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,

https://doi.org/10.1017/9781107338548.001 Published online by Cambridge University Press

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

https://doi.org/10.1017/9781107338548.001 Published online by Cambridge University Press

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

https://doi.org/10.1017/9781107338548.001 Published online by Cambridge University Press

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.

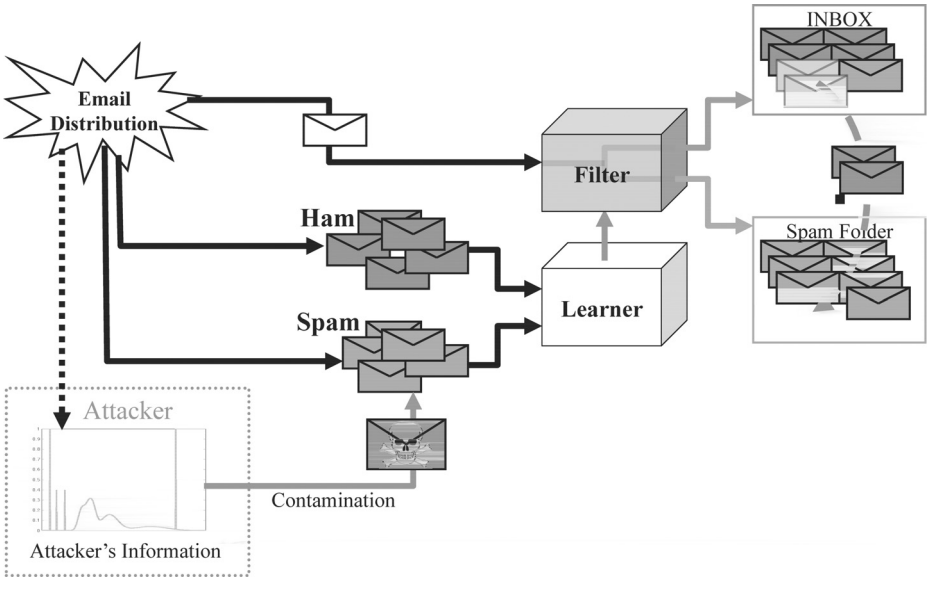
Developing robust learning and decision-making processes is of interest in its own right, but for security practitioners, it is especially important. To effectively apply

https://doi.org/10.1017/9781107338548.001 Published online by Cambridge University Press

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

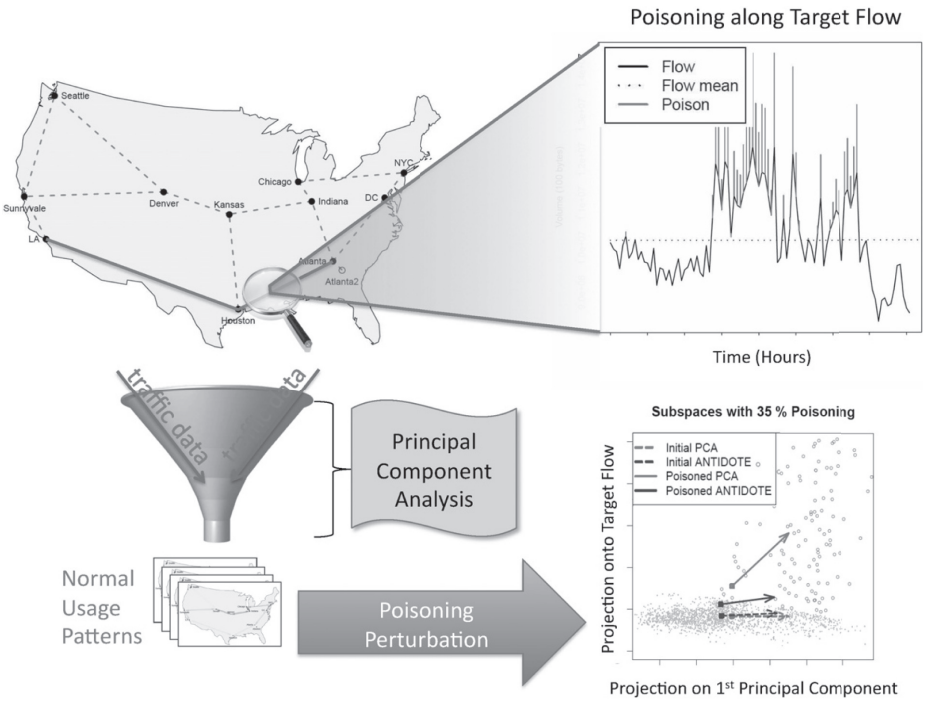
https://doi.org/10.1017/9781107338548.001 Published online by Cambridge University Press

