Securing Pervasive Systems Against Adversarial Machine Learning

Brent Lagesse*, Cody Burkard† and Julio Perez‡
Computing and Software Systems, University of Washington Bothell
Bothell, WA, USA
Email: *lagesse@uw.edu, †cburkard@uw.edu, ‡jperez99@uw.edu

***Abstract*—Applications and middleware in pervasive systems frequently rely on machine learning to provide adaptivity and customization that results in a seamless user experience despite operating in a dynamic environment. Machine learning algorithms have been shown to be vulnerable to covert, strategic attacks through the manipulation of training data. Machine learning algorithms in pervasive systems frequently train on data that could be manipulated by a malicious 3rd party. In this paper, we present our ongoing work to develop a security mechanism that is designed to work in the dynamic environments of pervasive computing as opposed to traditional security mechanisms that are designed for static environments. Furthermore, we present our modular testing framework that will be used to rapidly compare our work with other security mechanisms, applications and adversarial models.**

I. Introduction

Recent advances in the application of machine learning have been paramount in the advent of pervasive computing. Numerous pervasive systems utilize machine learning to provide necessary support for users and applications [1], [13], [4], [12], [9], [10]. As a result, these applications are left vulnerable to well-established attacks on machine learning algorithms [3]. Systems that adapt by training machine learning algorithms in the field are most vulnerable to these attacks as they will be training on data that has not been verified as trustworthy.

For example, mobile phones can be used to collect data and analyze it to relay health information to the user. These applications range in complexity from step counters to implantable medical devices. The use of automatic learning in health applications has created a more robust system that can give the user more precise information about their health [9]. Unfortunately this creates a risk of poisoning attacks that could result in incorrect and catastrophic decisions being made on behalf of the user. If these devices train on data sets sourced from untrustworthy sources then an attacker could be present to manipulate the data. Likewise, a malicious software can be installed in a mobile phone that will strategically skew sensor information collected to covertly perform these attacks.

Our work is designed to decrease the ability of attackers to target machine learning algorithms used in pervasive computing applications. Our work enables systems to more securely train on open data sources prior to deployment and to utilize adaptive machine learning techniques during deployment. The ability to use information that was collected after the initial training period can enable a more responsive system.

In the remainder of this paper we present two major components of our ongoing work. First, we present our game-theoretic system for more securely acquiring training data in dynamic environments. Then, we present our modularized system for rapidly comparing the performance of security mechanisms for machine learning applications against a variety of attacker models.

II. Background

There has been a variety of work on adversarial machine learning [3], [2], [6] that involves defeating machine learning by turning its adaptivity into a liability. The most common application areas are spam detection, intrusion detection, virus detection, and data mining, all of which gather training data online hence providing a clear path for an attack.

The attacks investigated focus on maximizing difficulties for machine learning systems according to some criteria within the constraints of the attacker's ability to modify the training data. For example, [2] describes how to maximally subvert a support vector machine with each new malicious exemplar, and [6] discusses a "boiling frog" approach where the primary criterion is avoiding detection and the secondary criterion is changing classifier performance.

While these attacks have thus far been focused on traditional computing infrastructure, pervasive systems are just as, if not more vulnerable to such attacks. One of the defining properties of pervasive systems is their ability to adapt to the world around them. This is often accomplished through the application of machine learning as the system persists. As a result, pervasive systems are left vulnerable to the aforementioned attacks.

III. Related Work

Existing security systems utilize passive techniques that focus on reducing negative impact of training on new data; however, the passive nature of these techniques makes it difficult for them to be directly applied to a dynamic world where features evolve.

Machine learning algorithms have typically included methods to be less influenced by outliers. Against strategic adversaries, the common approach to fortify machine learning against attacks is data sanitization, an approach where potentially malicious outliers are identified and eliminated from the training data. The Reject On Negative Impact (RONI)[11]

