2009 IEEE International Conference on Data Mining Workshops

A Game Theoretical Model for Adversarial Learning

Wei Liu
School of Information Technologies
The University of Sydney
Sydney, Australia
weiliu@it.usyd.edu.au

Sanjay Chawla
School of Information Technologies
The University of Sydney
Sydney, Australia
chawla@it.usyd.edu.au

***Abstract*—It is now widely accepted that in many situations where classifiers are deployed, adversaries deliberately manipulate data in order to reduce the classifier's accuracy. The most prominent example is email spam, where spammers routinely modify emails to get past classifier-based spam filters. In this paper we model the interaction between the adversary and the data miner as a two-person sequential noncooperative Stackelberg game and analyze the outcomes when there is a natural leader and a follower. We then proceed to model the interaction (both discrete and continuous) as an optimization problem and note that even solving linear Stackelberg game is NP-Hard. Finally we use a real spam email data set and evaluate the performance of local search algorithm under different strategy spaces.**

***Keywords*-Adversarial attacks; Stackelberg game; genetic algorithms;**

I. Introduction

The classification method has traditionally assumed that the training and test data are generated from the same underlying distribution. In practice this is far from true. Data evolves and the performance of deployed classifiers deteriorates. Part of the data evolution is due to natural drift. However there is increasing evidence that often there exists a sustained malicious effort to "attack" the classifier. The most prominent example is the rapid transformation of email spam to get around classification based spam filters. As a result, a new subfield of "adversarial learning" has emerged to understand and design classifiers which are robust to adversarial transformation[1], [2], [3], [4], [5], [6], [7].

Beginning from the work of Dalvi et. al [2] there has been an attempt to model adversarial scenarios. In their work the baseline assumption is that perfect information is available to both the classifier and the adversary: the classifier trains on data from a theoretical distribution D; the adversary is full aware of the decision boundary of the classifier and modifies the data to D' to get past the classifier; the classifier in turn retrains to create a new decision boundary. This process can potentially proceed infinitely. However, the key idea in game theory is that of an equilibrium: a state from which neither the classifier nor the adversary will have any incentive to deviate. In order to relax the assumption of perfect information, Lowd et. al [4] assume that the adversary has the ability to issue a polynomial number membership queries to the classifier in the form of data instances which in turn will report their labels. They refer to their approach as Adversarial Classifier Reverse Engineering (ACRE). However, they still do not model an equilibrium scenario and how the classifier will respond after ACRE learning. In practice, the ACRE learning quantifies the "hardness" of attacking a (linear) classifier system. More recently, Kantarcioglu et. al.[7] have proposed to model the adversarial classification as a sequential game (aka Stackelberg game) in which the adversary makes the first move to which the classifier responds. While our approach also uses the Stackelberg model we completely relax the assumption that the adversary knows about the classifier's payoff.

Our contributions in this paper are as follows:

1) We introduce Stackelberg games to model the interaction between the adversary and the data miner, and show how to infer the equilibrium strategy. We model the situation where the strategy space can be both finite and infinite.
2) We propose the use of genetic algorithms to solve the Stackelberg game for the infinite case where the players do not need to know each other's payoff function.

The rest of the paper are as follows. The game between the adversary and the data miner with finite strategy space is introduced in Section II. In Section III we formulate Stackelberg games with infinite strategy space. The game theoretical model components and the derivations of the players payoff functions are introduced in Section IV. Section V explains the genetic algorithms we design to search for equilibrium. Experiments are conducted in Section VI by using both synthetic and real data sets. We state our conclusions and future research directions in Section VII.

II. The Spammer and Data Miner Game

We begin by contextualizing a two person game between the spammer (S) and the data miner (D) (Fig. 1). We assume the strategy space of the spammer consists of two actions: Attack and Status Quo. The spammer can choose to attack the classifier by actively modifying spam emails in order to get through, or maintain the status-quo with the knowledge

978-0-7695-3902-7/09 $26.00 © 2009 IEEE
DOI 10.1109/ICDMW.2009.9

that no classifier is perfect and that some spam emails will still get through. Similarly, the data miner can take two actions: Retrain and Status Quo. The data miner can choose to retrain the classifier in order to lower the error rate or maintain the status quo and tolerate a potential increase in spam emails getting through (assuming the underlying data changes). We also assume that the spammer will make the first move and then the data miner will follow by taking one of the two possible actions.