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)



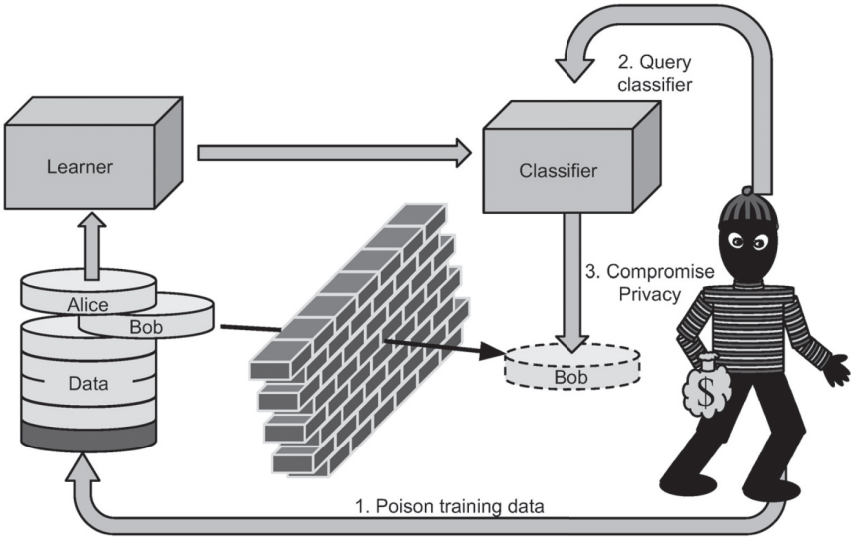
https://doi.org/10.1017/9781107338548.001 Published online by Cambridge University Press

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

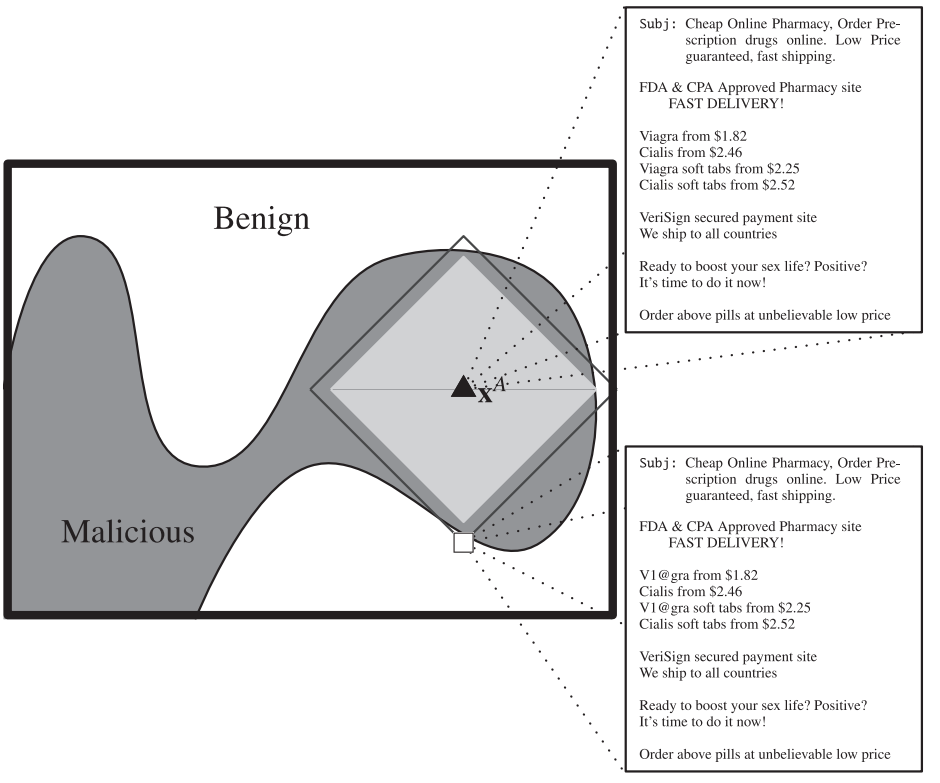
https://doi.org/10.1017/9781107338548.001 Published online by Cambridge University Press