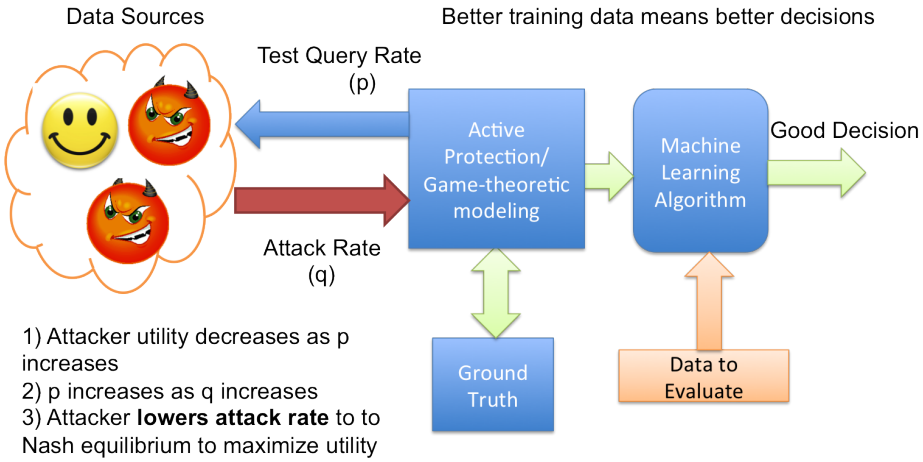

Fig. 1. System Overview

approach tests the effect of adding a new exemplar to a training set by executing a new trained classifier on a trusted test set, eliminating those exemplars that affect performance negatively. By eliminating training data that is different, this approach does not fit well in dynamic environments.

Another approach presented in research literature applies a game-theoretic analysis. [7] models the interaction between the attacker and defender as a Stackelberg game, solves for the equilibrium, and uses this analysis to select the best classification features to use. [5] performs a similar type of analysis modeling the interaction as a Nash game. Our system also uses a game-theoretical approach but focuses on solving the game created by our adaptive strategy rather than modeling the existing situation.

The previous approaches presented in research literature utilize techniques that passively accept input data for training sets and make decision based as to how best to use that data to prevent attacks. These approaches work well in static environments, but they do not work well in dynamic environments that are common in pervasive systems. Our approach differs significantly from the previous work in the field by changing the game that is played by attackers and defenders. This new approach actively detects attackers with poor attack strategies and deters attackers with more effective attack strategies. Further, our approach can be used in conjunction with previous work to improve the overall success of training a machine learning algorithm.

IV. Design

Some applications of machine learning algorithms actively acquire training data in untrusted environments. In such environments, attackers could poison training data and if a machine learning algorithm trained on that data, it would be likely to make errors in decisions at an increased rate. With many attacks, the errors would not just be random, but errors that are specifically targeted toward the malicious goals of an attacker. The technique we are using to prevent these attacks utilizes a game-theoretic approach similar to that designed in [8]. This approach has been shown to effectively motivate attackers not to supply corrupted resources and assist the user in avoiding data sources that do. We are leveraging this technique to create a security mechanism that actively deters attackers from producing poisoned data that would be used as training data in machine learning. Our approach utilizes a game-theoretic model and knowledge regarding ground truth to create a strategy of making requests, known as test queries, for data that can be used to determine if some of the features in the training data have been corrupted. The approximation of the Nash equilibrium is calculated from a model of the costs and benefits of the machine learning algorithm and estimates of those values from observed behaviors of the attacker in response to our test queries.

B_{ben}	User Benefit from New Training Data
B_{mal}	Attacker Benefit from Malicious Data Trained On
C_{vic}	Cost of Training on Malicious Data
C_{disc}	Cost of being Detected as Malicious

TABLE I
Notations used in this paper

$$p = \frac{B_{mal}}{C_{disc} + B_{mal}} \tag{1}$$

$$q = \frac{B_{ben}}{C_{vic} + B_{ben}} \tag{2}$$

The salient feature of our proposed mechanism is that an attacker must reduce its attack rate in order to optimize its utility. Our mechanism will use a game-theoretic approach to manipulate its test query rate, q, as shown in Figure 1 causing the attacker to adapt its attack rate to q and approximating the Nash equilibrium strategy that is described in Equations 1 and 2. This works because an unintelligent attacker that does not reduce its attack rate, p, will easily be detected, and an intelligent attacker (one that knows we are using this approach