The game has four possible outcomes so we label the payoff from 1 through 4 for both the two players. Here is how we rank the outcomes.

1) The spammer can choose to attack the classifier and the data miner can ignore and maintain the status quo (i.e., not retrain). This is the best scenario for the spammer and the worst case for the data miner and thus we give a payoff of 4 and 1 to the spammer and data miner respectively. The payoffs are shown next to the leaf nodes of the game tree in Fig. 1a.
2) The spammer can choose to attack and the data miner can retrain. This is like a tie and each players gets a payoff of 2 each.
3) The spammer can choose not to attack (i.e., maintain status quo) and the data miner can choose to retrain the classifier with the belief that more data will always improve the classifier. However, in this case there is a cost of retraining which must be factored in. Thus the payoff for the data miner is 3. This situation is in some sense the worst case for the spammer as, everything else being equal, the classifier will improve with time.
4) Finally, both the spammer and the data miner can maintain status quo. This is the best case scenario for the Classifier and the second best option for the spammer as some spam emails will always get through without taking on additional cost to transform the data distribution. Thus in this case we assign a payoff of 3 and 4 to the spammer and data miner respectively.

The key idea of determining the equilibrium strategy is **rollback** (also known as **backward induction**) which works as follows: *since the leader (spammer) will make the first move she*[1] *knows that a rational follower will react by maximizing the follower's payoff. The leader takes that into account before making the first move.*

Once the spammer makes a move, the play can proceed along the top or bottom branch from the root of the tree in Fig. 1a. If the play is at the upper branch, the data miner will choose the top leaf out of the first level as that will be the local maxima for its payoff (2 vs. 1). This is why the second leaf is pruned in Fig. 1b. Similarly when the competition is within lower branch the third leaf is pruned as the data miner gets a higher payoff when play proceeds to the forth leaf (4 vs. 3). The spammer is fully aware how a rational data miner will proceed and thus chooses the path which maximizes her payoff. This explains why the game will proceed along the (SQ, SQ) path along the bottom of the tree (Fig. 1c). Once at the equilibrium neither of them have any incentive to deviate, and the spammer and the data miner settle for an equilibrium payoff of 3 and 4 respectively.

[1]In the Game Theory literature there is a tradition of having female "she" players.

III. Infinite Stackelberg Games

The key idea in sequential games is the idea of *rollback* (*backward induction*). The leader, who is the spammer in our case, has a natural advantage as she can take into account the optimal strategy of the follower before making the initial move.

The goal of this section is to operationalize the idea of rollback mathematically. This will set the stage for the next section where we will drop the assumption that the players are aware of each others payoff functions. We focus on the Stackelberg games as they explicitly distinguish between a leader and a follower.

A. Definitions

The following are the components of a two-person Stackelberg game:

1) A game is played between two players the leader (L) and the follower (F). In our case the spammer is the leader and the data miner is the follower. The leader always makes the first move.
2) Associated with each player is a space (set) of actions (strategies), U and V for L and F respectively. For simplicity we assume that U and V are bounded and convex.
3) Also associated with each player is a payoff function J_L and J_F such that each J_i (i=L,F) is a twice-differentiable mapping $J_i(U, V) \rightarrow R$, where R stands for reaction.

Each player reacts to the other's move through a reaction function. Thus the reaction function of L, $R_L : V \rightarrow U$ is

$$R_L = \arg\max_{v \in V} J_L(u, v) \tag{1}$$

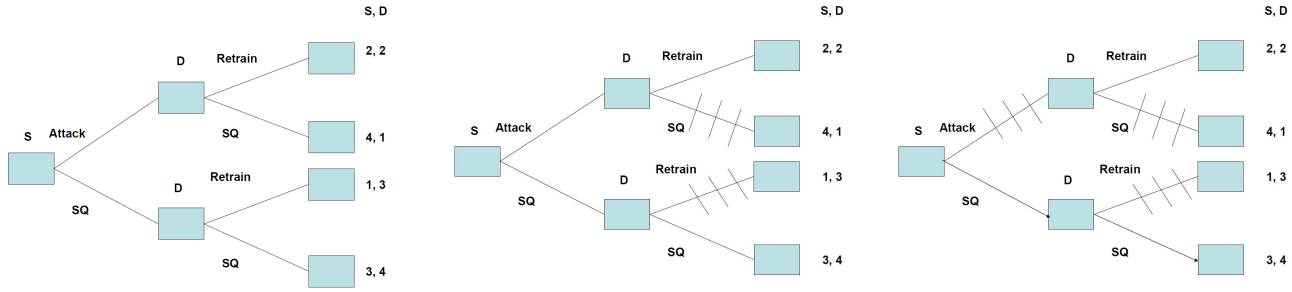
Similarly, the reaction function of F, $R_F : U \rightarrow V$ is

