Optimal Defense Actions Against Test-Set Attacks

Scott Alfeld[*] SALFELD@CS.WISC.EDU
Paul Barford[*][†] PB@CS.WISC.EDU
Xiaojin Zhu[*] JERRYZHU@CS.WISC.EDU
[*]Department of Computer Sciences, University of Wisconsin – Madison, Madison WI 53706, USA
[†]comScore, Inc. 11950 Democracy Drive, Suite 600 Reston, VA 20190, USA.

Abstract

Automated learning and decision making systems in public-facing applications are vulnerable to malicious attacks. These systems are at further risk of attack when money is involved, such as market forecasters or decision systems used in determining insurance or loan rates. In this paper, we consider the setting where a predictor Bob has a fixed model, and an unknown attacker Alice aims to perturb (or *poison*) future test instances so as to alter Bob's prediction to her benefit. We define a general framework for determining Bob's optimal defense action against Alice's worst-case attack. We then demonstrate our framework by considering linear predictors, where we provide tractable methods of determining the optimal defense action. Using these methods, we perform an empirical investigation of optimal defense actions for a particular class of linear models – autoregressive forecasters – and find that for ten real world futures markets, the optimal defense action reduces the predictor's loss by between 78 and 97%.

1. Introduction

Systems in domains such as finance, energy, medicine, entertainment, security, advertising, etc., increasingly rely on diverse input data. If decisions in these systems are based on the output of automated learning systems, then some actors may have incentive to alter (or *poison*) input data so as to affect the learned system. Thus, any such system should be robust to these threats.

The study of effective strategies in the presence of adversaries has long been of interest (Tzu, Circa 500 B.C.E.). In this work, we focus on the setting where an attacker alters data fed into an already learned model. Namely, we consider the setting where a predictor, Bob, has a fixed, publicly known prediction function. For example, Bob may be an insurance company estimating the expected future cost of an applicant to determine terms (*e.g.,* monthly premium) of the insurance plan offered. An actor, the adversary Alice with motivation unknown to Bob, asserts her limited control over the features (*e.g.*, to lie about her age, credit history, etc.) fed into Bob's prediction function, in aims to pull his prediction toward her goal. An attacker is defined by her target and her loss function (measuring the distance of Bob's resulting prediction to her target), both of which we assume are unknown to Bob. Bob may select some action from a set of available *defense actions*, and aims to limit the effectiveness of any potential attacker. In this work, we answer the question: What action should Bob take to best defend against an unknown attacker, assuming worst case initial values, attacker target, and attacker loss function?

We address the issue of defense explicitly, by framing the interaction between attacker and predictor as a two player, non-zero sum, Stackelberg game. Specifically, in this work we make three primary contributions: (i) We define a general framework for a predictor's explicit defense strategy against intelligent, unknown adversaries. (ii) We utilize the framework to provide tractably computable optimal actions for linear predictors. (iii) We empirically demonstrate our methods on real world data sets, and perform a investigation of their properties on synthetic data.

2. Defense Framework

An agent Bob is a predictor with a fixed function mapping instances in an input space $\mathcal{X}$ to target values in an output space $\mathcal{Y}$. We denote Bob's fixed, presumably learned, prediction function as $f : \mathcal{X} \to \mathcal{Y}$. Our framework is applicable to general prediction. That is, Bob's task may be, *e.g.,* (binary) classification ($f : \mathbb{R}^d \to \{0, 1\}$) regression ($f : \mathbb{R}^d \to \mathbb{R}$), clustering (hard or soft) ($f : \mathbb{R}^d \to [1, \ldots, k]$ or $f : \mathbb{R}^d \to \mathcal{S}_k$), rank-

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

ing ($\mathbb{R}^{d\times n} \to \binom{\{1,\dots,k\}}{k}$)) or other forms of prediction. For ease of notation we assume $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} \subset \mathbb{R}^m$, where $d, m \in \mathbb{Z}_+$.

