. Which parameter, C_ξ or C_δ, to use in the restrained attack model can be determined through cross validation on the initial data. Next subsection has a more detailed discussion on model parameters.

6.3 Setting C_f, C_ξ, and C_δ

The remaining question is how to set the parameters in the attack models. The AD-SVM algorithms proposed in this paper assume either a free-range attack model or a restrained attack model. In reality we might not know the exact attack model or the true utility function of the attackers. However, as Tables 1–8 demonstrate, although the actual attacks may not match what we have anticipated, our AD-SVM algorithm using the restrained attack model exhibits overall robust performance by setting C_δ or C_ξ values for more aggressive attacks. If we use the restrained attack model, choosing $C_\delta \leq 0.5$ ($C_\xi \geq 0.5$) consistently returns robust results against all f_{attack} values. If we use the free-range attack model in AD-SVM, we will have to set parameter values to avoid the very pessimistic results for mild attacks. Hence choosing $C_f \leq 0.3$ in general returns good classification results against all f_{attack} values.

As a general guideline, the baseline of C_f, C_δ or C_ξ has to be chosen to work well against attack parameters suggested by domain experts. This can be done through cross-validation for various attack scenarios. From there, we gradually increase C_f or C_ξ, or decrease in the case of C_δ. The best value of C_f, C_δ or C_ξ is reached right before perfor-

Table 5: Accuracy of AD-SVM, SVM, and one-class SVM on the *spambase* dataset as attacks intensify. The *free-range* attack is used in the learning model. C_f *increases* as attacks become more aggressive.

		$f_{attack}=0$	$f_{attack}=0.3$	$f_{attack}=0.5$	$f_{attack}=0.7$	$f_{attack}=1.0$
AD-SVM	$C_f=0.1$	**0.882**	**0.852**	**0.817**	**0.757**	**0.593**
	$C_f=0.3$	0.880	**0.864**	**0.833**	**0.772**	**0.588**
	$C_f=0.5$	0.870	**0.860**	**0.836**	**0.804**	**0.591**
	$C_f=0.7$	0.859	**0.847**	**0.841**	**0.814**	**0.592**
	$C_f=0.9$	0.824	**0.829**	**0.815**	**0.802**	**0.598**
SVM		0.881	0.809	0.742	0.680	0.586
One-Class SVM		0.695	0.686	0.667	0.653	0.572

Table 6: Accuracy of AD-SVM and SVM on *spambase* dataset as attacks intensify. The *restrained* attack model is used in the learning model. C_δ *decreases* as attacks become more aggressive.

		$f_{attack}=0$	$f_{attack}=0.3$	$f_{attack}=0.5$	$f_{attack}=0.7$	$f_{attack}=1.0$
AD-SVM ($C_\xi=1$)	$C_\delta=0.9$	0.874	**0.821**	**0.766**	**0.720**	0.579
	$C_\delta=0.7$	**0.888**	**0.860**	**0.821**	**0.776**	0.581
	$C_\delta=0.5$	0.874	**0.860**	**0.849**	**0.804**	0.586
	$C_\delta=0.3$	0.867	**0.855**	**0.845**	**0.809**	0.590
	$C_\delta=0.1$	0.836	**0.840**	**0.839**	**0.815**	**0.597**
SVM		0.884	0.812	0.761	0.686	0.591
One-class SVM		0.695	0.687	0.676	0.653	0.574

Table 7: Accuracy of AD-SVM, SVM, and one-class SVM on *webspam* dataset as attacks intensify. The *free-range* attack model is used in the learning model. C_f *increases* as attacks become more aggressive.

		$f_{attack}=0$	$f_{attack}=0.3$	$f_{attack}=0.5$	$f_{attack}=0.7$	$f_{attack}=1.0$
AD-SVM	$C_f=0.1$	0.814	**0.790**	**0.727**	**0.591**	**0.463**
	$C_f=0.3$	0.760	0.746	**0.732**	**0.643**	**0.436**
	$C_f=0.5$	0.684	0.649	0.617	**0.658**	**0.572**
	$C_f=0.7$	0.606	0.606	0.606	**0.606**	**0.606**
	$C_f=0.9$	0.606	0.606	0.606	**0.606**	**0.606**
SVM		0.874	0.769	0.644	0.534	0.427
One-class SVM		0.685	0.438	0.405	0.399	0.399

Table 8: Accuracy of AD-SVM, SVM, and one-class SVM on *webspam* dataset as attacks intensify. The *restrained* attack model is used in the learning model. C_ξ *increases* as attacks become more aggressive.

		$f_{attack}=0$	$f_{attack}=0.3$	$f_{attack}=0.5$	$f_{attack}=0.7$	$f_{attack}=1.0$
AD-SVM ($C_\delta=1$)	$C_\xi=0.1$	**0.873**	**0.822**	**0.699**	**0.552**	**0.435**
	$C_\xi=0.3$	0.870	**0.837**	**0.748**	**0.597**	**0.444**
	$C_\xi=0.5$	0.855	**0.833**	**0.772**	**0.641**	**0.454**
	$C_\xi=0.7$	0.841	**0.820**	**0.773**	**0.663**	**0.467**
	$C_\xi=0.9$	0.822	**0.803**	**0.749**	**0.671**	**0.478**
SVM		0.871	0.769	0.659	0.512	0.428
One-class SVM		0.684	0.436	0.406	0.399	0.400

mance deteriorates. Also note that it is sufficient to set only one of C_ξ and C_δ while fixing the other to 1. Furthermore, C_f, C_δ and C_ξ do not have to be a scalar parameter. In many applications, it is clear some attributes can be changed while others cannot. A C_f, C_δ/C_ξ parameter vector would help enforce these additional rules.