$$R_F = \arg\max_{u \in U} J_F(u, v) \tag{2}$$

B. Rollback as an Optimization Problem

The principle of rollback tells that the leader, who makes the first move, anticipates that rational followers will maximize their payoff in their reactions, and incorporates that knowledge before making the first move. Mathematically, the leader's action is the solution to the following optimization problem.

$$u^s = \arg\max_{u \in U} J_L(u, R_F(u)) \tag{3}$$



(a) Four outcomes with payoff sets from four different combinations of strategies.

(b) Two leaves are pruned from the perspective of data miner's payoff.

(c) The upper branch is pruned by the spammer based on the data miner's possible reactions.

Figure 1: Game tree for Stackelberg model between the spammer (*S*) and the data miner (*D*). "SQ" stands for status quo; "Retrain" means retraining the classifier.

The follower then reacts with the optimal action

$$v^s = R_F(u^*) \tag{4}$$

The pair (u^s, v^s) is the Stackelberg equilibrium.

In contrast, the Nash equilibrium (u^n, v^n) for a game in which the two players act simultaneously is given by the solution of the simultaneous equations $R_L = 0$, $R_F = 0$. [8] gives comprehensive examples illustrating the different calculations on obtaining the Nash and Stackelberg games; it also states how we express Stackelberg equilibrium as a Bilevel Programming Problem.

IV. The Game Theoretic Model in Classification Problems

We model the game between the spammer and the data miner as a two-class classification problem with varying data distributions. For simplicity, we assume the data are from one dimensional feature space.

We denote the distribution of the spams by $P(\mu_1, \sigma)$ and the legitimate emails by $Q(\mu_2, \sigma)$. Assume that $\mu_1 < \mu_2$ (Fig. 2a). Adversary plays by moving μ_1 to $\mu_1 + u$ (towards μ_2) as shown in Fig. 2b, while the classifier reacts by moving boundary from $\frac{\mu_1+\mu_2}{2}$ to w (also towards μ_2) as shown in Fig. 2c. We constrain that $\mu_1 \leq u \leq \mu_2 - \mu_1$, and $\frac{\mu_1+\mu_2}{2} \leq w \leq \mu_2$.

To estimate the influence of transformation u on the original intrusion data, we use the Kullback-Leibler divergence (KLD) to measure the effects of transformation from $N_1(\mu_1, \Sigma_1)$ to $N_2(\mu_2, \Sigma_2)$ [9]:

$$\begin{aligned} D_{KL}(N_1|N_2) &= \frac{1}{2}(\log_e(\frac{det\Sigma_2}{det\Sigma_1})) + tr(\Sigma_2^{-1}\Sigma_1) \\ &+ (\mu_2 - \mu_1)^{Transpose}\Sigma_2^{-1}(\mu_2 - \mu_1) - q \end{aligned} \tag{5}$$

where det and tr stands for the determinant and trace of matrices, and q is the number of features in an attribute. KLD explains compared to the information need to explain the distribution N_1, how much extra information is required to explain N_2. Examples demonstrating the effects of KLD can find from [8].

