Part I

Overview of Adversarial Machine Learning

1 Introduction

Machine learning has become a prevalent tool in many computing applications. With the rise of machine learning techniques, however, comes a concomitant risk. Adversaries may attempt to exploit a learning mechanism either to cause it to misbehave or to extract or misuse information.

This book introduces the problem of secure machine learning; more specifically, it looks at learning mechanisms in adversarial environments. We show how adversaries can effectively exploit existing learning algorithms and discuss new learning algorithms that are resistant to attack. We also show lower bounds on the complexity of extracting information from certain kinds of classifiers by probing. These lower bound results mean that any learning mechanism must use classifiers of a certain complexity or potentially be vulnerable to adversaries who are determined to evade the classifiers. Training data privacy is an important special case of this phenomenon. We demonstrate that while accurate statistical models can be released that reveal nothing significant about individual training data, fundamental limits prevent simultaneous guarantees of strong privacy and accuracy.

One potential concern with learning algorithms is that they may introduce a security fault into systems that employ them. The key strengths of learning approaches are their adaptability and ability to infer patterns that can be used for predictions and decision making. However, these advantages of machine learning can potentially be subverted by adversarial manipulation of the knowledge and evidence provided to the learner. This exposes applications that use machine learning techniques to a new class of security vulnerability; i.e., learners are susceptible to a novel class of attacks that can cause the learner to disrupt the system it was intended to benefit. In this book we investigate the behavior of learning systems that are placed under threat in security-sensitive domains. We will demonstrate that learning algorithms are vulnerable to a myriad of attacks that can transform the learner into a liability for the system they are intended to aid, but that by critically analyzing potential security threats, the extent of these threats can be assessed and proper learning methods can be selected to minimize the adversary's impact and prevent system failures.

We investigate both the practical and theoretical aspects of applying machine learning to security domains in five main foci: a taxonomy for qualifying the security vulnerabilities of a learner, two novel practical attacks and countermeasure case studies, an algorithm for provable privacy-preserving learning, and methods for evading detection by a classifier. We present a framework for identifying and analyzing threats to learners and use it to systematically explore the vulnerabilities of several proposed learning systems. For these systems, we identify real-world threats, analyze their potential impact, and study learning techniques that significantly diminish their effect. Further,

we discuss models for privacy-preserving learning and evasion of classifiers and use those models to defend against, and analyze, classifier vulnerabilities. In doing so, we provide practitioners with guidelines to identify potential vulnerabilities and demonstrate improved learning techniques that are resilient to attacks. Our research focuses on learning tasks in virus, spam, and network anomaly detection, but also is broadly applicable across many systems and security domains and has momentous implications for any system that incorporates learning. In the remainder of this chapter, we further motivate the need for a security analysis of machine learning algorithms and provide a brief history of the work that led us to this research and the lessons learned from it.

Our work has wide applicability. While learning techniques are already common for tasks such as natural language processing (cf. Jurafsky & Martin 2008), face detection (cf. Zhao, Chellappa, Phillips, & Rosenfeld 2003), and handwriting recognition (cf. Plamondon & Srihari 2000), they also have potentially far-reaching utility for many applications in security, networking, and large-scale systems as a vital tool for data analysis and autonomic decision making. As suggested by Mitchell (2006), learning approaches are particularly well suited to domains where either the application *i*) is too complex to be designed manually or *ii*) needs to dynamically evolve. Many of the challenges faced in modern enterprise systems meet these criteria and stand to benefit from agile learning algorithms able to infer hidden patterns in large complicated datasets, adapt to new behaviors, and provide statistical soundness to decision-making processes. Indeed, learning components have been proposed for tasks such as performance modeling (e.g., Bodík, Fox, Franklin, Jordan, & Patterson 2010; Bodík, Griffith, Sutton, Fox, Jordan, & Patterson 2009; Xu, Bodík, & Patterson 2004), enterprise-level network fault diagnosis (e.g., Bahl, Chandra, Greenberg, Kandula, Maltz, & Zhang 2007; Cheng, Afanasyev, Verkaik, Benkö, Chiang, Snoeren, Savage, & Voelker 2007; Kandula, Chandra, & Katabi 2008), and spam detection (e.g., Meyer & Whateley 2004; Segal, Crawford, Kephart, & Leiba 2004).