7. CONCLUSIONS AND FUTURE WORK

Adversarial attacks can lead to severe misrepresentation of real data distributions in the feature space. Learning algorithms lacking the flexibility of handling the structural change in the samples would not cope well with attacks that modify data to change the make up of the sample space. We present two attack models and an adversarial SVM learning model against each attack model. We demonstrate that our adversarial SVM model is much more resilient to adversarial attacks than standard SVM and one-class SVM models. We also show that optimal learning strategies derived to counter overly pessimistic attack models can produce unsatisfactory results when the real attacks are much weaker. On the other hand, learning models built on restrained attack models perform more consistently as attack parameters vary. One fu-

ture direction for this work is to add cost-sensitive metrics into the learning models. Another direction is to extend the single learning model to an ensemble in which each base learner handles a different set of attacks.

8. ACKNOWLEDGEMENT

This work was partially supported by Air Force Office of Scientific Research MURI Grant FA9550-08-1-0265, National Institutes of Health Grant 1R01LM009989, National Science Foundation (NSF) Grant Career-CNS-0845803, and NSF Grants CNS-0964350, CNS-1016343, CNS-1111529.

9. REFERENCES

[1] *LIBSVM Data: Classification, Regression, and Multi-label*, 2012.

[2] *UCI Machine Learning Repository*, 2012.

[3] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 16–25, New York, NY, USA, 2006. ACM.

[4] M. Bruckner and T. Scheffer. Nash equilibria of static prediction games. In *Advances in Neural Information Processing Systems*. MIT Press, 2009.

[5] M. Bruckner and T. Scheffer. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2011.

[6] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 99–108, New York, NY, USA, 2004. ACM.

[7] O. Dekel and O. Shamir. Learning to classify with missing and corrupted features. In *Proceedings of the International Conference on Machine Learning*, pages 216–223. ACM, 2008.

[8] O. Dekel, O. Shamir, and L. Xiao. Learning to classify with missing and corrupted features. *Machine Learning*, 81(2):149–178, 2010.

[9] L. El Ghaoui, G. R. G. Lanckriet, and G. Natsoulis. Robust classification with interval data. Technical Report UCB/CSD-03-1279, EECS Department, University of California, Berkeley, Oct 2003.

[10] P. Fogla and W. Lee. Evading network anomaly detection systems: formal reasoning and practical techniques. In *Proceedings of the 13th ACM conference on Computer and communications security*, CCS '06, pages 59–68, New York, NY, USA, 2006. ACM.

[11] A. Globerson and S. Roweis. Nightmare at test time: robust learning by feature deletion. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 353–360. ACM, 2006.

[12] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. `http://cvxr.com/cvx/`, Apr. 2011.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11:10–18, November 2009.

[14] M. Kantarcioglu, B. Xi, and C. Clifton. Classifier evaluation and attribute selection against active adversaries. *Data Min. Knowl. Discov.*, 22:291–335, January 2011.

[15] M. Kearns and M. Li. Learning in the presence of malicious errors. *SIAM Journal on Computing*, 22:807–837, 1993.

[16] G. R. G. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and J. M. I. A robust minimax approach to classification. *Journal of Machine Learning Research*, 3:555–582, 2002.

[17] Z. Li, M. Sanghi, Y. Chen, M.-Y. Kao, and B. Chavez. Hamsa: Fast signature generation for zero-day polymorphic worms with provable attack resilience. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2006.

[18] W. Liu and S. Chawla. Mining adversarial patterns via regularized loss minimization. *Mach. Learn.*, 81:69–83, October 2010.

[19] D. Lowd. Good word attacks on statistical spam filters. In *In Proceedings of the Second Conference on Email and Anti-Spam (CEAS)*, 2005.

[20] D. Lowd and C. Meek. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 641–647, 2005.

[21] J. Newsome, B. Karp, and D. X. Song. Polygraph: Automatically generating signatures for polymorphic worms. In *2005 IEEE Symposium on Security and Privacy, 8-11 May 2005, Oakland, CA, USA*, pages 226–241. IEEE Computer Society, 2005.

[22] R. Perdisci, D. Dagon, W. Lee, P. Fogla, and M. Sharif. Misleadingworm signature generators using deliberate noise injection. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, pages 17–31, 2006.

[23] C. H. Teo, A. Globerson, S. T. Roweis, and A. J. Smola. Convex learning with invariances. In *Advances in Neural Information Processing Systems*, 2007.

[24] K. Wang, J. J. Parekh, and S. J. Stolfo. Anagram: A content anomaly detector resistant to mimicry attack. In *Recent Advances in Intrusion Detection, 9th International Symposium*, pages 226–248, 2006.

[25] G. L. Wittel and S. F. Wu. On attacking statistical spam filters. In *Proceedings of the first Conference on Email and Anti-Spam (CEAS)*, 2004.