From the probability density function (pdf) of a Normal distribution $N(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, we obtain the cumulative pdf of a Normal distribution $F(t, \mu, \sigma) = \int_{-\infty}^{z} N(t, \mu, \sigma)\, dx$. We define the payoff for the adversary as the increase in the false negative rate (FNR) minus the KLD of the transformation:

$$\begin{aligned} J_L(u, w) &= FNR - \alpha KLD(\mu_1 + u, \sigma, \mu_1, \sigma) \\ &= 1 - F(w, \mu_1 + u, \sigma) - \alpha KLD(\mu_1 + u, \sigma, \mu_1, \sigma) \end{aligned} \tag{6}$$

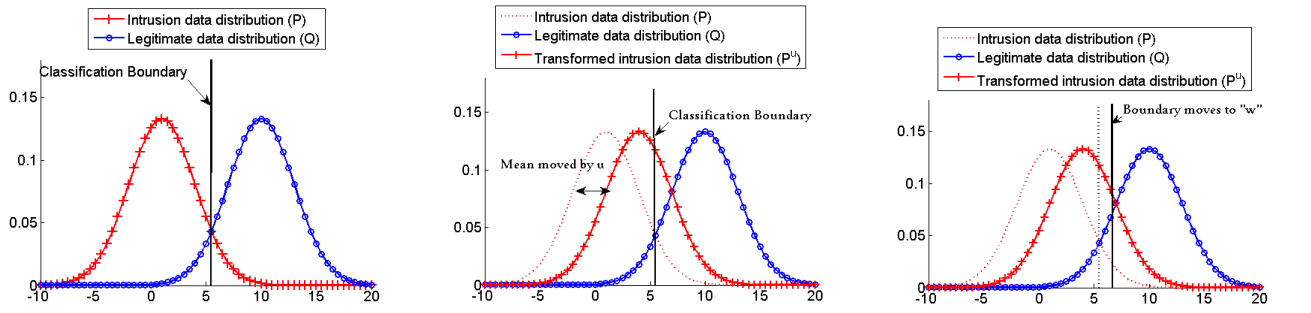
The parameter α in Equation 6 determines the strength of the KLD penalty. We also call the value of the leader's payoff as the *adversarial gain*.

The payoff of the classifier is given by increasing both the true positive and the true negative rate (TPR and TNR) minus the cost of moving the boundary:

$$\begin{aligned} J^F(u, w) &= TPR + TNR - \beta(w - \frac{\mu_1 + \mu_2}{2})^2 \\ &= F(w, \mu_1 + u, \sigma) - F(w, \mu_2, \sigma) + (1 - F(w, \mu_2, \sigma)) \\ &\quad - (1 - F(w, \mu_1 + u, \sigma)) - \beta(w - \frac{\mu_1 + \mu_2}{2})^2 \\ &= 2F(w, \mu_1 + u, \sigma) - 2F(w, \mu_2, \sigma) - \beta(w - \frac{\mu_1 + \mu_2}{2})^2 \end{aligned} \tag{7}$$

Similar to α, the term β controls the strength of the cost of the boundary adjustment.

When there are multiple attributes, we assume all attributes are conditionally independent given their class labels, and also independently transformed by the spammer. Given adversarial transformation u, we denote the distributions of spam and legitimate instances by $P(\mu_P, \Sigma_P)$ and $Q(\mu_Q, \Sigma_Q)$, and the distribution of transformed spams by $P^u(\mu_{P^u}, \Sigma_{P^u})$, where $\mu_{P^u} = \mu_P + u_{mu}$, $\Sigma_{P^u} = \Sigma_P + u_{sigma}$, and $(u_{mu}, u_{sigma}) = u$ (we assume the mean and standard deviation are separately transformed). Thus the



(a) Initial model state

(b) The adversary's movement

(c) The data miner's reaction to the adversary's movement

Figure 2: Three status of the game theoretical model in classification scenario. The vertical lines represent the classification boundary built by naive bayesian classifier.

leader's (the spammer) and the follower's (the data miner) payoff functions can be defined as follows:

$$J_L(U,W) = \frac{1}{q}\sum_{i=1}^{q}(1 - F(w_i, \mu_i^1 + u_i^\mu, \sigma_i^1 + u_i^\sigma) - \alpha KLD(\mu_i^1, \sigma_i^1, \mu_i^1 + u_i^\mu, \sigma_i^1 + u_i^\sigma))) \quad (8)$$

$$J_F(U,W) = \frac{1}{q}\sum_{i=1}^{q}(2F(w_i, \mu_i^1 + u_i^\mu, \sigma_i^1 + u_i^\sigma) - 2F(w, \mu_i^2, \sigma_i^2) - \beta(w_i - \frac{\mu_i^1 \times \sigma_i^2 + \mu_i^2 \times \sigma_i^1}{\sigma_i^1 + \sigma_i^2})^2) \quad (9)$$