1.1 Motivation

Machine learning techniques are being applied to a growing number of systems and networking problems, a tendency that can be attributed to two emerging trends. First, learning techniques have proven to be exceedingly successful at finding patterns in data-rich domains and have provided statistically grounded techniques applicable to a wide variety of settings. In rapidly changing environments, machine learning techniques are considerably advantageous over handcrafted rules and other approaches because they can infer hidden patterns in data, they can adapt quickly to new signals and behaviors, and they can provide statistical soundness to a decision-making process. Second, the need to protect systems against malicious adversaries continues to increase across systems and networking applications. Rising levels of hostile behavior have plagued many application domains including email, web search, pay-per-click advertisements, file sharing, instant messaging, and mobile phone communications. The task of detecting (and subsequently preventing) such malicious activity is broadly known as the malfeasance detection problem, and it includes spam, fraud, intrusion, and virus detection. In such problem domains, machine learning techniques are arguably necessary because

they provide the ability for a system to respond more readily to evolving real-world data, both hostile and benign, and to learn to identify or possibly even prevent undesirable activities.

In the malfeasance detection problem, machine learning techniques are proving themselves to be an invaluable tool to maintain system security. From spam filtering to malware detection to fast attack response and many other applications, machine learning is quickly becoming a useful tool for computer security. For example, network intrusion detection systems (NIDSs) monitor network traffic to detect abnormal activities such as attempts to infiltrate or hijack hosts on the network. The traditional approach to designing an NIDS relies on an expert to codify rules defining normal behavior and intrusions (e.g., Paxson 1999). Because this approach often fails to detect novel intrusions, a number of researchers have proposed incorporating machine learning techniques into intrusion detection systems (e.g., Mahoney & Chan 2002; Lazarevic, Ertöz, Kumar, Ozgur, & Srivastava 2003; Mukkamala, Janoski, & Sung 2002; Eskin, Arnold, Prerau, Portnoy, & Stolfo 2002). Machine learning techniques offer the benefit of detecting novel patterns in traffic—which presumably represent attack traffic—by being trained on examples of innocuous (known good) and malicious (known bad) traffic data. Learning approaches to malfeasance detection have also played a prominent role in modern spam filtering (e.g., Meyer & Whateley 2004; Segal et al. 2004) and have been proposed as elements in virus and worm detectors (e.g., Newsome, Karp, & Song 2005; Stolfo, Hershkop, Wang, Nimeskern, & Hu 2003; Stolfo, Li, Hershkop, Wang, Hu, & Nimeskern 2006), host-based intrusion detection systems (HIDSs) (e.g., Forrest, Hofmeyr, Somayaji, & Longstaff 1996; Hofmeyr, Forrest, & Somayaji 1998; Mutz, Valeur, Vigna, & Kruegel 2006; Somayaji & Forrest 2000; Warrender, Forrest, & Pearlmutter 1999), and some forms of fraud detection (cf. Bolton & Hand 2002). These systems utilize a wide variety of machine learning techniques including clustering, Bayesian inference, spectral analysis, and maximum-margin classification that have been demonstrated to perform well for these diverse dynamical domains. However, many such techniques also are susceptible to attacks against their learning mechanism, which jeopardize learning systems used in any adversarial setting.

However, while there is an increasing need for learning algorithms to address problems like malfeasance detection, incorporating machine learning into a system must be done carefully to prevent the learning component itself from becoming a means for attack. The concern is that, in security-sensitive domains, learning techniques may expose a system to the threat that an adversary can maliciously exploit vulnerabilities that are unique to learning. Pursuing these exploits is particularly incentivized when learning techniques act as countermeasures against cybercrime threats; e.g., in malfeasance detection. With growing financial incentives to engage in cybercrime inviting ever more sophisticated adversaries, attacks against learners present a lucrative new means to disrupt the operations of or otherwise damage enterprise systems. This makes assessing the vulnerability of learning systems an essential problem to address to make learning methods effective and trustworthy in security-sensitive domains.

The essence of this threat comes from the ability of an adversary to adapt against the learning process. A well-informed adversary can alter its approach based on knowledge of the learner's shortcomings or mislead it by cleverly crafting data to corrupt or deceive

the learning process; e.g., spammers regularly adapt their messages to thwart or evade spam detectors. In this way, malicious users can subvert the learning process to disrupt a service or perhaps even compromise an entire system. In fact, a growing body of literature, which we discuss in detail in Chapter 3, shows that attackers can indeed successfully attack machine learning systems in a variety of application domains including automatic signature generation (Chung & Mok 2006, 2007; Newsome, Karp, & Song 2006), intrusion detection systems (Fogla & Lee 2006; Tan, Killourhy, & Maxion 2002), and email spam filtering (Lowd & Meek 2005*b*; Wittel & Wu 2004). It is imperative to ensure that learning is successful despite such attacks—in other words, to achieve *secure learning*.