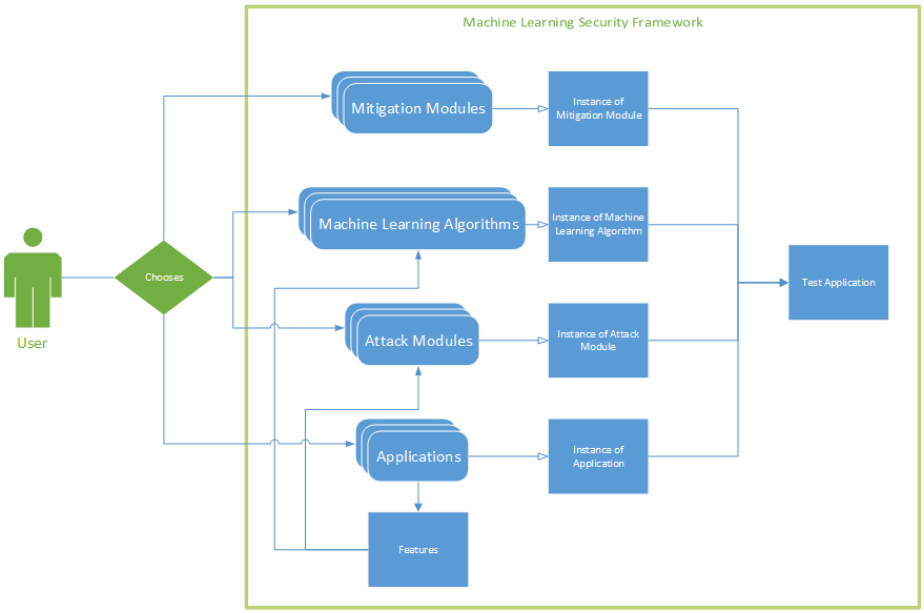

Fig. 2. Machine Learning Security Framework Architecture

and can adapt its attack rate to a Nash equilibrium when it either observes that its attacks are ineffective or it observes that it is no longer being queried for training data) must attack less often in order to maximize its own utility. This property holds despite the fact that we make the assumption that the attacker plays the game with perfect information while our mechanism plays with imperfect information. When this assumption does not hold, our mechanism can further improve the security of the machine learning algorithm as shown in [8] by making adaptations to the test query rate that exploit the attackers inefficient strategy.

This system models a training algorithm and data sources as a set of costs, benefits, and strategies. The terms in Equations 1 and 2 consist of these underlying costs and benefits. Training algorithms are modeled based on the costs of misclassification, sending requests, and generating/acquiring ground truth along with the benefits of the successfulness of their outputs and the strategy for how frequently to make exploratory queries. Malicious data sources are modeled based on their costs of providing data, the benefit they receive from a successful attack, and with strategies for how many pieces of information in a data source to corrupt. By building payoff equations based on these costs and benefits, our system estimates the utility of deploying various defense strategies against potential attacker strategies in order to minimize damage caused by an attacker as the system identifies potentially dangerous sources of training data.

V. Evaluation

In order to evaluate our approach, it is important to test a variety of scenarios to determine how effectively we can mitigate machine learning attacks in different environments. As a means to evaluate our work, we are developing the Machine Learning Security Framework (MLSF) as an open source platform aimed to quickly develop and test defenses against machine learning attacks. We modularize the important elements of a machine learning attack simulation in a way that allows these components to be interchanged with little effort, granting us the ability to quickly test our defensive approach in many scenarios and against comparable mechanisms.

A. MLSF Architecture

Our framework (shown in Figure 2) divides a machine learning attack simulation into five different components, including mitigation techniques, machine learning algorithms, attack modules, applications, and features.

- **Mitigation Techniques:** Defenses such as Arex[8] or RONI [11] that are being tested against a simulated attack; these modules should act independently of the rest of the framework, interacting only with the acquisition of training data that is passed to the machine learning algorithm.
- **Machine Learning Algorithms:** Pluggable modules that can be used to test a variety of machine learning implementations.
- **Attack Modules:** While largely dependent on the type of application being tested, these modules allow us to test a defensive measure against various attacks. At the core these implement reusable strategies derived from attacks such as [6], [2].
- **Applications:** Implementations of applications that use machine learning algorithms that we would like to test, such as a mobile health analytics application. These modules control what machine learning algorithms can be used and determine how an attack module can interact with the data being trained on.

- **Features:** Entirely dependent on the application being used, these modules extract and format training data from an application for a machine learning algorithm.

Use of the MLSF enables us to quickly test our defensive layer in a variety of pervasive computing scenarios. Positioning systems have increasingly employed machine learning techniques to support activity classification [9], [1], [4]. In an adversarial environment, these applications may be vulnerable to a variety of retraining attacks as new data is fed to the learning algorithm. If the attacker can actively impact the training data, attacks may be carried out to trick the algorithm into learning incorrect correlations in behavior. These attacks can have potentially dangerous outcomes in a pervasive computing environment, such as tricking a smart home into turning an oven on as a user leaves for work instead of preheating an oven as a user arrives at home.

With MLSF, we can test the application against types of attacks such as those previously described by writing attack modules to poison the training data and determine if the attack is feasible. In addition, the defensive layer can be added to this system to determine if our layer can stop the system from training on this poisoned data. These attack modules and mitigation modules can be dropped in, removed, and replaced without adjusting the entirety of the simulation, allowing for rapid testing of many different scenarios.

