Pattern Recognition Systems under Attack

Fabio Roli, Battista Biggio, and Giorgio Fumera

Dept. of Electrical and Electronic Engineering, University of Cagliari,
Piazza d'Armi, 09123 Cagliari, Italy
{roli,battista.biggio,fumera}@diee.unica.it
http://pralab.diee.unica.it/

Abstract. Pattern recognition systems have been increasingly used in security applications, although it is known that carefully crafted attacks can compromise their security. We advocate that simulating a proactive arms race is crucial to identify the most relevant vulnerabilities of pattern recognition systems, and to develop countermeasures in advance, thus improving system security. We summarize a framework we recently proposed for designing proactive secure pattern recognition systems and review its application to assess the security of biometric recognition systems against poisoning attacks.

Keywords: adversarial pattern recognition, biometric authentication, poisoning attacks.

1 Introduction

Pattern recognition systems have been widely deployed in security-sensitive applications like spam filtering, malware detection, and biometric authentication [10,6]. Such scenarios exhibit an intrinsic adversarial nature that fully violates data stationarity usually assumed for design of pattern recognition systems. Accordingly, a different design procedure is required to explicitly deal with the arms race existing in security settings between system designers and adversaries. We advocate that design should be based on a what-if analysis simulating a proactive arms race, for improving system security. We further argue that evaluating security properties through simulations of different, potential attack scenarios is a crucial step in this arms race for identifying the most relevant vulnerabilities and for suggesting how to potentially counter them. In Sect. 2 we briefly review an example of a reactive arms race occurred in spam filtering, and discuss differences with proactive approaches. In Sect. 3 we summarize a framework we recently proposed for designing proactive secure pattern recognition systems, and review its application to assess the security of biometric recognition systems against poisoning attacks. In Sect. 4 we try to be proactive by outlining three attacks that may emerge in the near future. Conclusions and future research lines are highlighted in Sect. 5.

J. Ruiz-Shulcloper and G. Sanniti di Baja (Eds.): CIARP 2013, Part I, LNCS 8258, pp. 1–8, 2013.

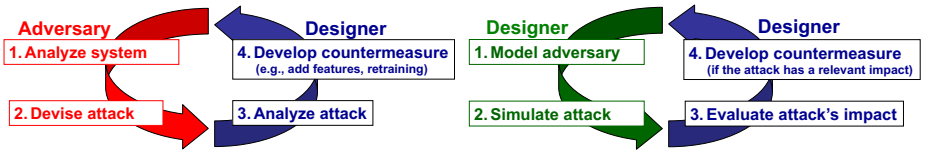

Fig. 1. A schematic representation of the reactive (left) and proactive (right) arms races incurring in security applications involving pattern recognition systems

2 The Arms Race in Pattern Recognition

As a typical example of arms race in pattern recognition we summarize in Sect. 2.1 the story of image-based spam. It also allows us to introduce the concepts of reactive and proactive security, that are explained in Sect. 2.2.

2.1 The Story of Image-Based Spam

Since the 90s, computer viruses and attack threats have evolved towards an increased level of variability and sophistication in response to an increase of the complexity and number of vulnerable attack points of modern security systems. Together with the fact that automatic tools for designing novel variants of attacks can be easily obtained and exploited by not very skilled attackers, and that a flourishing underground economy strongly motivates them, an exponential proliferation of malware and other threats has been recently observed. To cope with such a large amount of malicious data exhibiting both an increasing variability and number of never-before-seen attacks, machine-learning approaches have been increasingly adopted to complement the earlier rule-based systems (*e.g.*, signature-based systems based on string-matching techniques): the latter offer fast and lightweight filtering of most known attacks, while the former can process the remaining (unfiltered) samples and identify novel attacks.

A recent example of arms race in pattern recognition is the so-called *image-based spam* (or image spam, for short) [5,1]. This technique consists of rendering the spam message into attached images to evade the textual-based analysis performed by most of the modern anti-spam filters. Due to the massive volume of image spam sent in 2006 and 2007, researchers and companies developed countermeasures, like generating signatures to filter known spam images, or analyzing suspect images by OCR tools to extract text for standard spam detection. This started an arms race between designers and spammers. Spammers reacted by *randomly* obfuscating images with *adversarial* noise, both to to evade signature-based detection, and to make OCR-based detection ineffective. Researchers responded with (fast) approaches mainly based on machine-learning techniques using visual features extracted from images, aimed at discriminating between images attached to spam and to legitimate e-mails. Image spam volumes have since declined, although the exact cause is debatable: these countermeasures may have played a deterrent role, or image spam became too costly in terms of time to generate and bandwidth to deliver.

