REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
08-05-2017	Final Report	27-Sep-2012 - 26-Sep-2015

4. TITLE AND SUBTITLE
Final Report: A Game Theoretic Framework for Adversarial Classification

5a. CONTRACT NUMBER
W911NF-12-1-0558

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHORS
Murat Kantarcioglu, Bowei Xi, Bhavani Thuraisingham

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
University of Texas at Dallas
800 West Campbell Road, AD15

Richardson, TX 75080 -3021

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
58345-CS.18

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
Many real world applications, ranging from spam filtering to intrusion detection, are facing malicious adversaries who actively transform the objects under their control to avoid detection. Unfortunately, traditional machine learning techniques are insufficient to handle such adversarial problems directly. Adversaries change the dynamics in standard settings where machine learning techniques are designed to excel. They adopt their attacks to deceive the machine learning models built using the past data. Therefore, data encountered at application time and data used at training time do not necessarily resemble each other. As a result, despite assurance of the contrary at the model

15. SUBJECT TERMS
adversarial machine learning, game theoretical models for adversarial machine learning, adversarial support vector machines

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
UU	UU	UU	UU		Murat Kantarcioglu
					19b. TELEPHONE NUMBER
					972-883-6616

Standard Form 298 (Rev 8/98)
Prescribed by ANSI Std. Z39.18

Report Title

Final Report: A Game Theoretic Framework for Adversarial Classification

ABSTRACT

Many real world applications, ranging from spam filtering to intrusion detection, are facing malicious adversaries who actively transform the objects under their control to avoid detection. Unfortunately, traditional machine learning techniques are insufficient to handle such adversarial problems directly. Adversaries change the dynamics in standard settings where machine learning techniques are designed to excel. They adopt their attacks to deceive the machine learning models built using the past data. Therefore, data encountered at application time and data used at training time do not necessarily resemble each other. As a result, despite assurance of the contrary at the model training time, the accuracy of the trained machine learning models start to derail and become unreliable.

In this project, we put together a holistic solution framework for learning problems where there are adversaries. As a starting point, we modeled the adversarial machine learning as a Stackelberg game, where the machine learning model builder and the adversary make sequential moves, and each player aims to maximize its own utility. Our game theoretic approach is to avoid constantly adapting to the adversary's actions. Instead, we focus on a learning algorithm's long term performance, i.e., its equilibrium performance. At an equilibrium, neither the defender nor the adversary has an incentive to change its action. Based on the learning algorithm's equilibrium performance, we are able to address many questions, such as predicting adversary's most likely actions, identifying which learning algorithms are least susceptible to attacks, and developing counter measures against potential adversaries. We continue to resolve the weaknesses of various learning algorithms by playing a zero-sum game between two opponents. Finally, we expand our problem to take into account multiple adversaries of various unknown types. We develop a nested Stackelberg game framework to find an optimal mixed strategy that provides consistent performance universally.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received Paper

TOTAL:

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received Paper

TOTAL: 1

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
05/08/2017	14	Yan Zhou, Murat Kantarcioglu. Adversarial Learning with Bayesian Hierarchical Mixtures of Experts, 2014 SIAM International Conference on Data Mining. 24-APR-14, Philadelphia, Pennsylvania, USA. : ,
05/08/2017	15	Richard Wartell, Yan Zhou, Kevin W. Hamlen, Murat Kantarcioglu. Shingled Graph Disassembly: Finding the Undecideable Path, Advances in Knowledge Discovery and Data Mining - 18th Pacific-Asia Conference, PAKDD 2014. 12-MAY-14, Tainan, Taiwan. : ,
05/08/2017	13	Yan Zhou, Murat Kantarcioglu, Bhavani Thuraisingham, Bowei Xi. Adversarial Support Vector Machine Learning, Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. 12-AUG-12, Beijing, China. : ,
05/08/2017	16	Yan Zhou, Murat Kantarcioglu. Modeling Adversarial Learning as Nested Stackelberg Games, Advances in Knowledge Discovery and Data Mining - 20th Pacific-Asia Conference, PAKDD 2016. 12-APR-16, Auckland, New Zealand. : ,
05/08/2017	17	Murat Kantarcioglu, Bowei Xi. Adversarial Data Mining: Big Data Meets Cyber Security, Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 24-OCT-16, Vienna, Austria. : ,
05/08/2017	12	Yan Zhou, Murat Kantarcioglu, Bhavani M. Thuraisingham. Self-Training with Selection-by-Rejection, 2012 IEEE 12th International Conference on Data Mining (ICDM). 10-DEC-12, Brussels, Belgium. : ,

TOTAL: **6**

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
05/08/2017	1.00	Yan Zhou, Murat Kantarcioglu, Bhavani Thuraisingham. Sparse Bayesian Adversarial Learning Using Relevance Vector Machine Ensembles, 2012 IEEE 12th International Conference on Data Mining. 10-DEC-12, Brussels, Belgium Belgium. : ,
05/08/2017	8.00	Murat Kantarcioglu, Bowei Xi. Adversarial Data Mining: A Game Theoretic Approach, Symposium on "Analysis Support to Decision Making in Cyber Defence and Security" (SAS-106). 10-JUN-14, Talinn, Estonia. : ,

TOTAL: **2**

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received	Paper

TOTAL:

Number of Manuscripts:

Books

Received		Book
05/08/2017	11.00	Bhavani M. Thuraisingham, Tyrone Cadenhead, Murat Kantarcioglu, Vaibhav Khadilkar. Secure Data Provenance and Inference Control with Semantic Web, Florida: Auerbach Publications, (08 2014)

TOTAL: **1**

Received	Book Chapter

TOTAL:

Patents Submitted

Patents Awarded

Awards

Murat Kantarcioglu, PAKDD 2016 Best Application Paper Award for "Modeling adversarial learning as nested stackelberg games"

Murat Kantarcioglu, Homer Warner Award (Best Paper), American Medical Informatics Association (AMIA) Annual Symposium, 2014

Murat Kantarcioglu, Distinguished Scientist, Association for Computing Machinery (ACM), (2016)

Murat Kantarcioglu, Senior Member, IEEE (2013)

Bowei Xi, A publication is top 5 most popular article on STAT in 2014.

Bhavani Thuraisingham, the SDPS 2012 Transformative Achievement Gold Medal for interdisciplinary research on integrating computer sciences with social sciences

Bhavani Thuraisingham, 2013 IBM Faculty Award in Cyber Security.

Bhavani Thuraisingham, Society for Information Reuse and Intregration (SIRI) Research Leadership 2014

Bhavani Thuraisingham, Erik Jonsson School of Engineering and Computer Science (ECS) Senior Faculty Research Award 2016

Graduate Students

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Post Doctorates

NAME PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Names of Faculty Supported

NAME PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Names of Under Graduate students supported

NAME PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: 0.00

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:...... 0.00

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):...... 0.00

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense 0.00

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:...... 0.00

Names of Personnel receiving masters degrees

NAME

Total Number:

Names of personnel receiving PHDs

NAME

Total Number:

Names of other research staff

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

See attachment.