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)



https://doi.org/10.1017/9781107338548.001 Published online by Cambridge University Press

In addition to misleading learning algorithms, attackers also have an interest in evading detectors by making their miscreant activity undetectable. As previously mentioned in Example 1.1, this practice is already common in the spam filtering domain where spammers attempt to evade the filter by *i*) obfuscating words indicative of spam to human-recognizable misspellings; e.g., "Viagra" to "V1@gra" or "Cialis" to "Gia|is," *ii*) using HTML to make an unrendered message difficult to parse, *iii*) adding words or text from other sources unrelated to the spam, and *iv*) embedding images that display a spam message. All of these techniques can be used to evade spam filters, but they also are costly for the spammer—altering its spam can make the message less profitable because the distortions reduce the message's legibility or its accessibility. In evading the filter, the spammer would like to minimally modify its messages, but for a dynamically learned filter, the spammer does not know the learned filtering rules. Instead, the spammer constructs test spams for probing the filter and refining modifications according to some cost function. This raises the following question: *How difficult is it for the spammer to optimally evade the filter by querying it?*

The near-optimal evasion problem, which we examine in Chapter 8, formalizes this question in terms of the query complexity required by the spammer to evade a particular family of classifiers. We study the broad family of convex-inducing classifiers and show that there are efficient algorithms for near-optimal evasion under certain ℓ_p cost functions.

1.2 A Principled Approach to Secure Learning

In this book, we analyze four separate security problems for machine learning systems. In each, we first specify the threat model posed, subsequently analyze the threat's impact, and, where appropriate, propose defenses against the threat. It is a well-established practice in computer security that evaluating a system involves a continual process of, first, determining classes of attacks on the system; second, evaluating the resilience of the system against those attacks; and third, strengthening the system against those classes of attacks. Within each of the security problems we consider, we follow exactly this model to evaluate the vulnerabilities of learning algorithms.

To assess the vulnerabilities of learning systems, our approach builds on many well-established principles from traditional computer security. Generally speaking, computer security is concerned with quantifying, managing, and reducing the risks associated with computer systems and their usage. Traditional topics in security include cryptography, authentication, secure channels, covert channels, defensive programming practices, static code analysis, network security, and operating system security; traditional (code-based) vulnerabilities include buffer overflows, format string vulnerabilities, cross application scripting, code injection attacks, and privilege escalation. Unlike classical security settings, attacks against a learning system exploit the adaptive nature of the learning system. Not only can the adversary exploit existing flaws in the learner,

https://doi.org/10.1017/9781107338548.001 Published online by Cambridge University Press

it can also mislead the learner to create new vulnerabilities. Nonetheless, classical security principles are still applicable for analyzing machine learning algorithms. In particular, the principles of *proactively studying attacks*, *Kerckhoffs' Principle*, *conservative design*, and *formal threat modeling* are the foundations of our approach.

Proactive Analysis

The first guideline of computer security is to conduct proactive studies to anticipate potential attacks before a system is deployed or widely used. Analysis of and open debate about the security of a system provide a level of confidence in it. Further, if vulnerabilities are successfully identified before deployment, the system's design can be preemptively repaired, thereby reducing the chance that costly patches, rewrites, or recalls of a flawed system will be necessary after its deployment. Finally, revelations of security flaws in a deployed system erode users' confidence in it and can lead to costly damages, which may have been avoidable had the system been adequately vetted. In this book we advocate a proactive approach to the security of learning systems. For the already existing systems that we analyze, we seek to identify their vulnerabilities and raise awareness about them before the system is damaged or compromised by an unscrupulous adversary. Further, we not only expose vulnerabilities but we also offer alternative systems that thwart their exploits or mitigate their effect. Lastly, we provide general guidelines to system designers to aid them in analyzing the vulnerabilities of a proposed learning system so that learning can be deployed as an effective and reliable component even in critical systems.