2.2 Reactive and Proactive Security

As highlighted by the image spam story, security problems are often cast as a long-lasting *reactive* arms race between the system designer and the adversary, in which each player attempts to achieve his goal by reacting to the changing behavior of his opponent, *i.e.*, basically *learning from the past*. This arms race can be modeled as the following cycle [6]. First, the adversary analyzes the existing pattern recognition system and manipulates data to violate system security (*e.g.*, to evade detection). For instance, a spammer may gather some knowledge of the words used by the targeted anti-spam filter to block spam, and then manipulate the textual content of spam emails accordingly; *e.g.*, words like "cheap" that are indicative of spam can be misspelled as "che4p". Second, the pattern recognition system designer reacts by analyzing the novel attack samples and updating the system consequently; *e.g.*, by retraining the classifier on the newly collected samples, and/or by adding features that can better detect the novel attacks. In the previous spam example, this amounts to retraining the filter on the newly collected spam and, thus, to adding novel words into the filter's dictionary (*e.g.*, "che4p" may be now learned as a spammy word). This *reactive* arms race continues in perpetuity as illustrated in the left plot in Fig. 1.

However, *reactive* approaches to this arms race do not anticipate the next generation of security vulnerabilities, *i.e.*, they do not attempt to *forecast future attacks*, and thus, the system potentially remains vulnerable to new attacks. Computer security guidelines accordingly advocate a *proactive* approach in which the designer should also attempt to *anticipate* the adversary's strategy by (i) identifying the most relevant threats, (ii) designing proper countermeasures for his system, when required, and (iii) repeating this process for his new design *before* deploying the pattern recognition system. This can be accomplished by modeling the adversary (based on knowledge of the adversary's goals and capabilities) and using this model to simulate attacks, to complement the reactive arms race, as shown in Fig. 1 (right). While such an approach does not account for unknown or changing aspects of the adversary, it can improve the level of security by delaying each step of the *reactive* arms race, as it should reasonably force the adversary to exert greater effort (in terms of time, skills, and resources) to find new vulnerabilities. Accordingly, pattern recognition systems that are properly designed according to the reactive and proactive security paradigms should remain useful for a longer time, with less frequent supervision or human intervention and with less severe vulnerabilities.

Although the approach of proactive security has been implicitly followed in most of previous work, it has only recently been formalized within a more general framework for the empirical evaluation of pattern classifier's security [6], which we summarize in the next section.

3 Security Evaluation of Pattern Recognition Systems

We summarize our proactive security evaluation framework [6], and its application to assess the security of adaptive biometric recognition systems.

Fig. 2. Main design steps for deploying a pattern recognition system

3.1 Proactive Security Evaluation Framework

Our framework [6] systematizes and unifies previous work. It aims at empirically evaluating the security of a pattern recognition system under design, through *simulations* of different, potential attack scenarios, *i.e.*, by a systematic *what-if analysis*. Our framework addresses the first three steps of the proactive arms race (Fig. 1, right), overcoming the shortcomings of reactive security: identifying potential attack scenarios, devising the corresponding attacks, and systematically evaluating their impact. This may also suggest countermeasures to the hypothesized attacks, whose implementation is however to be addressed separately in an application-specific manner.

Our framework focuses on attacks consisting of manipulating the data processed by a pattern recognition system to subvert the results. It does not consider attacks to the system's physical infrastructures (*e.g.*, the sensors). It exploits the taxonomy of potential attacks against learning-based pattern classifiers of [2,10], which consists of three main features: (1) the kind of *influence* of attacks on the classifier, either **causative** or **exploratory**, respectively aimed at undermine the learning and the classification phase; (2) the kind of *security violation*: either **integrity** (to gain unauthorized access to the system), **availability** (to generate many classification errors to compromise the normal system operation), or **privacy** (to obtain confidential information from the classifier); (3) the *specificity* of an attack, ranging continuously from **targeted** (focused on a few specific samples) to **indiscriminate** (*e.g.*, affecting all malicious samples).

We describe our framework for the case of supervised pattern classifiers (for different tasks like unsupervised clustering, see [8]). Their classical design steps [9], that do not take adversarial settings into account, are summarized in Fig. 2. In adversarial settings, *each* design step can be subject to attacks. To evaluate their impact, we model the adversary in terms of specific assumptions about (i) her goal, (ii) knowledge of the system, and (iii) capability to modify the data distribution by manipulating samples; this allows one to (iv) develop optimal attack strategies, and to guide the design of resilient classifiers. (i) The **adversary's goal** is based on the kind of anticipated security violation, on the attack's specificity, and of an objective function that the adversary is willing to maximize, which allows for a formal characterization of the *optimal* attack strategy. (ii) The **adversary's knowledge** ranges from no information to complete information, and it is defined for each design step of Fig. 2: the training set, the feature representation, the learning algorithm and its decision function, the learned classifier's parameters, and the feedback from the deployed classifier. Assuming *perfect knowledge* of the targeted classifier is a usual worst-case setting, which provides a lower bound on the classifier performance under attack.