Technology Transfer

We presented our work at the NATO S&T Symposium on ``Analysis Support to Decision Making in Cyber Defence and Security" (SAS- 106) in Talin, Estonia to transfer our research.

In addition, we start collaborating with ARL researchers on the topic and now currently working with ARL South researchers to transition some of our research to practice.

ARO Final Performance Report

Project Title:	A Game Theoretic Framework for Adversarial Classification
ARO Proposal Number:	58345-CS
Agreement Number:	W911NF-12-1-0558
Project Period:	9/27/2012 - 1/26/2015
Program Manager:	Dr. Cliff Wang
Principal Investigators:	Murat Kantarcioglu (University of Texas at Dallas (UT Dallas))
Co-Principal Investigators:	Bowei Xi (Purdue University) and Bhavani Thuraisingham (UT Dallas)

Contents

1 Statement of the Problem Studied

Many real world applications, ranging from spam filtering to intrusion detection, are facing malicious adversaries who actively transform the objects under their control to avoid detection. Unfortunately, traditional machine learning techniques are insufficient to handle such adversarial problems directly. Adversaries change the dynamics in standard settings where machine learning techniques are designed to excel. They adopt their attacks to deceive the machine learning models built using the past data. Therefore, data encountered at application time and data used at training time do not necessarily resemble each other. As a result, despite assurance of the contrary at the model training time, the accuracy of the trained machine learning models start to derail and become unreliable.

In this project, we put together a holistic solution framework for learning problems where there are adversaries. As a starting point, we modeled the adversarial machine learning as a Stackelberg game, where the machine learning model builder and the adversary make sequential moves, and each player aims to maximize its own utility. Our game theoretic approach is to avoid constantly adapting to the adversary's actions. Instead, we focus on a learning algorithm's long term performance, i.e., its equilibrium performance. At an equilibrium, neither the defender nor the adversary has an incentive to change its action. Based on the learning algorithm's equilibrium performance, we are able to address many questions, such as predicting adversary's most likely actions, identifying which learning algorithms are least susceptible to attacks, and developing counter measures against potential adversaries. Finally, we expanded our problem to take into account multiple adversaries of various unknown types. We developed a nested Stackelberg game framework to find an optimal mixed strategy that provides consistent performance universally.

Our game theoretic framework is very general and applies to many security applications. The research funded as a part of this grant has lead us to discover important results and insights. One important insight from our work is about how to select the right features for increasing the robustness of the machine learning algorithms [3]. Guided by a learning algorithm's equilibrium performance, we must jointly consider different aspects of a feature, including: 1) its modification cost, i.e., how expensive it is for an attacker to modify this feature that is used by the machine learning model; 2) its effectiveness, i.e., its power to differentiate different object classes such as malware vs benign software. Focusing on only one aspect of a feature leads to poor results. For example, for malware detection, we notice initially useful features such as signatures extracted from a binary executable could be easily modified, and become useless quickly in the near future. On the other hand, a hard-to-modify feature such as system calls could be useless if such system calls are also used by legitimate software. Our game theoretic framework can assist practitioners to jointly evaluate the features and select the right ones for their machine learning models.

Another important insight from our work is to consider different types of adversaries with different capabilities and goals. For example, focusing on unsophisticated attackers that can only use the existing tools is not enough. At the same time, assuming all the attackers are sophisticated state-funded attackers is not necessary and may even make it harder to catch the crude attackers. To address these challenges, we show how classifiers, each tailored for a specific type of attackers, can be optimally combined into a defensive system against different types of adversaries [7].

Besides the general game theoretic framework itself, the insights we gained from the framework can be used to directly construct robust machine learning techniques. For example, by leveraging the game theory inspired ideas, we have developed a robust support vector machine (SVM) technique that has overall good performance against various potential malicious attacks [8]. In our other work, we showed how to develop more robust relevance vector machines [10],and robust Bayesian hierarchical mixtures of experts [6].

2 Summary of the Most Important Results

Below, we provide overview of our major results. In section 2.1, we discuss our generic Stackelberg framework and how it can be used for feature selection. In section 2.2, we discuss our adversarial support machine model. In section 2.3, we discuss our robust relevance vector machine learning framework. In section 2.4, we discuss our robust Bayesian hierarchical mixtures of experts model learning. In section 2.5, we provide an overview of our award winning generic learning framework that is resistant against multiple types of adversaries. Finally, in section 2.7, we conclude by summarizing the major accomplishments of the team members during the project period.

2.1 Stackelberg Games and Feature Selection [3]

Our first work is guided by a game theoretic framework initially developed for understanding and reasoning about several adversarial classification applications. In our model, the adversarial classification scenario is formulated as a two class problem, where class one (π_g) is the "good" class and class two (π_b) is the "bad" class. Assume q attributes are measured from an object coming from either classes. We denote the vector of attributes by $\mathbf{x} = (x_1, x_2, \ldots, x_q)'$. Furthermore, we assume that the attributes of an object $\mathbf{x}$ follow different distributions for different classes. Let $f_i(\mathbf{x})$ be the probability density function of class π_i, $i = g$ or b. The overall population is formed by combining the two classes. Let p_i denote the proportion of class π_i in the overall population. Note $p_g + p_b = 1$. The distribution of the attributes $\mathbf{x}$ for the overall population can be considered as a mixture of the two distributions, with the density function written as $f(\mathbf{x}) = p_g f_g(\mathbf{x}) + p_b f_b(\mathbf{x})$.

We assume that the adversary can control the distribution of the "bad" class π_b (e.g., malware class). In other words, the adversary can modify the distribution by applying a transformation $\mathbf{T}$ to the attributes of an object $\mathbf{x}$ that belongs to π_b (e.g., by applying binary obfuscation techniques). Hence $f_b(\mathbf{x})$ is transformed into $f_b^{\mathbf{T}}(\mathbf{x})$. Each such transformation comes with a cost; the transformed object is less likely to benefit the adversary, although more likely to pass the classifier. When a "bad" object from π_b is mis-classified as a "good" object into π_g, it generates profit for the adversary. A transformed object from $f_b^{\mathbf{T}}(\mathbf{x})$ generates less profit than the original one. In our prior work, we assume that the values of p_g and p_b are not affected by transformation, meaning that the adversary transforms the distribution of π_b, but in a short time period cannot significantly increase or decrease the proportion of "bad" objects. However, for Bayesian classifier p_b and p_g are just parameters that define the classification regions. They can be transformed by the adversary and be adjusted in Bayesian classifier to optimize the classification rule by the learner. Here we examine the case where a rational adversary and a rational learner play the following game: 1) Given the initial distribution and density $f(\mathbf{x})$, the adversary chooses a transformation $\mathbf{T}$ from the set of all feasible transformations $\mathcal{S}$, the strategy space; 2) After observing the transformation $\mathbf{T}$, learner creates a classification rule h.