Alice is an adversary with limited ability to perturb or *poison* test instances before Bob observes them. She aims to perform an *attractive* (Alfeld et al., 2016) attack, moving Bob's prediction towards some target. After observing a test instance[1] $\boldsymbol{x} \in \mathcal{X}$, Alice will select a *poison* vector $\boldsymbol{\alpha}^{\text{atr}}$ and supply Bob with the poisoned instance $\boldsymbol{x} + \boldsymbol{\alpha}^{\text{atr}}$. We define Alice in terms of: (i) Her target $\boldsymbol{t} \in \mathcal{Y}$, which she aims to pull Bob's prediction toward. (ii) Her loss function $\|\cdot\|_A$, where $\|\boldsymbol{0}\|_A = 0$ and $\|\boldsymbol{a}\|_A > 0\ \forall \boldsymbol{a} \neq \boldsymbol{0}$. (iii) Her set of feasible attacks $\mathcal{A}$. (iv) Her effort function $g(\cdot) : \mathcal{A} \to \mathbb{R}$, defining the costs she incurs for a given attack ($g(\boldsymbol{\alpha}) \geq 0\ \forall \boldsymbol{\alpha}$).

We assume a powerful attacker. Namely, Alice has full knowledge of Bob, and will select the attack which minimizes the sum of her loss and effort. Formally, Alice selects the optimal attack by solving:

$$\begin{aligned}\boldsymbol{\alpha}^{\text{atr}}\left(\mathcal{A}, \boldsymbol{x}, \boldsymbol{t}, \|\cdot\|_A, g(\cdot)\right) & \qquad (1)\\ \triangleq \arg\min\nolimits_{\boldsymbol{\alpha}\in\mathcal{A}} \|f(\boldsymbol{x}+\boldsymbol{\alpha}) - \boldsymbol{t}\|_A + g(\boldsymbol{\alpha}) &\end{aligned}$$

For a variety of settings, there are known, tractable methods for computing Alice's optimal attack (Alfeld et al., 2016). We instead focus on Bob, defining a framework for determining his optimal method of defending against an unknown adversary Alice.

We phrase the interplay between Alice and Bob as a one-shot, two-player, non-zero-sum, Stackelberg game. For brevity, we restrict our attention to settings where Bob considers only pure (as opposed to mixed) strategies, and his actions are to further restrict Alice. Our methods, however, extend beyond this. In further interest of clarity, we make the order of events explicit: (1) Bob selects action $\beta \in \mathcal{B}$. (2) Alice observes f, β, and $\boldsymbol{x}$. (3) Alice selects her poison vector $\boldsymbol{\alpha}^{\text{atr}}$ (from $\mathcal{A}$ constrained by β). (4) Bob observes $\boldsymbol{x}+\boldsymbol{\alpha}^{\text{atr}}$, and suffers loss $\|f(\boldsymbol{x}+\boldsymbol{\alpha}^{\text{atr}}) - f(\boldsymbol{x})\|_B$. Note that Bob does not observe his loss, as he never observes the unpoisoned $\boldsymbol{x}$.

In keeping with the assumption of a powerful attacker, we assume that Bob does not know Alice's target, loss function, or effort function, but he does know her constraints (defining $\mathcal{A}$). This allows our methods to be used in evaluating the robustness of a system against bounded attackers – Bob can evaluate the worth of limiting an attackers abilities. We further assume that Bob does not know the unpoisoned value $\boldsymbol{x}$ (if he did, he could simply undo Alice's attack). Bob aims to minimize the deviation, as defined by his loss function $\|\cdot\|_B$, between his prediction on the poisoned test set and what he would have predicted on the unpoisoned set. To do this, Bob selects some action $\beta \in \mathcal{B}$. By selecting action $\beta \in \mathcal{B}$, Bob reduces Alice's feasible set of attacks to $\mathcal{A}_\beta$. Formally, Bob seeks to solve the bi-level optimization problem:

$$\begin{aligned}\arg\min_{\beta\in\mathcal{B}} \max_{\boldsymbol{x}, \|\cdot\|_A, g(\cdot), \boldsymbol{t}, \boldsymbol{\alpha}^{\text{atr}}} & \left\|f(\boldsymbol{x}) - f(\boldsymbol{x}+\boldsymbol{\alpha}^{\text{atr}})\right\|_B \qquad (2)\\ \text{s.t. } \boldsymbol{\alpha}^{\text{atr}} = \arg\min_{\boldsymbol{\alpha}\in\mathcal{A}_\beta} & \|f(\boldsymbol{x}+\boldsymbol{\alpha}) - \boldsymbol{t}\|_A + g(\boldsymbol{\alpha})\end{aligned}$$