The primary vulnerability in learners that attackers can exploit lies in the assumptions made about the learners' data. Many common learning algorithms assume that their training and evaluation data come from a natural or well-behaved distribution that remains stationary over time, or at worst, drifts gradually in a benign way. However, these assumptions are perilous in a security-sensitive domain—settings where a patient adversary has motive and the capability to alter the data used by the learner for training or prediction. In such a domain, learners can be manipulated by an intelligent adversary capable of cleverly violating the learners' assumptions for their own gains, making learning and adaptability into potential liabilities for the system rather than benefits. We analyze how learners behave in these settings and we explore alternative methods that can bolster resilience against an adversary.

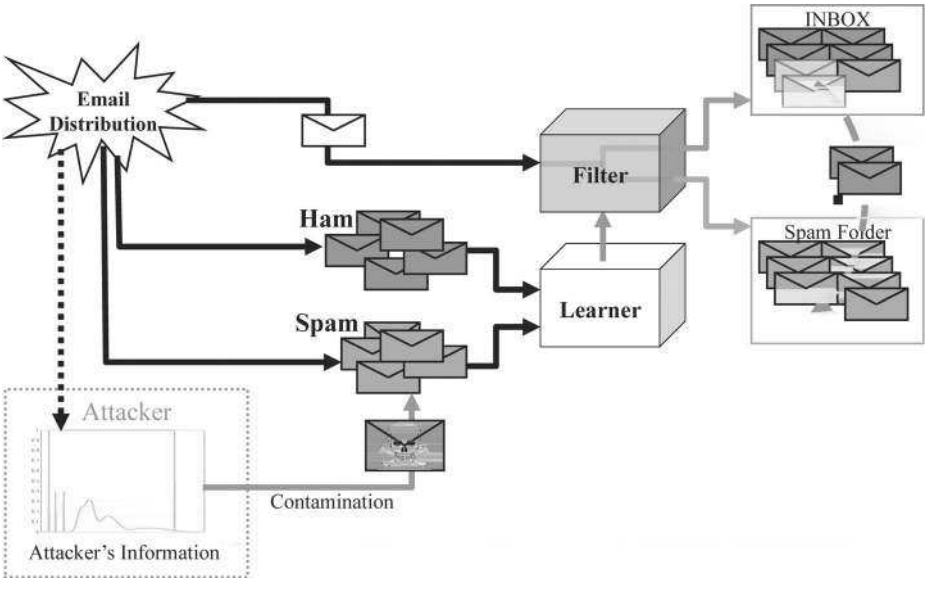
We consider several potential dangers posed to a learning system. The principal threat is that an attacker can exploit the adaptive nature of a machine learning system to mistrain it and cause it to fail. Failure includes causing the learning system to produce classification errors: if it misidentifies a hostile instance as benign, then the hostile instance is erroneously permitted through the security barrier; if it misidentifies a benign instance as hostile, then a permissible instance is erroneously rejected and normal user activity is interrupted. The adversarial opponent has the potential ability to design training data to cause a learning system to mistakenly make decisions that will misidentify instances and degrade the overall system. If the system's performance sufficiently degrades, users will lose confidence in it and abandon it, or its failures may even significantly compromise the integrity of the system. A second threat is that the learner will reveal secrets about its training data and thereby compromise its data's privacy. In this case, the failure concerns the amount of information inadvertently leaked by the learner, rather than being a direct consequence of the decisions it makes. Learning algorithms necessarily reveal some information about their training data to make accurate predictions, which could potentially lead to a breach of privacy, again eroding the confidence of users. These threats raise several questions. *What techniques can a patient adversary use to mistrain or evade a learning system or compromise data privacy?* and *How can system designers assess the vulnerability of their system to vigilantly incorporate trustworthy learning methods?* We provide a framework for a system designer to thoroughly assess these threats and demonstrate how it can be applied to evaluate real-world systems.

Developing robust learning and decision-making processes is of interest in its own right, but for security practitioners, it is especially important. To effectively apply

machine learning as a general tool for reliable decision making in computer systems, it is necessary to investigate how these learning techniques perform when exposed to adversarial conditions. Without an in-depth understanding of the performance of these algorithms in an adversarial setting, the systems will not be trusted and will fail to garner wider adoption. Worse yet, a vulnerable system could be exploited and discourage practitioners from using machine learning in the future. Hence, it is essential for security practitioners to analyze the risks associated with learning algorithms and select techniques that adequately minimize these risks. When a learning algorithm performs well under a realistic adversarial setting, it is an algorithm for secure learning. Of course, whether an algorithm's performance is acceptable is a highly subjective judgment that depends both on the constraints placed on the adversary and on the job the algorithm is tasked with performing. This raises two fundamental questions: *What are the relevant security criteria necessary to evaluate the security of a learner in a particular adversarial environment?* and *Are there machine learning techniques capable of satisfying the security requirements of a given problem domain, and how can such a learner be designed or selected?* We demonstrate how learning systems can be systematically assessed and how learning techniques can be selected to diminish the potential impact of an adversary.