Consider the case where learner wants to minimize its mis-classification cost. Given transformation $\mathbf{T}$ and the associated $f_b^{\mathbf{T}}(\mathbf{x})$, the learner responds with a classification rule $h(\mathbf{x})$. Let $L(h, i)$ be the region where the objects are classified as π_i based on $h(\mathbf{x})$ for $i = g$ or b. Let the expected cost of mis-classification be $C(\mathbf{T}, h)$, which is always positive. Define the payoff function of the learner as $u_g(\mathbf{T}, h) = -C(\mathbf{T}, h)$. In order to maximize its payoff u_g, the learner needs to minimize the mis-classification cost $C(\mathbf{T}, h)$.

Note that adversary only profits from the "bad" objects that are classified as "good". Also note that transformation may change the adversary's profit of an object that successfully passes detection. Define $g(\mathbf{T}, \mathbf{x})$ as the profit function for a "bad" object $\mathbf{x}$ being classified as a "good" one, after transformation $\mathbf{T}$ being applied. Define the adversary's payoff function of a transformation $\mathbf{T}$ given a classification rule h as the following:

$$u_b(\mathbf{T}, h) = \int_{L(h,g)} g(\mathbf{T}, \mathbf{x}) f_b^{\mathbf{T}}(\mathbf{x}) \, d\mathbf{x}.$$

Within the vast literature of game theory, the *extensive game* provides a suitable framework for us to model the sequential structure of adversary and learner's actions. Specifically, the *two-player Stackelberg game* suits our need. In a Stackelberg game, one of the two players (Leader) chooses an action a_b first and the second player (Follower), after observing the action of the leader, chooses an action a_g. The game ends with payoffs to each player based on their utility functions and actions. In our model, we assume all players act rationally throughout the game. For the Stackelberg game, this implies that the follower responds with the action a_g that maximizes its utility u_g given the action a_b of the leader. The assumption of acting rationally at every stage of the game eliminates the Nash equilibria with non-credible threats and creates an equilibrium called the *subgame perfect equilibrium*. In this project, we showed that such a model could be used to choose a set of features that balance between modification-cost and classification-effectiveness by examining the equilibrium performance of the above game theoretic model.

2.2 Adversarial Support Vector Machine Learning [8]

In our continued work, we developed an adversarial learning framework in which we model the adversary's attack strategies and developed robust learning models to mitigate the attacks. 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 rational adversary would deploy under penalties. We developed optimal SVM learning strategies against the two attack models. We demonstrated that it is possible to develop a much more resilient SVM learning model under loose assumptions about the data corruption models (e.g., loose assumption on attacker transformation $\mathbf{T}$).

Problem Definition:
Let $\{(x_i, y_i) \in (\mathcal{X}, \mathcal{Y})\}_{i=1}^n$ denote a sample set, 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 $x_i|_{y_i=1}$.

2.2.1 Adversarial Attack Models

We construct 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 based on 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.

Free-Range Attack The only knowledge the adversary needs is the valid range of each feature. Let $x_{\cdot j}^{max}$ and $x_{\cdot j}^{min}$ 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 $x_{\cdot j}^{max}$ and $x_{\cdot j}^{min}$ 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:

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

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.

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. More realistically, the adversary can set x_i^t to be the estimated centroid of innocuous 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

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

The *restrained-attack* model is given as follows:

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

where $C_\delta \in [0,1]$ is a constant modeling the loss of malicious utility as a result of the movement δ_{ij}, and $C_\xi \in [0,1]$ is a discount factor directly used to model the severeness of attacks.

2.2.2 Adversarial SVM Learning

We build an adversarial support vector machine model (AD-SVM) against each of the two attack models. 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.

AD-SVM against Free-range Attack Model Given the hinge loss model as follows:

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

where δ_i is the displacement vector for x_i, $\preceq$ and $\succeq$ denote component-wise inequality, following the standard SVM risk formulation and further reducing the bilinear problem to its 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, we have the following SVM risk minimization problem:

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

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:

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

where δ_i denotes the modification to x_i, $\preceq$ is component-wise inequality, and $\circ$ denotes component-wise operations. We solve the following SVM risk minimization problem:

$$
\begin{aligned}
\underset{w,b,\xi_i,t_i,u_i,v_i}{\arg\min} \quad & \tfrac{1}{2}||w||^2 + C\textstyle\sum_i \xi_i \\
s.t. \quad & \xi_i \geq 0 \\
& \xi_i \geq 1 - y_i\cdot(w\cdot x_i+b)+t_i \\
& t_i \geq \textstyle\sum_j e_{ij}u_{ij} \\
& (-u_i+v_i)\circ(x_i^t-x_i) = \tfrac{1}{2}(1+y_i)w \\
& u_i \succeq 0 \\
& v_i \succeq 0
\end{aligned}
$$

where

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

2.2.3 Overview of the Experimental Result

In our experiments, we investigate the robustness of the AD-SVM models as we increase the severeness of the attacks. 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. Table 1 and Table 2 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.

2.3 Sparse Bayesian Adversarial Learning Using Relevance Vector Machine Ensembles [10]

In this part of the project, we explore a new proactive defense strategy in which at training time we search for the most effective direction for the adversary to move data in the feature space to influence the classifier. Once

Table 1: 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 2: 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_δ = 0.9	0.874	**0.821**	**0.766**	**0.720**	0.579
	C_δ = 0.7	**0.888**	**0.860**	**0.821**	**0.776**	0.581
	C_δ = 0.5	0.874	**0.860**	**0.849**	**0.804**	0.586
	C_δ = 0.3	0.867	**0.855**	**0.845**	**0.809**	0.590
	C_δ = 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

we find such a direction, we can improve the classifier by countering these potential moves. The learning model we choose to implement this strategy is the *relevance vector machine*. Similar to the support vector machine method, the relevance vector machine (RVM) is a sparse linearly parameterized model. It is built on a Bayesian framework of the sparse model. Unlike the support vector machine in which a penalty term is introduced to avoid over-fitting the model parameters, the relevance vector machine model introduces a prior over the weights in the form of a set of hyperparameters, one associated independently with each weight. Very large values of the hyperparameters (corresponding to zero-weights) imply irrelevant inputs. Training data points associated with the remaining non-zero weights are referred to as *relevance vectors*. The relevance vector machine typically use much fewer kernel functions compared to the SVM.

We developed a sparse relevance vector machine ensemble for adversarial learning. The basic idea of this approach is to learn an individual kernel parameter η_i for each dimension d_i in the input space. The parameters are iteratively estimated from the data along with the weights and the hyperparameters associated with the weights. The kernel parameters are updated in each iteration so that the likelihood of the positive (malicious) data points are minimized. This essentially models adversarial attack as if the adversary were granted access to the internal states of the learning algorithm. Instead of using fixed kernel parameters, we search for kernel parameters that simulate worst-case attacks while the learning algorithm is updating the weights and the weight priors of a relevance vector machine. We learn M such models and combine them to form the final hypothesis.