where q is the number of attributes, μ_i^j and σ_i^j are the mean and standard deviation of the ith feature of class j (j=1 for the spam class, and j=2 for the legitimate class), $U = (u_1^\mu, u_2^\mu, ..., u_q^\mu, u_1^\sigma, u_2^\sigma, ..., u_q^\sigma)$ is the strategy of a certain play of the adversary, and $W = (w_1, w_2, ..., w_q)$ is the reconstructed classification boundary as the data miner's reaction. The effect of parameter α in Equation 6 and 8, and β in Equation 7 and 9 is analyzed in Section VI. Denote the data miner's best reaction given adversarial transformation U by $R_F(U)$ (i.e. $W = R_F(U)$) subject to $J_F(U,W)$, then the optimization problem for solving Stackelberg game, explained in Equation 3, can be restated as:

$$U^s = \arg\max_{U \in U^{all}} J_L(U, R_F(U)) \quad (10)$$

V. Genetic Algorithms

We use Genetic algorithms (GA) to solve Equation 10 (Algorithm 1). The best transformation of the last generation is returned as equilibrium transformation u^s. Detailed explanations of this algorithm can be obtained from [8].

Algorithm 1 Genetic Algorithm for Solving Stackelberg Equilibrium

Input: Number of individuals in a generation k
Output: Stackelberg equilibrium transformation u^s

1: Randomly generate k transformations u_i, i∈(1,2,...,k);
2: Initiate the best adversarial gain *BestGain* ← 0;
3: **repeat**
4: **for** i = 1 to k **do**
5: The adversary apply transformation u_i;
6: The data miner reacts classifier $R_F^{u_i}$;
7: The adversarial gain produced by u_i is evaluated by the adversarial payoff function $J_L(u_i, R_F^{u_i})$
8: **end for**
9: Among all u_i, identify u^s with the highest adversarial payoff $J_L(u^s, R_F^{u^s})$;
10: *ImprovedGain* = $J_L(u^s, R_F^{u^s})$ – *BestGain*;
11: *BestGain* ← $J_L(u^s, R_F^{u^s})$;
12: Create new generation of transformations by selection, mutation and crossover of the old generation;
13: **until** *ImprovedGain* == 0.
14: Return u^s;

VI. Experiments

In this section, we use both synthetic and real data to demonstrate the process of searching for an equilibrium by genetic algorithms [2].

A. Experiments on Synthetic Data

We use one dimensional feature space to create the synthetic data set, the distributions of spam and legitimate instances are N1(1,2) and N2(10,2), respectively. Then as explained in Section IV, "u" is restricted between [0, 9], and "w" between [5.5, 10]. The effects of α (in Equation 8) and

[2]All source code and data sets used in our experiments can be obtained from *http://www.it.usyd.edu.au/~weiliu/DDDM09*.

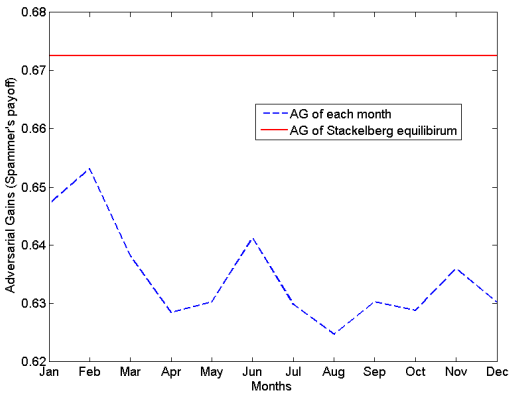


Figure 4: Adversarial adversarial gain from each month is bounded by the equilibrium adversarial gain. "AG" in the figure legend means adversarial gain.

β (in Equation 9) on Stackelberg equilibrium in the leader's and follower's payoff is shown in Table I. The classification errors produced at equilibrium are presented by false positive rate (FPR) and false negative rate (FNR): while FPR tells what percentage of legitimate instances are wrongly detected as intrusions, FNR gives the proportion of intrusions that are undetected.

When β is fixed and α is zero, the spammer has no cost in shifting the spam data, and hence always makes μ_1 completely overlap with μ_2 (u = 9 in the first three rows of Table I); but when α is non-zero (e.g. α = 0.1), the adversary can hardly move μ_1 (u $<$ 1 in the last three rows) before the equilibrium is achieved. This information suggests the data miner make more use of features that are more expensive to be transformed, if it is critical to constraint the adversary's actions.