To simplify, we exploit the duality between considering all possible attractive attacks, and considering the one *repulsive* attack – the attack which explicitly aims to maximize Bob's loss. We define the following single-level optimization problem, equivalent to (2) (we omit the proof for brevity):

$$\arg\min_{\beta\in\mathcal{B}} \max_{\boldsymbol{x}\in\mathbb{R}^d, \boldsymbol{\alpha}\in\mathcal{A}_\beta} \|f(\boldsymbol{x}) - f(\boldsymbol{x}+\boldsymbol{\alpha})\|_B \qquad (3)$$

In essence, we are creating a phantom adversary performing the repulsive attack: $\boldsymbol{\alpha}^{\text{rep}} \triangleq \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \|f(\boldsymbol{x}) - f(\boldsymbol{x}+\boldsymbol{\alpha})\|_B$ We then have Bob defend against this phantom Alice, thus limiting the potential effect of any attacker. In doing so, we formulate the problem in standard minimax form rather than a bi-level optimization problem. Rather than maximizing over all possible targets, effort, and loss functions, we now maximize over only the potential initial values $\boldsymbol{x}$.

3. Linear Predictors

Thus far we have defined a framework (eqn (3)) for selecting an optimal defense action for a general predictor. In what follows, we utilize this framework, and describe instantiations of Alice and Bob inspired by real-world settings. These instantiations result in tractable methods for determining Bob's optimal defense action. For simplicity, we assume that $\mathcal{B}$ is a finite set. In this setting, the task of determining Bob's optimal defense action reduces to computing the optimal repulsive attack. Cases where Bob has a continuous or countable infinite set of actions are left as future work.

When Bob's loss is convex, it is often tractable to compute Alice's optimal attractive attack – that is, minimizing a quadratic function subject to hard constraints. However, even when Bob's loss is convex, the task of computing an optimal *repulsive* attack – *maximizing* a quadratic function subject to hard constraints – is NP-Hard in general (e.g., under box constraints) (Nocedal & Wright, 2006). We consider a subset of all predictors (Bobs) and attackers (Alices) so as to yield tractable methods for solving (3).

[1] To avoid cluttersome notation we assume only one test instance. All methods described herein, however, extend easily to the case where Bob receives a test set of more than one point.

We consider the case where $\mathcal{Y}$ is continuous (e.g., regression) and let Bob be a (homogeneous) linear predictor. That is, his prediction function may be written as:

$$f(\boldsymbol{x}) \triangleq M\boldsymbol{x} \tag{4}$$

for some matrix M. We note that by linearity of f,

$$\begin{aligned}\boldsymbol{\alpha}^{\text{rep}}\left(\mathcal{A}, \boldsymbol{x}, \|\cdot\|_B\right) &\triangleq \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \|f(\boldsymbol{x}+\boldsymbol{\alpha}) - f(\boldsymbol{x})\|_B \\ &= \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \|f(\boldsymbol{\alpha})\|_B \qquad (5)\\ &\triangleq \boldsymbol{\alpha}^{\text{rep}}\left(\mathcal{A}, \|\cdot\|_B\right) \qquad (6)\end{aligned}$$

That is, *the optimal repulsive attack is independent of the test instance*. This results in (3) being equivalent to:

$$\begin{aligned}&\arg\min_{\beta\in\mathcal{B}} \max_{\boldsymbol{\alpha}\in\mathcal{A}_\beta} \|f(\boldsymbol{\alpha})\|_B \qquad (7)\\ &= \arg\min_{\beta\in\mathcal{B}} \|f(\boldsymbol{\alpha}^{\text{rep}}\left(\mathcal{A}_\beta, \|\cdot\|_B\right))\|_B \qquad (8)\end{aligned}$$

We let Bob's loss be the squared Mahalanobis norm:

$$\|f(\boldsymbol{\alpha})\|_B \triangleq \|f(\boldsymbol{\alpha})\|_W^2 = f(\boldsymbol{\alpha})^\top W f(\boldsymbol{\alpha}) \tag{9}$$

where $W = V^\top V$ is a positive-definite matrix. This loss function generalizes mean squared error. It both yields mathematical benefits, and is applicable in many real world settings. We consider the case where each of Bob's actions restricts Alice to select an attack from some ellipse. That is, each of Bob's actions $\beta \in \mathcal{B}$ defines Alice's feasible attacks as:

$$\mathcal{A}_\beta \triangleq \{\boldsymbol{\alpha} \;:\; \|\boldsymbol{\alpha}\|_{C_\beta} \leq c\} \tag{10}$$

where $\|\boldsymbol{\alpha}\|_{C_\beta} = \sqrt{(C_\beta\boldsymbol{\alpha})^\top C_\beta\boldsymbol{\alpha}}$, $C_\beta = G_\beta^\top G_\beta$ is a positive-definite matrix and $c \in \mathbb{R}_+$.