2.3.1 Kernel Parameter Fitting

The RVM training process iteratively updates the weight vector w and the hyperparameter vector α. Imagine in each iteration the adversary has an opportunity to modify the training data, particularly the positive (malicious) training data, so that it could cross the decision boundary inferred in the current iteration. What would be the best strategy for the adversary to modify the data? If the adversary has the freedom to move each data point in his own favor, he would follow the directions that increase the likelihood of misclassifying a positive instance the greatest.

Kernel Parameter Vector Consider the RBF kernel

$$K(x_i, x_j) = \exp(-\eta \cdot ||x_i - x_j||^2)$$

where $\eta = (\eta_1, \ldots, \eta_d)$ is a vector of d parameters, and η_k is its k^{th} parameter preceding the squared distance $(x_{ik} - x_{jk})^2$ in the k^{th} input dimension. Normally, there is only one kernel parameter and its value is typically determined through cross-validations. We use individual kernel parameters so that we can model adversarial data modification in each dimension. For example, when the adversary modifies the k^{th} dimension such that $x_{ik} \approx x_{jk}$, the same effect can be achieved by having $\eta_k \approx 0$. Therefore, by adjusting the kernel parameter of the k^{th} dimension of the input, we could model adversarial attacks in both the input space and the feature space. We can then update the weight parameter and the corresponding hyperparameters to counter the attacks.

Attacks Minimizing the Log-Likelihood Assuming the adversary is only interested in disguising positive data [1], during RVM training we search for a kernel parameter vector η that renders the most effective attacks on positive training instances. With a given w and α, we update for all positive instances η in the direction that decrease $\mathcal{L}_+$—the log-likelihood of the posterior distribution $p(y|w, \alpha)$ given as follows:

$$p(y|w) = \prod_{i=1}^{N} g(h(x_i; w))^{y_i} [1 - g(h(x_i; w))]^{1-y_i} \tag{1}$$

where $g(t)$ is the sigmoid function $g(t) = 1/(1 + e^{-t})$ applied to t. Taking the logarithm of both sides of Equation (1), we have:

$$log(p(t|w)) = \sum_{i=1}^{N} [y_i log(\sigma_i) + (1 - y_i)(1 - log(\sigma_i))] \tag{2}$$

where $\sigma_i = g(h(x_i; w))$ is the output of the sigmoid function. Let $\mathcal{L} = log(p(t|w)) = \mathcal{L}_+ + \mathcal{L}_-$, where

$$\mathcal{L}_+ = \sum_{i=1}^{N} y_i log(\sigma_i) \;\; \texttt{and} \;\; \mathcal{L}_- = \sum_{i=1}^{N} (1 - y_i)(1 - log(\sigma_i)).$$

The gradient of $\mathcal{L}$ given in (2) with respect to the η_k is:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \eta_k} &= \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\partial \mathcal{L}}{\partial K_{ij}} \frac{\partial K_{ij}}{\partial \eta_k} \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} (\frac{\partial \mathcal{L}_+}{\partial K_{ij}} + \frac{\partial \mathcal{L}_-}{\partial K_{ij}}) \frac{\partial K_{ij}}{\partial \eta_k}\end{aligned}$$

[1] This is a reasonable assumption since it is typically harder for adversaries to influence negative (legitimate) data.

where K_{ij} is the kernel function K applied to the i^{th} and j^{th} input x_i and x_j. To model attacks on the positive instances, we negate $\frac{\partial \mathcal{L}_+}{\partial K_{ij}}$, and use the following for a gradient-based local optimization over η:

$$\mathcal{G} = \sum_{i=1}^{N} \sum_{j=1}^{N} (-\frac{\partial \mathcal{L}_+}{\partial K_{ij}} + \frac{\partial \mathcal{L}_-}{\partial K_{ij}}) \frac{\partial K_{ij}}{\partial \eta_k}. \tag{3}$$

Working out each term, we have:

$$\begin{aligned} \frac{\partial \mathcal{L}_+}{\partial K_{ij}} &= y_i \cdot \frac{1}{\sigma_i} \cdot \frac{\partial \sigma_i}{\partial h} \cdot \frac{\partial h}{\partial K_{ij}} \\ &= y_i \cdot (1 - \sigma_i) \cdot w_j \end{aligned}$$

$$\begin{aligned} \frac{\partial \mathcal{L}_-}{\partial K_{ij}} &= (1 - y_i) \cdot \frac{-1}{1 - \sigma_i} \cdot \frac{\partial \sigma_i}{\partial h} \cdot \frac{\partial h}{\partial K_{ij}} \\ &= -(1 - y_i) \cdot \sigma_i \cdot w_j \end{aligned}$$

$$\frac{\partial K_{ij}}{\partial \eta_k} = -K_{ij} \cdot (x_{ik} - x_{jk})^2$$

Therefore,

$$\mathcal{G} = \sum_{i=1}^{N} \sum_{j=1}^{N} -(y_i - \sigma_i) \cdot w_j \cdot K_{ij} \cdot (x_{ik} - x_{jk})^2$$

which will be the basis for updating η in each iteration of training a relevance vector machine.

2.3.2 Overview of the Experimental Result

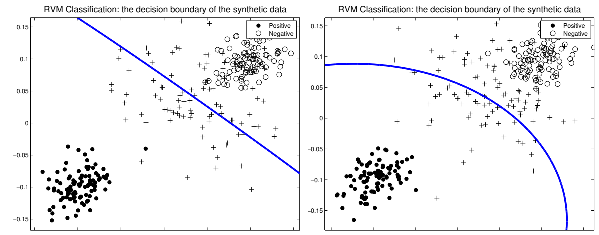
We model the attacks at classification time by moving positive test instances closer to randomly selected negative instances plus local random noise. Attacks on the test data are designed to challenge all the learning models at increasingly more difficult levels. The difficulty is controlled using the attack factor f_{attack}. More specifically,

$$x_{ij}^{+} = x_{ij}^{+} + f_{attack} \cdot (x_{ij}^{-} - x_{ij}^{+}) + \epsilon \tag{4}$$

where ϵ is local random noise. Notice $f_{attack} = 1$ models the worst case attacks where a positive data point is arbitrarily close to a negative one within the range of the random local noise. We compare four learning models: AD-RVM, RVM, SVM, and One-class SVM. On an artificial data set, we can clearly see how the adversarial RVM adjusts its decision boundary to counter adversarial attacks. The adjustment includes shifting and curving toward the negative data points as shown in Figure 1.

Table 4 shows the classification error rates of the four learning algorithms on the *webspam* [2] data set. The results are averaged over 10 random runs. As can be observed, adversarial-RVM is clearly superior to the other three models.