The parameter β takes effects when the data miner tries to reconstruct the classifier, and it makes the classifier favor previous decision boundaries after retraining. When β is zero, the new classification boundary is relocated at the average of the new means (i.e. $\frac{(\mu_1+u)+\mu_2}{2}$), and generates the same FPR and FNR. However, when the penalty of β is non-zero (e.g. β = 0.1), the classifier puts the boundary close to its original position (close to 5.5 in our example), resulting in a considerable number of spams unfiltered, and generating larger FNR compared to FPR. To this end, the data miner should put more weight on the penalty of boundary relocation if the cost of increasing FNR is higher than that of FPR; or put less weight if the data miner concerns more about the overall accuracy.

The results from various combinations of parameter settings prove the capability of GA in searching for Stackelberg equilibrium. Regardless what values α and β are set, GA can always effectively find Stackelberg equilibrium from infinite strategy spaces of the two players.

B. Stackelberg Equilibrium from Real Spam Data

The real spam data set consists of spam emails obtained from [10]. It is collected from an anonymous individual's email-box of about fifteen months' time. Not surprisingly, the mean and standard deviation of the attributes vary from one month to another. We assume the changes of mean and standard deviation are motivated by adversarial transformations applied on the original attributes.

We compute the equilibrium transformation (i.e. the best adversarial strategy) by GA from the spam training data which consists of 500 spams and 500 legitimate emails. Fig. 3 shows the adversarial gain from the best transformation found in each iteration, together with the classifier's error rate. Without losing generality, the values of α, β and the maximum number of iterations are set as the same as experiments on synthetic data. The equilibrium searching progress is illustrated in Fig. 3: the algorithm converges with the spammer's payoff at about 0.6726 with a significant increase of false negative rate from 0.5736 to around 0.6732. Similar to the scenarios in Table I, the value of FNR is generally higher than FPR, due to the non-zero penalty (β = 0.01) to the relocation of classification boundaries.

C. Upper Bound for Adversarial Gain

Since the classifier built on training data set is the initial stage of the game theoretical model, the data distributions of each month from test set can be treated as transformed distributions from the training data. Compared to the intrusion spams in the training set, the adversarial gain introduced by the transformed instances of each month may either increase or decrease from the original gain (the dashed line in Fig. 4). This is due to the fact that the spammers in this concept drift scenario do not have our rational playing strategy and thus transform their instances randomly.

The solid line on top of Fig. 4 indicate the adversarial gain given by the equilibrium transformations from generic algorithm. Since the equilibrium transformation gives the highest adversarial gain, the gain values of each month from the test data are all below the equilibrium lines. So *the gain of all possible adversaries is upper bounded by the equilibrium adversarial gain*.

VII. Conclusions and Future Research

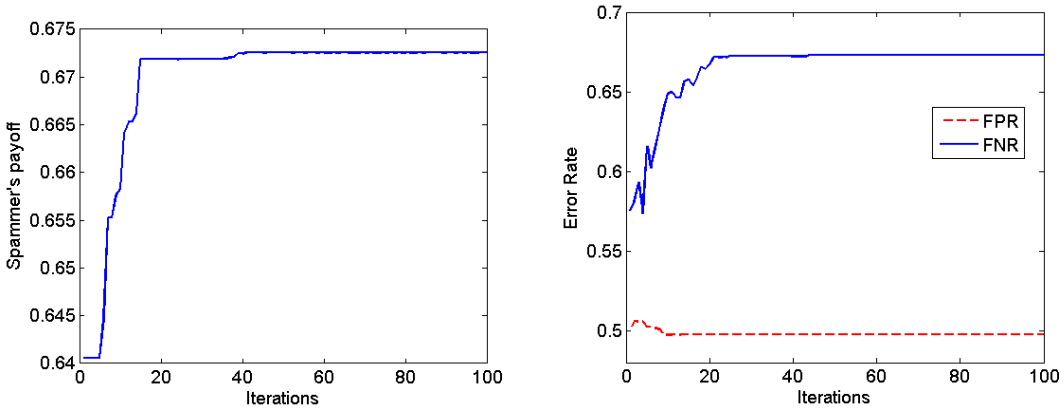
The race between adversary and the data miner can be interpreted in a game theoretical framework. Our empirical results show that two players can reach Stackelberg equilibrium when they are playing their best strategy at the same time. In future research we will focus on designing novel classifiers with robust retraining strategy against the adversary's transformation on intrusion data.