Under these conditions, we may leverage the alignment of Bob's loss with Alice's constraints (in that they are both quadratic) to convert the optimization problem to that of computing an induced matrix norm. With proof omitted for brevity, we have:

$$\boldsymbol{\alpha}^{\text{rep}}\left(\mathcal{A}_\beta, \|\cdot\|_B\right) \triangleq cG_\beta^{-1}\boldsymbol{s}_1 \tag{11}$$

where $\boldsymbol{s}_1$ is the right-singular vector corresponding to the largest singular value of VMG_β^{-1}. Further, Bob's loss against the worst case attack is

$$\|f\left(\boldsymbol{\alpha}^{\text{rep}}\left(\mathcal{A}_\beta, \|\cdot\|_B\right)\right)\|_B = \|cVMG_\beta^{-1}\|_2 \tag{12}$$

where $\|\cdot\|_2$ denotes the spectral norm. Therefore $\boldsymbol{\alpha}^{\text{rep}}$ may be found by computing G_β^{-1} and the SVD of VMG_β^{-1}, or approximated (*e.g.,* for large matrices) by applying the power method to $(VMG_\beta^{-1})^\top(VMG_\beta^{-1})$. By computing this quantity for each $\beta \in \mathcal{B}$ (recall $\mathcal{B}$ is finite), Bob may determine which action minimizes (7). We note that Bob need only use G_β^{-1}, not C_β or G_β when determining his defense action. We demonstrate the usefulness of this fact in the following section.

4. Experiments

While the framework we have described is broadly applicable to linear predictors, here we focus on the setting where Bob is forecasting future values of a time series. Specifically, we use a linear autoregressive model and recursive forecasting strategy. We select this setting because (a) data manipulation attacks are a real-world concern in forecasting, and (b) prior work (Alfeld et al., 2016) has determined optimal (attractive) attacks against linear forecasters, specifically evaluating the efficacy in the context of futures markets' settle prices.

Bob uses the values for the last d time periods $x_{-d}, \ldots, x_{-1}$ to forecast the next h values into the future $(\hat{x}_0, \ldots, \hat{x}_{h-1})$. He does so with an order-d linear autoregressive model: $x_t = \sum_{i=1}^d \theta_i x_{t-i}$, and recursive forecasting strategy. Without loss of generality we assume $h > d$. We note that the forecasting function may be written as:

$$f\left(\boldsymbol{x}\right) = M\boldsymbol{x} \triangleq S^h Z\boldsymbol{x} \tag{13}$$

Where S is the $h \times h$ one-step matrix for model $\boldsymbol{\theta}$, and Z is the $h \times d$ zero-padding matrix.

$$S \triangleq \begin{bmatrix} \mathbf{0}_h & I_{h-1\times h-1} \\ \mathbf{0}_{(h-d-1)\times 1}^\top & \overleftarrow{\boldsymbol{\theta}}^\top \end{bmatrix}, Z \triangleq \begin{bmatrix} \mathbf{0}_{(h-d)\times d} \\ I_{d\times d} \end{bmatrix}$$

Where we denote the reverse of $\boldsymbol{\theta}$ as $\overleftarrow{\boldsymbol{\theta}} : \overleftarrow{\theta}_i = \theta_{d-i+1}$.

In defining Bob's set of possible actions, we consider performing an inspection on a single time period. In the general setting of prediction, this is akin to Bob independently verifying a single feature. For simplicity of demonstration, we let one time period be one day, and assume that on the day Bob performs an inspection, Alice is unable to lie – on all other days, Alice is bound by her original set of feasible attacks $\mathcal{A}$. We assume that Alice's original set $\mathcal{A}$ of feasiable attacks is an d-ellipse defined by $\{\boldsymbol{\alpha} \;:\; \|\boldsymbol{\alpha}\|_C \leq c\}$ and Bob is therefore restricting her to a $(d-1)$-ellipse. Recall that Bob's actions may be defined in terms of G_β^{-1} directly. By letting $G_\beta^{-1}[i,j] = G_\beta^{-1}[j,i] = 0$ for all j, we encode Alice's inability to affect day i; her value for $\boldsymbol{\alpha}_i$ is ignored.