[2] http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/



(a) Decision Boundary Shifting (b) Decision Boundary Curving

Figure 1: Adjustment to decision boundary to take into account potential adversarial attacks. Solid lines in the plots illustrate the decision boundary.

Table 3: Classification errors of AD-RVM, RVM, SVM, and 1-class SVM on the webspam dataset. Best results are bolded.

	f_{attack}				
	0.1	0.3	0.5	0.7	0.9
AD-RVM	0.2426	**0.2926**	**0.3373**	**0.4945**	**0.5866**
RVM	**0.2355**	0.3169	0.4541	0.5560	0.5876
SVM	0.2725	0.4725	0.5604	0.6061	0.6061
One-class SVM	0.3155	0.5625	0.5945	0.6009	0.5997

2.4 Adversarial Learning with Bayesian Hierarchical Mixtures of Experts [6]

As adversaries become more sophisticated, their abilities of making versatile attacks grow. As a result, learning tools used in security applications are facing increasingly unpredictable and rapidly changing attacks. This calls for more flexible modeling techniques to handle ambiguities in the corrupted input. In this part of the project, we developed an adversarial learning framework using Bayesian hierarchical mixtures of experts (HME) as the baseline learning model. Our framework implements an optimal attack strategy that minimizes the likelihood of malicious data in each round of learning and a divide-and-conquer learning model that counters this type of adversarial attack. The learning process resembles the two-sided arms race by interactively manipulating data against the classifier.

The hierarchical mixtures-of-experts is a tree-structured probabilistic learning model. Unlike standard decision trees such as ID3, HME provides a soft split of data in the input feature space, allowing data to lie in multiple nested regions. The learning task is therefore divided into a set of overlapping sub-tasks of smaller sizes that are solved by components of the mixtures. The internal nodes are referred to as *gating networks* that score the competence of the experts located at the terminal nodes, for each input. Both internal and terminal nodes are input-sensitive predictors. When the adversary modifies the input vector of a data point, the outputs of both gating networks and expert networks are affected. By corrupting the input, the adversary can either poison the solutions of sub-tasks defined on soft partitions of the input or divert data away from the most probable path it is generated.

2.4.1 Robust Learning with Sparse Bayesian Hierarchical Mixtures of Experts

We consider the following adversarial learning problem in which an adversary alters malicious data to evade detection at test time. Here the traditional assumption that training data and test data follow identical distributions is violated.

Problem Definition:
Train a robust HME classifier C given $\{(x_i, y_i) \in (\mathcal{X}, \mathcal{Y})\}_{i=1}^N$ where $\mathcal{X} \subseteq \mathbb{R}^d$, $y_i \in \{-1, 1\}$ and there exists an adversary A at test time that transforms a malicious data point $x|_{y=1}$ to a (likely) legitimate one by adding a displacement vector Δx to $x|_{y=1}$.

Attacking Expert Networks We use the sparse Bayesian learning method with Gaussian kernels to train the expert networks. For regression the marginal likelihood of the experts is:

$$L_p(\boldsymbol{\alpha}) = -\frac{1}{2}[\log|\boldsymbol{D}| + y^T \boldsymbol{D}^{-1} y]$$

where $\boldsymbol{D} = \sigma^2 \boldsymbol{I} + \boldsymbol{\phi} \boldsymbol{A}^{-1} \boldsymbol{\phi}^T$. The gradient of the likelihood $L_p(\boldsymbol{\alpha})$ with respect to the k^{th} kernel parameter η_k is:

$$\frac{\partial L_p}{\partial \eta_k} = \sum_{i=1}^N \sum_{j=1}^N \frac{\partial L_p}{\partial \phi_{ij}} \frac{\partial \phi_{ij}}{\partial \eta_k},$$

where

$$\begin{aligned} \frac{\partial L_p}{\partial \phi_{ij}} &= -\frac{1}{2}[(2\boldsymbol{A}^{-1}\boldsymbol{\phi}^T\boldsymbol{D}^{-1})^T - 2\boldsymbol{D}^{-1}yy^T\boldsymbol{D}^{-1}\phi A^{-1}] \\ &= [\boldsymbol{D}^{-1}yy^T\boldsymbol{D}^{-1} - \boldsymbol{D}^{-1}]\boldsymbol{\phi}\boldsymbol{A}^{-1} \\ \frac{\partial \phi_{ij}}{\partial \eta_k} &= -\phi_{ij}(x_{ik} - x_{jk})^2 \end{aligned}$$

For binary classification with logistic sigmoid output, the likelihood of the expert is:

$$L_p(\boldsymbol{\alpha}) = \sum_{i=1}^N (y_i \log(\sigma_i) + (1 - y_i)(1 - \log(\sigma_i)))$$

where σ_i is the logistic sigmoid output given input $\boldsymbol{x}$. The gradient of $L_p(\boldsymbol{\alpha})$ with respect to η_k is:

$$\frac{\partial L_p}{\partial \eta_k} = -\sum_{i=1}^N \sum_{j=1}^N (y_i - \sigma_i) \cdot w_j \cdot \phi_{ij} \cdot (x_{ik} - x_{jk})^2$$

With the gradient $\partial L_p / \partial \eta_k$, our adversarial learning proceeds as we search for

$$\boldsymbol{\theta}(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\eta}) = \arg\max_{\boldsymbol{\alpha}, \boldsymbol{w}}(L_p^- + \arg\min_{\boldsymbol{\eta}}(L_p^+ + \ell_s)) \tag{5}$$

where $\boldsymbol{\theta}$ includes the learning model (the expert parameters $\boldsymbol{w}, \boldsymbol{\alpha}$) and the attack model (the kernel parameter $\boldsymbol{\eta}$), and $\ell_s = \sum_{i=1}^N (y_i - \sum_{m=1}^M g_m^{(i)} P_m(y_i))^2$ is the square loss and

$$\frac{\partial \ell_s}{\partial \eta_k} = -2 \cdot \sum_{i=1}^N \sum_{m=1}^M \sum_{j=1}^N \delta_i g_m^{(i)} (1 - g_m^{(i)}) v_{mj} \phi_{ij} (x_{ik} - x_{jk})^2$$

where $\delta_i = y_i - \sum_{j=1}^M g_m^{(i)} P_j(y_i)$.

The learning process is best understood as an arms race between the expert and the adversary: given expert parameters $(\boldsymbol{w}, \boldsymbol{\alpha})$, the adversary finds an $\boldsymbol{\eta}$ that minimizes the likelihood of the malicious data points, referred to as positive ('+') data points in the input. Note that in the minimization term in Equation (5) the adversary also attempts to minimize the square loss of the output. This may sound counter intuitive since minimizing training loss is not to the best interest of the adversary. A greedy adversary would attempt to maximize the loss of all malicious points. However, a simple validation on the training set would disclose the adversary's attempts. Therefore, the adversary's objective is to minimize the likelihood of malicious data and keep the attacks stealthy by maintaining minimum losses during training.