Kerckhoffs' Principle

The second guideline often referred to as *Kerckhoffs' Principle* (Kerckhoffs 1883) is that the security of a system should not rely on unrealistic expectations of secrecy. An over-dependence on secrets to provide security is dangerous because if these secrets are exposed the security of the system is immediately compromised. Ideally, secure systems should make minimal assumptions about what can realistically be kept secret from a potential attacker. The field of cryptography has embraced this general principle by demanding open algorithms that only require a secret *key* to provide security or privacy. We apply this principle to analyzing machine learning systems primarily by assuming that the adversary is aware of the learning algorithm and can obtain some degree of information about the data used to train the learner. However, determining the appropriate degree of secrecy that is feasible for secure machine learning systems is a difficult question, which we discuss further in Chapter 9. In each of the chapters of this book, we consider various levels of information that the adversary potentially obtains and assess how the adversary can best utilize this information to achieve its objective against the learner. In doing so, we demonstrate the impact of different levels of threat and show the value an adversary obtains from a particular source of information.

Conservative Design

The third foundational principle we apply is that the design and analysis of a system should avoid unnecessary or unreasonable assumptions about and limitations on

https://doi.org/10.1017/9781107338548.001 Published online by Cambridge University Press

the adversary. If designers fail to anticipate the capabilities of an adversary, underestimate the attacker's tenacity, or misunderstand the resources and information at its disposal, major security compromises can occur. Instead, by assuming the adversary has the broadest possible powers, one can understand the worst-case threat posed by an adversary, and users are less likely to be surprised by an attack by some unanticipated adversary. Conversely, however, analyzing the capabilities of an omnipotent limitless adversary reveals little about a learning system's behavior against realistic constrained attackers and may lead to an unnecessarily bleak outlook on the security of a system. Instead, while we also consider worst-case adversaries, our approach focuses on constructing an appropriate threat model to quantify the relationship between the adversary's effort and its effect on the system under a variety of threat levels.

Threat Modeling

Finally, to analyze the vulnerabilities of machine learning systems, we follow the typical security practice of constructing a formal (attacker-centric) threat model to analyze a learning system's vulnerabilities. As stated by Denning & Denning (1979),

> No mechanism is perfectly secure. A good mechanism reduces the risk of compromise to an acceptable level.

Under the approach we advocate, a system analyst constructs a threat model of a learning system to analyze the potential risks against it. This threat model describes each potential adversary in terms of its incentives, objectives, resources, and capabilities. The threat model allows the analyst to quantify the degree of security; that is, the level of security expected against potential adversaries in terms of the damage that can be inflicted and the feasibility of such an attack. Ultimately, this characterizes the security of the system, helps to identify its weaknesses, and provides a mechanism to compare the security of different proposals.

To construct a threat model for a particular learning system, first the analyst quantifies the security setting and objectives of that system to develop criteria to measure success and quantify the level of security offered. Formalizing the risks and objectives allows the analyst to identify potential limitations of the system and potential attacks and focuses this analysis on immediate threats, so as to avoid wasting effort protecting against nonexistent or ancillary threats. Next the analyst identifies potential adversarial incentives, objectives, resources, capabilities, and limitations. By examining the nature of anticipated adversaries and their goals, the analyst can quantify the effort required by the adversary to achieve a particular objective. Based on this threat model, the analyst can then analyze the security of the system and construct appropriate defenses against realistic forms of attack. Formal analysis provides a rigorous approach to security. Additionally, by formalizing the threats and security of a system, other analysts can critique assumptions made by the analyst and suggest potential flaws in the system's design. This open process tends to improve a system's security.

https://doi.org/10.1017/9781107338548.001 Published online by Cambridge University Press

1.3 Chronology of Secure Learning

The discipline of secure learning has emerged as an important field of study from a series of important developments within the fields of artificial intelligence and computer security primarily in the last two decades. Contributions that led to the formation of secure learning, however, originated with influential works from both disciplines dating decades earlier. While many of these developments addressed separate problems within these otherwise distinct communities, they share some common themes that converged as many security problems began to require adaptive approaches and as the vulnerability of learning techniques to malicious data became apparent. Together these trends formed a basis for the field of secure learning that has since coalesced. Here we provide a chronology of the major events that contributed to this development to better understand the context and motivations for secure learning. While this is not meant to be a complete history of all the significant work that forms the foundation of secure learning, this brief chronology provides a few highlights from the fields of computer security and artificial intelligence that contributed to the formation of secure learning as a discipline. See Chapter 3 for additional discussion of the secure learning framework and the organization of research in the field.