For brevity, we consider only $d = 5, h = 7$ with spherical loss (total squared deviation) for Bob and spherical constraints for Alice for each experiment: $W = C = I$.

4.1. Futures Markets Experiments

For ten different futures markets, we obtained[2] daily settle price data. For each, we estimated $\boldsymbol{\theta}$ using Yule-

[2] Data is freely available from www.quandl.com. Identification codes for individual datasets are provided in Figure 1.

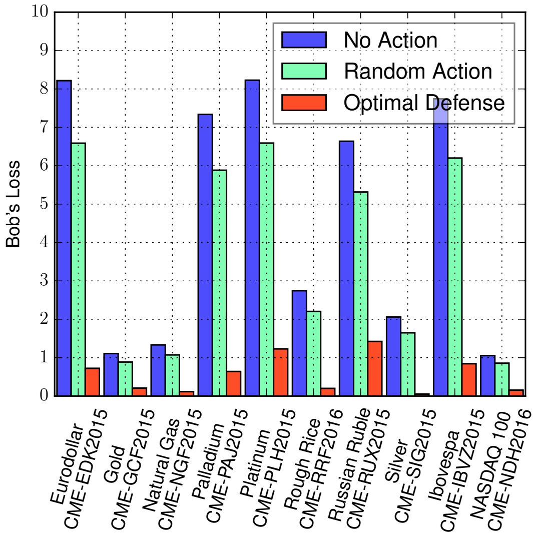

Figure 1: For each the 10 futures markets, we denote Bob's loss against the worst case attack in red. In blue we show Bob's loss against the worst case attack given his optimal defense action.

Walker estimation (Box et al., 2011) on approximately one month's worth of centered data – the exact dates used varied across markets based on values available. We compared Bob's loss under the optimal strategy, selecting an action at random, and the null strategy where he takes no action on each future market. We report loss under all three strategies in Figure 1. Universally, taking the optimal defense action considerably reduces Bob's worst-case loss – defending resulted in a 78% (Russian Ruble) to $> 97\%$ (Silver) reduction in loss compared to the null strategy across the ten markets. We note that on each futures market, the optimal action was to lock down day -1 (the last day), and for each learned model we find $|\theta_1| > |\theta_i|, i = 2, \dots, 5$.

4.2. Synthetic Experiments

From the futures markets data, one may be tempted to form two natural hypotheses: (a) The optimal action is always to select the day $i^{\max}$ corresponding to the maximal (in magnitude) θ_i: $i^{\max} \triangleq -\arg\max_i |\theta_i|$. (b) The optimal action is always to select the last day. Hypothesis (a) is supported by the observation that Bob's first prediction will be most affected by the value $\boldsymbol{\alpha}_i^{\max}$, and subsequent predictions will in turn be affected by the first. Hypothesis (b), in contrast, is motivated by the observation that while x_{-d} directly affects only $\hat{x}_0$ (all later predictions are affected by x_{-d} only through $\hat{x}_0$), the value x_{-1} directly affects predictions $\hat{x}_0, \dots, \hat{x}_{d-1}$.

To test these hypotheses, we run an additional experiment. To emulate models that may be encountered in practice, we construct 10,000 stationary models $\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(10000)}$ by drawing each $\boldsymbol{\theta}^{(i)}$ *iid* from a unit Gaussian, and then rejecting any non-stationary samples. We then determine the percentage of models on which hypotheses (a) and (b) yield the optimal defense action. We find that selecting the day corresponding with the maximal θ_i (hypothesis (a)) is optimal only $\approx 55\%$ of the time. Selecting the last day (hypothesis (b)) is optimal only $\approx 49\%$ of the time.