We now present four high-level examples (1.1 to 1.4) that describe different attacks against a learning system. Each of these examples is a preview of the in-depth case studies that we will comprehensively analyze in Chapters 5, 6, 7, and 8. In each synopsis we motivate the learning task and the goal of the adversary; we then briefly describe plausible attacks that align with these goals.

Example 1.1 (Spam Filter and Data Sanitization)



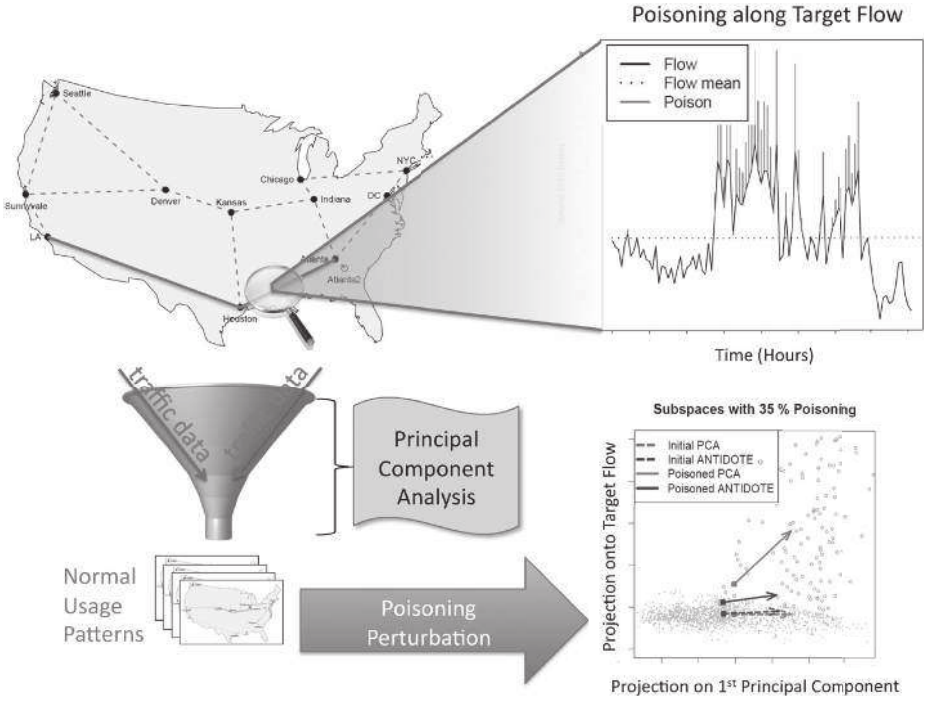
Email spam filtering is one of the most well-known applications of machine learning. In this problem, a set of known good email (ham) and unwanted email (spam) messages

is used to train a spam filter. The learning algorithm identifies relevant characteristics that distinguish spam from ham (e.g., tokens such as "Viagra," "Cialis," and "Rolex" or envelope-based features) and constructs a classifier that combines observed evidence of spam to make a decision about whether a newly received message is spam or ham.

Spam filters have proven to be successful at correctly identifying and removing spam messages from a user's regular messages. This has inspired spammers to regularly attempt to evade detection by obfuscating their spam messages to confuse common filters. However, spammers can also corrupt the learning mechanism. As depicted in the diagram above, a spammer can use information about the email distribution to construct clever *attack spam* messages that, when trained on, will cause the spam filter to misclassify the user's desired messages as spam. Ultimately, this spammer's goal is to cause the filter to become so unreliable that the user can no longer trust that its filter has accurately classified the messages and must sort through spam to ensure that important messages are not erroneously filtered.

In Chapter 5, we demonstrate several variants of this attack based on different goals for the spammer and different amounts of information available to it. We show that this attack can be quite effective: if a relatively small number of attack spam messages are trained on, then the accuracy of the filter is significantly reduced. However, we also show that a simple data sanitization technique designed to detect deleterious messages is effective in preventing many of these attacks. In this case, the attacker's success depends primarily on the scope of its goal to disrupt the user's email.