VI. Future Work

We will be working on deploying our solution and testing it in a variety of environments. We are developing numerous scenarios in which to test our work and the work of others, after which we will analyze the results and classify the applications for which each approach is most effective. We will also be building a publicly available website at which other researchers can submit their security mechanisms to be assessed in our testing framework. The source code for the framework (and our algorithms) will be made publicly available for those researchers who would like to set up the system for their own testing. As part of the web interface, we are developing a user interface that will enable users to specify via a data collection module where their training data is coming from (for example, collect live data from sensor as opposed to just simulating the data from a file) and where to deploy the actual testing algorithm. Aside from enabling remote deployment to a computationally powerful backend, it will also enable users to easily deploy their applications on mobile devices for testing to ensure not only the security of the system, but also to ensure performance requirements are met on real devices used in pervasive systems.

VII. Conclusion

In this paper we have presented our ongoing work in secure machine learning for pervasive systems. We presented two main projects; enhancing the security of pervasive systems that use machine learning and our system for rapid evaluation of security mechanisms. Our work is critical to the security of numerous pervasive systems. Existing defense mechanisms that were developed for more traditional, static applications are not effective for applications that need to operate in dynamic environments experienced in pervasive systems. As the pervasive computing community further develops security mechanisms for machine learning, our testing framework will enable open and rapid comparison to help researchers and developers quickly decide what mechanisms will be most effective for their applications.

Acknowledgements

The material in this paper was supported through CAE Cybersecurity Grant H98230-15-1-0284. Any opinions expressed in this paper are those of the authors and not necessarily those of the DHS or NSA.

References

[1] Theodoros Anagnostopoulos, Christos Anagnostopoulos, Stathes Hadjiefthymiades, Miltos Kyriakakos, and Alexandros Kalousis. Predicting the Location of Mobile Users: A Machine Learning Approach. In *Proceedings of the 2009 International Conference on Pervasive Services*, ICPS '09, pages 65–72, New York, NY, USA, 2009. ACM.

[2] Battista Biggio and Pavel Laskov. Poisoning attacks against Support Vector Machines. In *In International Conference on Machine Learning (ICML*, 2012.

[3] Battista Biggio, Konrad Rieck, Davide Ariu, Christian Wressnegger, Igino Corona, Giorgio Giacinto, and Fabio Roli. Poisoning Behavioral Malware Clustering. In *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*, AISec '14, pages 27–36, New York, NY, USA, 2014. ACM.

[4] S. Bozkurt, G. Elibol, S. Gunal, and U. Yayan. A comparative study on machine learning algorithms for indoor positioning. In *2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)*, pages 1–8, September 2015.

[5] Michael Brckner, Christian Kanzow, and Tobias Scheffer. Static prediction games for adversarial learning problems. *The Journal of Machine Learning Research*, 13(1):2617–2654, January 2012.

[6] Ling Huang, Anthony D. Joseph, Blaine Nelson, Benjamin I.P. Rubinstein, and J. D. Tygar. Adversarial machine learning. page 43. ACM Press, 2011.

[7] Murat Kantarcolu, Bowei Xi, and Chris Clifton. Classifier evaluation and attribute selection against active adversaries. *Data Mining and Knowledge Discovery*, 22(1-2):291–335, January 2011.

[8] B. Lagesse, M. Kumar, and M. Wright. AREX: An Adaptive System for Secure Resource Access in Mobile P2p Systems. In *Eighth International Conference on Peer-to-Peer Computing , 2008. P2P '08*, pages 43–52, September 2008.

[9] B. Longstaff, S. Reddy, and D. Estrin. Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on-NO PERMISSIONS*, pages 1–7, March 2010.

[10] S. McBurney, E. Papadopoulou, N. Taylor, and H. Williams. Adapting Pervasive Environments through Machine Learning and Dynamic Personalization. In *International Symposium on Parallel and Distributed Processing with Applications, 2008. ISPA '08*, pages 395–402, December 2008.

[11] Blaine Nelson, Marco Barreno, Fuching Jack Chi, Anthony D. Joseph, Benjamin I. P. Rubinstein, Udam Saini, Charles Sutton, J. D. Tygar, and Kai Xia. Misleading Learners: Co-opting Your Spam Filter. In *Machine Learning in Cyber Trust*, pages 17–51. Springer US, 2009. DOI: 10.1007/978-0-387-88735-7_2.

[12] Thuong Nguyen, Dinh Phung, S. Gupta, and S. Venkatesh. Extraction of latent patterns and contexts from social honest signals using hierarchical Dirichlet processes. In *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 47–55, March 2013.

[13] D.E. Phillips, Rui Tan, M. Moazzami, Guoliang Xing, Jinzhu Chen, and D.K.Y. Yau. Supero: A sensor system for unsupervised residential power usage monitoring. In *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 66–75, March 2013.