1940s Wartime cryptanalysis: Using computers, many of the Axis Powers' crypto-ciphers are broken. Claude Shannon publishes declassified material in his paper, "Communication Theory of Secrecy Systems," which shows that any theoretically unbreakable crypto-system has the same requirements as a one-time pad (Shannon 1949). The standards developed for crypto-systems later serve as a model for standards of computer security.

1940s and 1950s Foundation of artificial intelligence: With the advent of early computing machines, researchers began investigating artificial intelligence. Famously, in 1950 Turing publishes Computing Machinery and Intelligence and introduces the famous *Turing Test* for determining whether a machine can demonstrate intelligence (Turing 1950).

1960s and 1970s Foundation of robust statistics: While the history of robust statistics dates back to the origins of statistics, the first formal methods in this area arose in the 1960s and 1970s. The principal texts summarizing this work are Huber (1981) and Hampel, Ronchetti, Rousseeuw, and Stahel (1986), which present two different approaches to robust statistics: the minimax approach and the approach based on the influence function, respectively.

Late 1970s Advent of public-key cryptography: First proposed by Whitfield Diffie and Martin Hellman in 1976, public-key cryptography revolutionized cryptography (Diffie & Hellman 1976). Ronald Rivest, Adi Shamir, and Len Adleman, in 1978, introduce the RSA public-key system (Rivest, Shamir, & Adleman 1978).

1979 Denning & Denning publish "Data Security": In this seminal paper, Dorothy and Peter Denning introduce four different forms of safeguards for data: access controls, flow controls, inference controls, and encryption (Denning

https://doi.org/10.1017/9781107338548.001 Published online by Cambridge University Press

& Denning 1979). With inference controls, they address the need for guarding against information leaks in database aggregation procedures, a topic that precedes the broader notion of privacy-preserving learning.

1984 Introduction of PAC learning framework: Valiant introduces probably approximately correct (PAC) learning as a framework for studying the ability of learning algorithms to learn a particular concept given instances exemplifying it (Valiant 1984). Importantly, the PAC framework ties learning theory to hypothesis complexity theory, and belongs in the area of empirical process theory within mathematical statistics.

1990s Polymorphic viruses are developed: Computer viruses begin using encryption, code obfuscation, and polymorphism to avoid detection (Chen & Robert 2004). These stealthy viruses are engineered to thwart signature-based antivirus detectors, making it increasingly difficult to identify malicious code via signatures.

1993 Kearns and Li study learning under malicious errors: Using the PAC model, Kearns and Li (1993) investigate the feasibility of learning with worst-case errors in the training data in their paper, "Learning in the Presence of Malicious Errors."

1996 First commercial Bayesian spam filters: The iFile program written by Jason Rennie is credited as the first mail program that incorporated the probabilistic approach to spam filtering that is popularly known as *Bayesian filtering*. The first research paper on Bayesian methods for spam filtering was published by Sahami, Dumais, Heckerman, and Horvitz (1998).

1999–2004 Emergence of fast-spreading computer worms: Capitalizing on the spread of the Internet, computer worms including Melissa, ILOVEYOU, Code Red, Nimda, Klez, Slammer, Blaster, Sobig, MyDoom, and Netsky quickly infect vulnerable hosts and cause widespread damage to computer and network resources (Chen & Robert 2004). These prolific and damaging worms highlight the need for fast adaptive detection systems.

2002 Paul Graham's approach to spam filtering: In his essay, "A Plan for Spam," Paul Graham refined and popularized the so-called Bayesian methods for spam filtering (Graham 2002). The basic technique he introduced is still employed today in many spam filters, including SpamBayes, BogoFilter, and the learning component of the Apache SpamAssassin filter (Apa n.d.).

August 2006 AOL search data leak: AOL Research releases 20 million search keywords of 650,000 users over a three-month period, which were anonymized by replacing names with a unique key. The *New York Times* and other organizations denonymized these users by cross-referencing public information (Barbaro & Zeller 2006). This led to a class action lawsuit against AOL.

2006 Differential privacy proposed: With the development of differential privacy (Dwork, McSherry, Nissim, & Smith 2006), learning while strongly preserving privacy was set on a firm footing.