Acknowledgment

This work is sponsored by Australia Research Council Discovery Grant (DP0881537).

Table I: Variations of Stackelberg Equilibrium with different combinations of α and β when $\mu_1 = 1$, $\mu_2 = 10$ and $\sigma = 2$

		u	w	J^L	J^F	ErrRate	FPR	FNR
$\alpha = 0$	$\beta = 0$	9	10	0.5	0	50.00%	50.00%	50.00%
	$\beta = 0.01$	9	5.5017	0.9877	0	50.00%	1.22%	98.77%
	$\beta = 0.1$	9	5.5	0.9878	0	50.00%	1.22%	98.78%
$\alpha = 0.01$	$\beta = 0$	0.8229	5.9114	0.0148	1.9181	2.05%	2.05%	2.05%
	$\beta = 0.01$	1.1598	5.9937	0.0175	1.8972	2.51%	2.26%	2.76%
	$\beta = 0.1$	7.8368	5.9463	0.4652	0.0858	47.36%	2.13%	92.58%
$\alpha = 0.05$	$\beta = 0$	0.3943	5.6971	0.0138	1.9371	1.57%	1.57%	1.57%
	$\beta = 0.01$	0.5069	5.7069	0.0147	1.9320	1.69%	1.59%	1.79%
	$\beta = 0.1$	1.6520	5.8346	0.0217	1.8399	3.72%	1.86%	5.58%
$\alpha = 0.1$	$\beta = 0$	0.1749	5.5875	0.0129	1.9453	1.37%	1.37%	1.37%
	$\beta = 0.01$	0.2179	5.5869	0.0133	1.9437	1.41%	1.37%	1.45%
	$\beta = 0.1$	0.3752	5.5556	0.0148	1.9368	1.57%	1.31%	1.83%



(a) Highest spammer's payoff (adversarial gain) of each iteration.

(b) Error rate of each iteration.

Figure 3: Searching process of Stackelberg equilibrium on real spam data sets by GA. The stationary spammer's payoff and error rate after around 40th iteration indicate the Stackelberg equilibrium is achieved. In error rate observations, FNR is usually higher than FPR due to non-zero penalties to the relocation of classification boundaries.

REFERENCES

[1] M. Barreno, P. Bartlett, F. Chi, A. Joseph, B. Nelson, B. Rubinstein, U. Saini, and J. Tygar, "Open problems in the security of learning," in *Proceedings of the 1st ACM workshop on Workshop on AISec*. ACM New York, NY, USA, 2008, pp. 19–26.

[2] M. Barreno, B. Nelson, A. D. Joseph, and D. Tygar, "The security of machine learning," *Machine Learning Journal (MLJ) Special Issue on Machine Learning in Adversarial Environments*, 2008.

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

[4] D. Lowd and C. Meek, "Adversarial learning," in *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. New York, NY, USA: ACM, 2005, pp. 641–647.

[5] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. P. Rubinstein, U. Saini, C. Sutton, J. D. Tygar, and K. Xia, "Exploiting machine learning to subvert your spam filter," in *LEET'08: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*. Berkeley, CA, USA: USENIX Association, 2008, pp. 1–9.

[6] B. Nelson and A. Joseph, "Bounding an attacks complexity for a simple learning model," in *Proceedings of the First Workshop on Tackling Computer Systems Problems with Machine Learning Techniques (SysML)*, 2006.

[7] M. Kantarcioglu, B. Xi, and C. Clifton, "Classifier Evaluation and Attribute Selection against Active Adversaries," vol. No.09-01, 2009.

[8] W. Liu and S. Chawla, "A Game Theoretical Model for Adversarial Learning," *Technical Report, The University of Sydney*, no. TR 642, September 2009.

[9] S. Kullback and R. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, pp. 79–86, 1951.

[10] S. J. Delany, P. Cunningham, A. Tsymbal, and L. Coyle, "A case-based technique for tracking concept drift in spam filtering," *Knowledge-Based Systems*, vol. 18, no. 4–5, pp. 187–195, 2005.