Example 1.2 (Network Anomaly Detector)

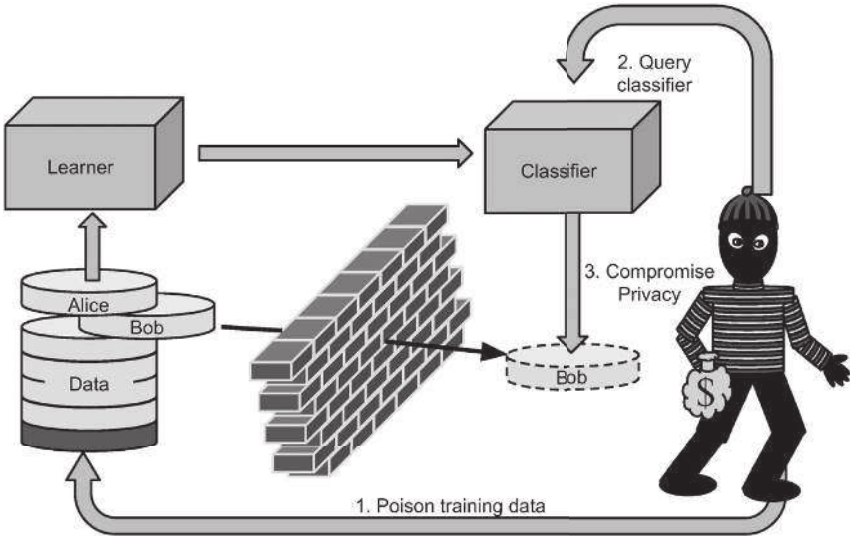


Machine learning techniques have also been proposed by Lakhina, Crovella, and Diot (2004*b*) for detecting network volume anomalies such as denial-of-service (DoS) attacks. Their proposal uses a learning technique known as principal component analysis (PCA) to model normal traffic patterns so as to identify anomalous activity in the network. We demonstrate that this technique is also susceptible to contamination.

As depicted in the above diagram, PCA is first used to extract patterns from traffic observed in a backbone communications network to construct a normal model. This model is subsequently used to detect DoS attacks. An adversary determined to launch a DoS attack must first evade this detector. A crafty adversary can successfully evade detection by mistraining the detector. The attacker can systematically inject chaff traffic that is designed to make its target flow align with the normal model—this chaff (depicted in the top-right figure) is added along the target flow to increase its variance. The resulting perturbed model (see the bottom-right figure) is unable to detect DoS attacks along the target flow.

We explore attacks against the PCA-based detector in Chapter 6 based on different sources of information available to the adversary. Attacks against PCA prove to be effective—they successfully increase its rate of misdetection eight- to tenfold. We also explore an alternative robust statistics-based detection approach called ANTIDOTE designed to be more resilient to chaff. The evasion success rate for the same attacks against ANTIDOTE is roughly halved compared to the PCA-based approach. However, resilience to poisoning comes at a price—ANTIDOTE is less effective on nonpoisoned data than is the original detector.

Example 1.3 (Privacy-Preserving Learning)



Privacy is another important facet for learning practitioners to consider. In many situations, a practitioner may want to employ a learning algorithm on privileged data to subsequently provide a public utility without compromising data privacy. For example, a hospital may want to use private medical records to construct a classifier that can

identify likely H1N1 swine flu patients, and they may want to share that classifier with the general public in the form of a self-assessment tool (Microsoft 2009). However, in providing this classifier, the health care provider must not expose privileged information from its records. As such, it requires strong guarantees that the classifier will not compromise the privacy of its training data.

Learning algorithms pose a risk to privacy because the behavior of the learner is a reflection of its data and hence may reveal the underlying secrets contained within. Fundamentally, a learning algorithm produces a summary of data it was trained on based on the patterns it gleans from that data. This summary reveals aggregate information about the data and can potentially be exploited by an adversary to violate a specific datum's privacy. It is possible that a clever adversary could contaminate the learner's data or query the learner to eventually infer private data.

Privacy-preserving learning is a field within learning, statistical databases, and theory that studies the privacy properties of learning algorithms and seeks to develop learning algorithms with strong privacy guarantees (cf. Dwork 2010). In Chapter 7, we explore a model that provides strong privacy-preserving guarantees and develop a privacy-preserving support vector machine within that model. Further, we explore the limits of privacy-preserving learning that demonstrate the fundamental tradeoff between accuracy and privacy preservation.

Example 1.4 (Near-Optimal Evasion)