5. Related Work

The setting of so-called test-set attacks has been examined under a variety of titles. One such example is "evasion attacks", where the predictor performs binary classification (*e.g.,* spam detection (Nelson et al., 2009; Lowd & Meek, 2005), intrusion detection (Tan et al., 2002)) and the attacker aims to have their bad (*e.g.,* "spam" or "intrusion") sample classified as good (*e.g.,* "ham" or "normal traffic"). Robust Learning (Globerson & Roweis, 2006; El Ghaoui et al., 2003), considers the setting where a test set is drawn from a distribution distinct from the training set's. The setting presented herein is distribution free, and an example covariate shift (Quionero-Candela et al., 2009). (Goodfellow et al., 2015) argues that linearity in the models is a primary cause of attack vulnerability. This theory is supported by in our work and warrants further investigation.

A primary goal of this line of research is defense. We borrow from the framework and methodology used in (Alfeld et al., 2016), which derived optimal (attractive) attacks against autoregressive forecasters. A separate line of research has posed the problem of learning in the presence of adversaries in game theoretic contexts ((Liu & Chawla, 2009; Brückner et al., 2012; Dalvi et al., 2004; Brückner & Scheffer, 2009; 2011)). (Dalvi et al., 2004) and (Letchford & Vorobeychik, 2013) phrase the interplay between Alice and Bob as game similar to ours, and specifically addresses Bob's defense strategy.

6. Conclusions

The framework for our study is a predictor targeted by an attacker which seeks to influence its predictions. Our goal is to identify an optimal defense against such an attack. We allowed for a powerful, knowledgeable attacker, yielding a two player, non-zero sum Stackelberg game. By, in essence, constructing a phantom attacker based on Bob's loss function, we are able to phrase this interplay as a standard minimax formulation. We utilize our framework to identify the optimal defense action for worst-case attacks against linear predictors. In future work, we plan to apply our methods to non-linear predictors, in hopes to derive tractable methods of identifying optimal defense actions.

References

Alfeld, Scott, Zhu, Xiaojin, and Barford, Paul. Data poisoning attacks against autoregressive models. In *AAAI*, 2016.

Box, G.E.P., Jenkins, G.M., and Reinsel, G.C. *Time series analysis: forecasting and control*, volume 734. Wiley, 2011.

Brückner, Michael and Scheffer, Tobias. Nash equilibria of static prediction games. In *Advances in neural information processing systems*, pp. 171–179, 2009.

Brückner, Michael and Scheffer, Tobias. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 547–555. ACM, 2011.

Brückner, Michael, Kanzow, Christian, and Scheffer, Tobias. Static prediction games for adversarial learning problems. *the Journal of Machine Learning Research*, 13(1):2617–2654, 2012.

Dalvi, Nilesh, Domingos, Pedro, Sanghai, Sumit, Verma, Deepak, et al. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 99–108. ACM, 2004.

El Ghaoui, Laurent, Lanckriet, Gert René Georges, Natsoulis, Georges, et al. *Robust classification with interval data*. Computer Science Division, University of California, 2003.

Globerson, Amir and Roweis, Sam. Nightmare at test time: robust learning by feature deletion. In *Proceedings of the 23rd international conference on Machine learning*, pp. 353–360. ACM, 2006.

Goodfellow, Ian J, Shlens, Jonathon, and Szegedy, Christian. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015.

Letchford, Joshua and Vorobeychik, Yevgeniy. Optimal interdiction of attack plans. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 199–206. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

Liu, Wei and Chawla, Sanjay. A game theoretical model for adversarial learning. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*, pp. 25–30. IEEE, 2009.

Lowd, Daniel and Meek, Christopher. Good word attacks on statistical spam filters. In *CEAS*, 2005.

Nelson, Blaine, Barreno, Marco, Chi, Fuching Jack, Joseph, Anthony D, Rubinstein, Benjamin IP, Saini, Udam, Sutton, Charles, Tygar, JD, and Xia, Kai. Misleading learners: Co-opting your spam filter. In *Machine learning in cyber trust*, pp. 17–51. Springer, 2009.

Nocedal, Jorge and Wright, Stephen. *Numerical optimization*. Springer Science & Business Media, 2006.

Quionero-Candela, Joaquin, Sugiyama, Masashi, Schwaighofer, Anton, and Lawrence, Neil D. *Dataset shift in machine learning*. The MIT Press, 2009.

Tan, Kymie MC, Killourhy, Kevin S, and Maxion, Roy A. Undermining an anomaly-based intrusion detection system using common exploits. In *Recent Advances in Intrusion Detection*, pp. 54–73. Springer, 2002.

Tzu, Sun. *The Art of War*. Circa 500 B.C.E.