Attacking Gating Networks We use separate kernel parameters to control the input to the gating functions. The log-likelihood of the gating function is:

$$L_g(\boldsymbol{v}) = \sum_{i=1}^{N} \sum_{m=1}^{M} h_m^{(i)} \log g_m^{(i)} \tag{6}$$

Rewrite Equation (6) as:

$$L_g(\boldsymbol{v}) = \sum_{i=1}^{N} \sum_{m=1}^{M} (h_m^{(i)} \boldsymbol{v_m}^T \boldsymbol{\phi}_i - \log \sum_{m=1}^{M} \exp(\boldsymbol{v_m}^T \boldsymbol{\phi}_i))$$

where h_m is the posterior and defined as:

$$h_m = \frac{g_m p_m(y)}{\sum_{k=1}^{M} g_k p_k(y)},$$

and h_m is estimated in the E-step in the Bayesian EM learning algorithm. We use the Gaussian kernel to compute the basis function:

$$\phi_{ij} = \exp(-\sum_{k=1}^{d} \eta_k (x_{ik} - x_{jk})^2)$$

where d is the number of dimensions in the input space. The gradient of the likelihood L_g with respect to η_k is:

$$\frac{\partial L_g}{\partial \eta_k} = -\sum_i \sum_m \sum_j (h_m^{(i)} - g_m^{(i)}) v_{mj} \phi_{ij} (x_{ik} - x_{jk})^2.$$

Learning proceeds as iterative re-estimation of: (1) $\boldsymbol{v}$ that maximizes L_g given $\boldsymbol{\eta}$, and (2) $\boldsymbol{\eta}$ that minimizes L_g^+ given $\boldsymbol{v}$ until the algorithm convergences.

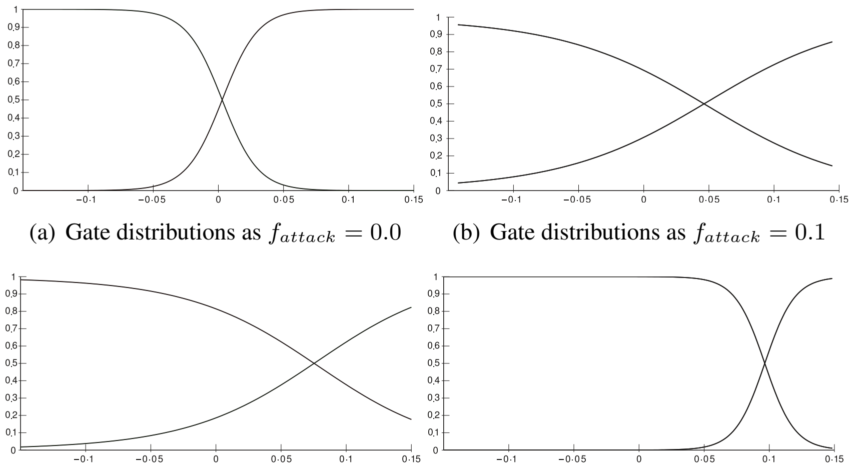
2.4.2 Overview of Experimental Results

We compare our adversarial HME learning algorithm to the following algorithms: the standard hierarchical mixtures of experts (HME), relevance vector machine (RVM) and its adversarial learning counterpart (AD-RVM), support vector machine (SVM) and its one-class learning counterpart (1-class SVM). We use a single level HME with two expert networks in our experiments. In order for apples-to-apples comparison, we repeat the experiments reported in [10] on one artificial data set and two real data sets. In these settings, the training data is clean, while the test datasets are corrupted by adversarial attacks modeled at increasingly intense levels. The intensity of attacks is controlled by the attack factor f_{attack} as follows:

$$\boldsymbol{x}^+ = \boldsymbol{x}^+ + f_{attack} \cdot (\boldsymbol{x}^- - \boldsymbol{x}^+) + \epsilon \tag{7}$$

where ϵ is local random noise, $\boldsymbol{x}^+$ and $\boldsymbol{x}^-$ are a positive data point and a random negative data point in the test set. As f_{attack} increases from 0 to 1 the intensity of attacks grows from none to the extreme where a malicious data point can be arbitrarily close to a legitimate data point, within a range of small random local noise. We compare six learning models: AD-HME, HME, AD-RVM, RVM, SVM, One-class SVM, and all results reported are averaged over 10 random runs.

AD-HME (gate) in general outperforms all the other five learning algorithms. Note that gating functions rank the competence of experts in classifying a data point. On the artificial data set, we can illustrate how AD-HME (gate) adaptively selects the expert that is most likely to generate the data point as shown in Figure 2 . Table 4 shows the error rates of the six algorithms as the strength of attacks increases on the *webspam* data set. The AD-HME algorithms were superior to others in all cases. Their superiority is also attributed to the baseline HME algorithm that significantly outperformed SVM and RVM. Nevertheless, the AD-HME algorithms consistently outperformed the baseline HME algorithm in all cases.



(a) Gate distributions as $f_{attack} = 0.0$ (b) Gate distributions as $f_{attack} = 0.1$

(c) Gate distributions as $f_{attack} = 0.5$ (d) Gate distributions as $f_{attack} = 0.9$

Figure 2: Gate distributions on the test dataset as attacks intensify as $f_{attack} = 0.0 \rightarrow 0.1 \rightarrow 0.5 \rightarrow 0.9$. The x-axis is the input to the gating functions and the y-axis is the posterior of each expert (approximating "+" and "-" data respectively).

Table 4: Classification error rates of HME, AD-HMEs, RVM, AD-RVM, SVM, and one-class SVM on the webspam dataset. Attacks are generated with $f_{attack} = 0.1, 0.3, 0.5, 0.7, 0.9$. The best results are bolded.

	f_{attack}				
	0.1	0.3	0.5	0.7	0.9
HME	0.1323 ± 0.0076	0.1566 ± 0.0206	0.2748 ± 0.0477	0.4360 ± 0.0522	0.5413 ± 0.0118
AD-HME$^{(exp)}$	0.1359 ± 0.0157	0.1550 ± 0.0253	0.2394 ± 0.0474	**0.4253 ± 0.0331**	0.5409 ± 0.0084
AD-HME$^{(gate)}$	**0.1276 ± 0.0089**	**0.1423 ± 0.0330**	**0.2383 ± 0.0422**	0.4298 ± 0.0346	**0.5353 ± 0.0139**
AD-HME$^{(exp+gate)}$	0.1302 ± 0.0091	0.1540 ± 0.0130	0.2534 ± 0.0441	0.4387 ± 0.0463	0.5401 ± 0.0115
RVM	0.2355 ± 0.0542	0.3169 ± 0.0512	0.4541 ± 0.0761	0.5560 ± 0.0731	0.5876 ± 0.0869
AD-RVM	0.2426 ± 0.0276	0.2926 ± 0.0565	0.3373 ± 0.0460	0.4945 ± 0.0149	0.5866 ± 0.0032
SVM	0.2725 ± 0.0383	0.4725 ± 0.0773	0.5604 ± 0.1232	0.6061 ± 0.1002	0.6061 ± 0.0874
One-class SVM	0.3155 ± 0.0040	0.5625 ± 0.0034	0.5945 ± 0.0041	0.6009 ± 0.0039	0.5997 ± 0.0053