January 2007 Storm worm discovered: This malware spreads through a Trojan horse and creates a network of compromised machines called a *botnet*—Storm

https://doi.org/10.1017/9781107338548.001 Published online by Cambridge University Press

was one of the first botnets discovered (cf. Holz, Steiner, Dahl, Biersack, & Freiling 2008). Botnets have been used for a variety of malicious purposes including sending spam, distributed denial-of-service (DDos) attacks, and cybertheft, significantly facilitating these abusive activities.

December 2007 NIPS Workshop on Machine Learning in Adversarial Environments for Computer Security: The first formal workshop to address the role of machine learning as a tool for computer security was held in conjunction with the conference on Advances in Neural Information Processing Systems (NIPS). This workshop, organized by Richard Lippmann and Pavel Laskov, brought together researchers from both fields and led later to a special issue on this topic in the journal, *Machine Learning* (Laskov & Lippmann 2010).

October 2008–November 2016 CCS Workshop on Security and Artificial Intelligence: The longest running workshop organized to address uses of artificial intelligence within computer security was first held in 2008. This first workshop was organized by Dirk Balfanz and Jessica Staddon and held in conjunction with the ACM Conference on Computer and Communications Security (CCS) and continues to meet annually through the time of this writing in 2016 (Balfanz & Staddon 2008, 2009; Greenstadt 2010; Cárdenas, Greenstadt, & Rubinstein 2011; Cárdenas, Nelson, & Rubinstein 2012; Nelson, Dimitrakakis, & Shi 2013; Dimitrakakis, Mitrokotsa, & Rubinstein 2014; Dimitrakakis, Mitrokotsa, & Sinha 2015; Freeman, Mitrokotsa, & Sinha 2016).

December 2009 Netflix privacy breach: A class action lawsuit is brought against Netflix, Inc., for releasing private information of about 500,000 customers in a movie rating contest for machine learning. Researchers were able to deanonymize the disclosed users by cross-referencing their ratings with those on IMDB (Narayanan & Shmatikov 2008).

September 2010 Workshop on Privacy and Security issues in Data Mining and Machine Learning (PSDML): The PSDML workshop is held in conjunction with the ECML/PKDD conference (Dimitrakakis, Gkoulalas-Divanis, Mitrokotsa, Verykios, & Saygin 2011). This workshop provides machine learning researchers with a forum for discussion of privacy and security applications.

September 2012 Dagstuhl Perspectives Workshop: Machine Learning Methods for Computer Security: A workshop is convened in Schloss Dagstuhl, Germany, to bring together leading researchers from both the computer security and machine learning communities to discuss challenges and future research directions for adversarial learning and learning-based security techniques and to foster a common community Joseph et al. (2013).

June 2014 Workshop on Learning, Security, and Privacy: Held in conjunction with the International Conference on Machine Learning (ICML), this workshop provides a new venue for machine learning researchers to address problems with security and privacy considerations (Dimitrakakis, Laskov, Lowd, Rubinstein, & Shi 2014).

https://doi.org/10.1017/9781107338548.001 Published online by Cambridge University Press

February 2016 Workshop on Artificial Intelligence for Cyber Security (AICS): The AICS workshop, organized by groups from MIT Lincoln Labs and the University of Southern California, is held in conjunction with the premier AI conference AAAI (Martinez, Streilein, Carter, & Sinha 2016). The workshop brings together members of the network security, security games, and learning communities.

1.4 Overview

In this book, we present a systematic approach for identifying and analyzing threats to a machine learning system. We examine a number of real-world learning systems, assess their vulnerabilities, demonstrate real-world attacks on their learning mechanism, and propose defenses that can successfully mitigate the effectiveness of such attacks. In doing so, we provide a systematic methodology for assessing a learner's vulnerability and developing defenses to strengthen its system against such threats. Additionally, we also examine and answer theoretical questions regarding the limits of adversarial contamination, privacy-preserving learning, and classifier evasion.

This text is organized into four parts. In the first part, we present the background and foundational materials that form the basis of our approach. In Chapter 2, we present a synopsis of machine learning and introduce our notation. Then in Chapter 3, we introduce a framework for assessing the security properties of learning agents, present the taxonomy of attacks against learners, and categorize and discuss the prior work within the context of this framework. We identify different classes of attacks on machine learning systems and show that there are at least three interesting dimensions on which to qualify threats to a learning system. Further, we cast secure learning as a game between an *attacker* and a *defender*. The taxonomy determines the structure of this game and cost model for our players. Based on this framework, the remainder of the text is organized according to different vulnerabilities in learning systems.