2.5 Modeling Adversarial Learning as Nested Stackelberg Games [7]

So far we have only considered adversarial learning problems in which there is only a single type of adversary. In practice, a learner often has to face multiple types of adversaries that may employ different attack tactics. In this part of the project, we tackle the challenges of multiple types of adversaries with a nested Stackelberg game framework. The framework handles both data corruption and unknown types of adversaries. It consists of a set of *single leader single follower* (SLSF) Stackelberg games and a *single leader multiple followers* (SLMF) Bayesian Stackelberg game. We first solve a SLSF Stackelberg game for each adversary type. This level of Stackelberg game takes into consideration that training and test data are not necessarily identically distributed in practice. Given the learner's learning model, the adversary responds to the learner's strategy by optimally transforming data to maximize the learner's predictive error. The Stackelberg equilibrium solution consists of optimal learning parameters for the learner and data transformations for the adversary. The optimal solutions will be used as pure strategies in the Bayesian Stackelberg game. The Bayesian Stackelberg game consists of one learner and multiple adversaries of various types. When facing adversaries of multiple types, instead of settling on one learning model by playing a pure strategy, it is more practical for the learner to play a mixed strategy consisting of a set of learning models with assigned probabilities. The optimal solution to the Bayesian Stackelberg game introduces randomness to the solution, and hence increases the difficulty of attacking the underlying learning models via reverse engineering.

2.5.1 Nested Bayesian Stackelberg Games

We first develop strategies to construct component SLSF learning models given adversary types, and then solve the SLMF Stackelberg game with the component SLSF models to counter adversaries of various types.

A Single Leader Single Follower Stackelberg Game Each component learning model in our framework is obtained by solving a Stackelberg game between the learner and the adversary. The learner first commits to its strategy that is observable to the adversary and the adversary plays its optimal strategy to maximize the learner's loss while minimizing its own loss. Therefore, the adversarial learning problem of this *single leader single follower* (SLSF) game is:

$$\begin{aligned} &\arg\min_{w^*}\arg\max_{\delta_x^*} && L_\ell(w, x, \delta_x) \\ &s.t. && \delta_x^* \in \arg\min_{\delta_x} L_f(w, x, \delta_x) \end{aligned}$$

where L_ℓ is the leader's loss:

$$L_\ell = \sum_{i=1}^{n} c_{\ell,i} \cdot \ell_\ell(\hat{y}_i, y_i) + \lambda_\ell ||w||^2 \tag{8}$$

and L_f is the follower's loss where the second term penalizes for the L_2 norm of data transformation:

$$L_f = \sum_{i=1}^{n} c_{f,i} \cdot \ell_f(\hat{y}_i, y_i) + \lambda_f \sum_{i=1}^{n} ||\phi(x_i) - \phi(f_t(x_i, w))||^2. \tag{9}$$

λ_ℓ, λ_f, c_ℓ, and c_f are the weights of the penalty terms and the costs of data transformation. ℓ_ℓ and ℓ_f are the classification loss functions of the leader and the follower. A Stackelberg equilibrium solution exists if the adversary's loss is convex and continuously differentiable.

A Single Leader Multi-followers Stackelberg Game In a *single leader multiple followers* (SLMF) game, the leader makes its optimal decision prior to the decisions of multiple followers. The Stackelberg game played by the leader is:

$$\begin{aligned} \min_{x,y^*} \quad & F(x, y^*) \\ s.t. \quad & G(x, y^*) \leq 0 \\ & H(x, y^*) = 0 \end{aligned}$$

where F is the leader's objective function, constrained by G and H; x is the leader's decision and y^* is in the set of the optimal solutions of the lower level problem:

$$y^* \in \left\{ \begin{array}{lll} \arg\min_{y_i} & f_i(x, y_i) & \\ s.t. & g_i(x, y_i) \leq 0 & \\ & h_i(x, y_i) = 0 & \end{array} \right\} \forall i = 1, \ldots, m$$

where m is the number of followers, f_i is the i^{th} follower's objective function constrained by g_i and h_i. For the sake of simplicity, we assume the followers are not competing among themselves. This is usually a valid assumption in practice since adversaries rarely affect each other through their actions. In a Bayesian Stackelberg game, the followers may have many different types and the leader does not know exactly the types of adversaries it may face when solving its optimization problem. However, the distribution of the types of adversaries is known or can be inferred from past experience. The followers' strategies and payoffs are determined by the followers' types. The followers play their optimal responses to maximize the payoffs given the leader's strategy. The Stackelberg equilibrium includes an optimal mixed strategy of the learner and corresponding optimal strategies of the followers.

Problem Definition:
Given the payoff matrices R^ℓ and R^f of the leader and the m followers of n different types, find the leader's optimal mixed strategy given that all followers know the leader's strategy when optimizing their rewards. The leader's pure strategies consist of a set of generalized linear learning models $\langle \phi(x), w \rangle$ and the followers' pure strategies include a set of vectors performing data transformation $x \rightarrow x + \Delta x$.

The defined Stackelberg game can be solved as a Mixed-Integer-Quadratic-Programming (MIQP) problem. For a game with a single leader and m followers with n possible types where the m followers are independent of each other and their actions have no impact on each other's decisions, we reduce the problem to solving m instances of the *single leader single follower* game.

2.5.2 Overview of the Experimental Results

In the experiments, we use three types of adversaries. The first type $Adversary^{*1}$ can modify both positive and negative data, and the second type $Adversary^{*2}$ is only allowed to modify positive data as normally seen in spam filtering. The third type of adversary $Adversary^{*3}$ can transform data freely in the given domain. The prior distribution of the three adversary types is randomly set. Let p be the probability that the adversary modifies negative data. Then for each negative instance x^- in the test set, with probability p, x^- is modified as follows:

$$x^- = x^- + f_a \cdot (x^+ - x^-) + \epsilon$$

where ϵ is local random noise, and x^+ is a random positive data point in the test set. The intensity of attacks is controlled by the attack factor $f_a \in (0, 1)$. The greater f_a is, the more aggressive the attacks are. Similarly, for each positive instance x^+ we modify x^+ as follows:

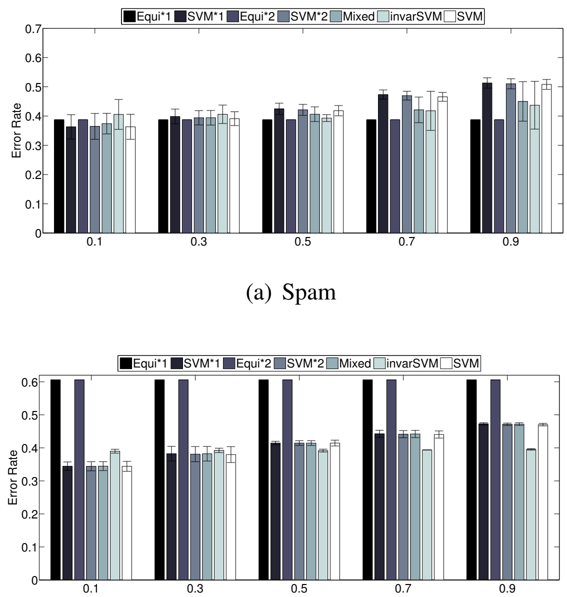
$$x^+ = x^+ + f_a \cdot (x^- - x^+) + \epsilon$$

where x^- is a random negative data point in the test set. For the third type of attack, x^+ and x^- can be freely transformed in the data domain as follows:

$$x^{\pm} = \begin{cases} min(x^{max}, x^{\pm} + f_a \cdot \delta \cdot (x^{max} - x^{min})) & \delta > 0 \\ max(x^{min}, x^{\pm} + f_a \cdot \delta \cdot (x^{max} - x^{min})) & \delta \leq 0 \end{cases}$$

where δ is randomly set and $\delta \in (-1, 1)$, x^{max} and x^{min} is the maximum and minimum values an instance can take. The learner's pure strategy set contains three learning models: 1.) Stackelberg equilibrium predictor Equi*; and 2.) two SVM models SVM^{*1} and SVM^{*2} trained on equilibrium data transformations. Note that SVM^{*1} and SVM^{*2} are optimal only when the SVM learner knows the adversary's strategy ahead of time. Therefore, SVM*s alone are not robust solutions to the adversarial learning problem. When solving the prediction games, we assume the adversary can modify data in both classes. SVM^{*1} and SVM^{*2} are trained on the two equilibrium data transformations when λ_f is set to 0.01 and 0.02. The two SVM models are essentially optimal strategies against the adversaries' equilibrium strategies. The learner will choose which learning model to play according to the probability distribution determined in the mixed strategy. The results are displayed as *Mixed* in the following sections. We also compare our results to the invariant SVM and the standard SVM methods. In all of our experiments, we modify the test sets to simulate the three types of adversaries.

We make the learning tasks more complicated by making the attack factor $f_a \in (0, 1)$ completely random under uniform distribution for each attacked sample in the test set. We assume the positive data is always modified by the adversary. In addition, we allow the probability of negative data being attacked to increase gradually from 0.1 to 0.9. The advantage of our mixed strategy is more obvious on these two datasets as illustrated in Figure 3. The equilibrium predictors Equi^{*1,2} are better than the SVM^{*1,2} predictors on the *spambase* data, but significantly worse on the *web spam* data. Our mixed strategy consistently outperforms SVM^{*1,2} on the *spambase* data, and outperforms Equi^{*1,2} on the *web spam* data.



(a) Spam

(b) Webspam

Figure 3: Classification error rates (with error bars) of $Equi^{*1}$, SVM^{*1}, $Equi^{*2}$, SVM^{*2}, $Mixed$, $invariant\ SVM$, and SVM on the `spambase` and `webspam` datasets.

2.6 Technology Transfer and External Outreach Activities

We presented our work [1] at the NATO S&T Symposium on "Analysis Support to Decision Making in Cyber Defense and Security" (SAS-106) in Talin, Estonia to disseminate our research findings.

In addition, we start collaborating with ARL researchers on the topic and now currently working with ARL South researchers to transition some of our research to practice.

2.7 Honors/Awards

- Murat Kantarcioglu:
  - PAKDD 2016 Best Application Paper Award for the paper [7](discussed in Section 2.5)
  - Homer Warner Award (Best Paper), American Medical Informatics Association (AMIA) Annual Symposium, 2014
  - Distinguished Scientist, Association for Computing Machinery (ACM), (2016)

  - Senior Member, IEEE (2013)

- Bhavani Thuraisingham:
  - the SDPS 2012 Transformative Achievement Gold Medal for interdisciplinary research on integrating computer sciences with social sciences
  - 2013 IBM Faculty Award in Cyber Security
  - Society for Information Reuse and Intregration (SIRI) Research Leadership
  - Erik Jonsson School of Engineering and Computer Science (ECS) Senior Faculty Research Award 2016

- Bowei Xi:
  - A publication is top 5 most popular article on STAT in 2014.

Publications Accepted/In-print Directly Funded By This Project

[1] M. Kantarcioglu and B. Xi. Adversarial data mining: A game theoretic approach. In *North Atlantic Treaty Organization (NATO) SAS-106 Symposium on Analysis Support to Decision Making in Cyber Defence, Estonia*, pages 1–11, 2014.

[2] M. Kantarcioglu and B. Xi. Adversarial data mining: Big data meets cyber security. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1866–1867, 2016.

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

[4] R. Wartell, Y. Zhou, K. W. Hamlen, and M. Kantarcioglu. Shingled graph disassembly: Finding the undecideable path. In *Advances in Knowledge Discovery and Data Mining - 18th Pacific-Asia Conference, PAKDD 2014, Tainan, Taiwan, May 13-16, 2014. Proceedings, Part I*, pages 273–285, 2014.

[5] B. Xi, K. M. Tan, and C. Liu. Logarithmic transformation-based gamma random number generators. *Journal of Statistical Software*, 55(1):1–17, 2013.

[6] Y. Zhou and M. Kantarcioglu. Adversarial learning with bayesian hierarchical mixtures of experts. In *SDM*, pages 929–937, 2014.

[7] Y. Zhou and M. Kantarcioglu. Modeling adversarial learning as nested stackelberg games. In *Advances in Knowledge Discovery and Data Mining - 20th Pacific-Asia Conference, PAKDD 2016, Auckland, New Zealand, April 19-22, 2016, Proceedings, Part II*, volume 9652 of *Lecture Notes in Computer Science*, pages 350–362. Springer, 2016.

[8] Y. Zhou, M. Kantarcioglu, B. Thuraisingham, and B. Xi. Adversarial support vector machine learning. In *SIGKDD*, pages 1059–1067. ACM, 2012.

[9] Y. Zhou, M. Kantarcioglu, and B. M. Thuraisingham. Self-training with selection-by-rejection. In *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, pages 795–803, 2012.

[10] Y. Zhou, M. Kantarcioglu, and B. M. Thuraisingham. Sparse bayesian adversarial learning using relevance vector machine ensembles. In *ICDM*, pages 1206–1211, 2012.