In the second part of the book, we explore attacks in which the attacker actively interferes with learning by maliciously tampering with the learner's training data. In Chapter 4, we investigate how an attacker could poison a hypersphere-based anomaly detector through an iterative attack. For several different attack scenarios, we quantify the effort required by an attacker to achieve a successful poisoning result in terms of the amount of data the attacker must control and the number of required attack iterations. We provide bounds on the difficulty of the problem for the attacker under a number of different attack and retraining scenarios and discuss how these results affect the attack's feasibility.

We next provide two real-world case studies of practical learning systems, and for both, we use our framework to methodically explore potential threats to these systems; finally we suggest corresponding defenses that counter or mitigate the effects of the attacks. The first learning system is a spam filter called SpamBayes that we investigate

https://doi.org/10.1017/9781107338548.001 Published online by Cambridge University Press

in Chapter 5. We show that this spam filter is highly vulnerable to adversarial contamination of its training data. In particular, we show how an attacker can construct highly effective attack messages that, once injected into the learner's training set, cause SpamBayes to subsequently misclassify many normal messages as spam. However, we also propose a data sanitization defense that is able to successfully detect and remove attack messages based on the estimated damage the message causes. This defense effectively eliminates most attack messages and only has a minimal impact on the filter's performance. The second learning system we analyze is a network anomaly detection system in which a PCA-based subspace estimation technique is used to identify network-wide DoS attacks in a backbone communication network. In Chapter 6, we study a class of data poisoning strategies against this PCA-based anomaly detector. The goal of the adversary in this system is to evade detection, and to do so, it contaminates the learner's training data by gradually perturbing the network traffic used to train the detector. Again, we show that this class of algorithms is highly susceptible to poisoning methods. To combat our attacks against this detector, we also propose an alternative detector. In this case, we show that an alternative learning algorithm based on a robust variant of PCA is able to substantially mitigate the effect of the poisoning.

In the third part of the book, we consider attackers who passively exploit a learner's vulnerabilities without requiring data contamination, and we conduct a theoretical investigation of two tasks for secure learning: preserving the privacy of the learner's data and preventing evasion of the learner. In Chapter 7 we explore the task of releasing a support vector machine (SVM) classifier while preserving the privacy of its training data. We adopt the strong, semantic definition of privacy known as differential privacy Dwork et al. (2006), which guarantees the privacy of even a single training datum when the adversary has full knowledge of the remaining data and the release mechanism. We present mechanisms for privately releasing SVM-like classifiers, which we prove guarantee differential privacy while attaining a guaranteed level of utility. We also explore the limits of privately releasing classifiers that approximate SVMs through lower bounds. Last, we explore the near-optimal evasion problem for the family of convex-inducing classifiers and apply our framework to a theoretical model of classifier evasion in Chapter 8. To find a classifier's blind spots an adversary can systematically issue membership queries and use the classifier's responses to glean important structural information about its boundary. To study this problem, we generalize the near-optimal evasion framework of Lowd & Meek (2005*a*), which quantifies the evasion problem in terms of query complexity. Under this model, we study the evasion of a diverse family of classifiers called the convex-inducing classifiers , and we present algorithms for evading these classifiers based on the family of ℓ_p costs. We demonstrate both positive and negative results for the feasibility of classifier evasion within the family of convex-inducing classifiers and discuss future directions for studying the evasion problem.

In the final part of the book, we explore future directions for adversarial machine learning. In Chapter 9, we discuss important themes and open questions for the field of adversarial learning in security-sensitive domains.

https://doi.org/10.1017/9781107338548.001 Published online by Cambridge University Press

This book provides machine learning practitioners with a framework for evaluating learning systems in the presence of an adversary. Under this framework we demonstrate the vulnerability of real-world learning systems, and we suggest alternative techniques that are resilient to the demonstrated exploits. Our approach and techniques provide a common foundation for secure learning, which is vital for the continued development and deployment of learning in security-sensitive or adversarial environments.

https://doi.org/10.1017/9781107338548.001 Published online by Cambridge University Press