A more realistic *limited knowledge* setting can also be considered; however, it would be contingent on *security through obscurity*, which strongly relies upon *secrets* that must be kept unknown to the adversary. This is complementary to the former setting, that is related to *security by design*, which advocates that systems should be designed from the ground-up to be secure, and secrets, if any, must be well-justified. Accordingly, the knowledge of at least the learning algorithm and feature representation is often assumed. (iii) The **adversary's capability** is defined according to the attack taxonomy, and can incorporate application-specific constraints. Since training and test data may follow different distributions when they are manipulated by the adversary, one should specify: whether the attack manipulates training (TR) and/or testing (TS) data (*i.e.*, the attack influence); whether and to what extent it affects the class priors for TR and TS; which and how many samples can be modified in each class; which features can be modified and how can their values be altered. To perform security evaluation according to the hypothesized attack scenario, the collected data and generated attack samples should be resampled according to the above distributions to produce suitable training and test set pairs [6]. (iv) Assumptions (i)–(iii) allow one to compute the optimal **attack strategy** (*i.e.*, the adversary model), by solving the optimization problem defined by the adversary's goal, under constraints corresponding to her knowledge and capabilities. The attack samples needed to evaluate the classifier's security are produced using the attack strategy.

The above procedure must be repeated for different levels of adversary's knowledge and/or capabilities, if necessary, and for each different hypothesized attack. In the next section we give a specific example of the application of our framework to a biometric identity recognition system.

3.2 Poisoning Attacks to Compromise Biometric Templates

The application of our framework led us to highlight a novel vulnerability of adaptive face recognition systems [7,4]. They aim at dealing with natural temporal variations of the clients' faces, by exploiting biometric data acquired over time during system operation. Template self-update is the simplest approach, inspired by semi-supervised learning techniques. It consists of periodically updating a user's template gallery using samples assigned with high confidence to the corresponding identity during operation. Although adaptation may allow a face recognition system to maintain a good performance over time, an *attacker* may exploit it to compromise the stored templates. This can be achieved by submitting a suitable sequence of fake faces to the camera while claiming the identity of a *victim* user (*poisoning* attack). The fake (or *spoofed*) faces can be obtained by printing a face image on paper [3]. This may eventually compromise the victim's templates by replacing some of them with other desired face images, that may either be sufficiently different from the victim's templates, to deny access to him; or they may include attacker's images, to allow her to impersonate the victim without eventually using any fake trait. In [7,4] we have derived optimal poisoning attacks against adaptive face verification systems, *i.e.*, attacks that *minimize* the number of fake faces to present to the camera, under

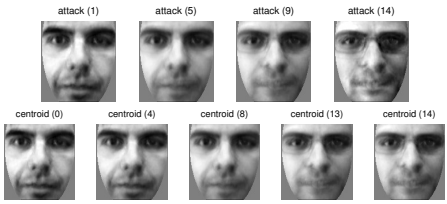

Fig. 3. Attack samples (top) and victim's centroid (bottom) for poisoning with *limited* knowledge, at different iterations

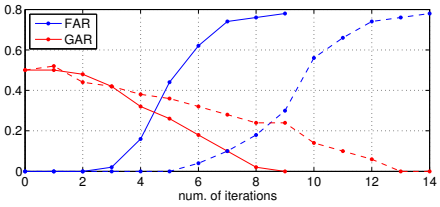

Fig. 4. FAR and GAR for poisoning with *perfect* (solid lines) and *limited* (dashed lines) knowledge, at different iterations

both perfect and limited knowledge of the attacked system. A simple example of attack is detailed in the following, according to our framework of Sect. 3.1.

We consider a face verification system based on Principal Component Analysis (PCA), where each client is authenticated by comparing the submitted face image with the stored template belonging to the claimed identity, in the feature space induced by PCA. If the similarity score exceeds a pre-defined acceptance threshold, then the claimed identity is authenticated as genuine, otherwise it is rejected as an impostor attempt. The *unique* template of each client is obtained by averaging $n = 5$ distinct face images of the same user acquired during enrollment, and it is thus referred to as *centroid*. It is self-updated during operation using face images that satisfy the update condition, *i.e.*, if the similarity score with the stored template is greater than a pre-defined update threshold, which is typically more restrictive (*i.e.*, higher) than the acceptance threshold. The centroid is updated as the average of the latest n images that have satisfied the update condition (moving average update rule with a fixed window size). **Adversary's goal:** we assume she aims to impersonate the victim without eventually using any fake trait, by replacing his template while minimizing the number of submitted fake faces (*queries*). **Adversary's knowledge:** we consider both perfect and limited knowledge. In the former case, the attacker knows the victim's templates, the feature representation, the verification and update algorithm, and their acceptance and update thresholds. In the latter, more realistic case, the attacker does not know the victim's template, but is able to get a similar enough image (*e.g.*, from social networks) such that the update condition is met and the poisoning attack can successfully start. **Adversary's capability:** she can submit a number of fake faces to get access to the victim's template gallery, *i.e.*, to a portion of the *training* data.

We refer the reader to [7,4] for the computation of the optimal attack. Figs. 3 and 4 show some experimental results for a specific attacker-victim pair. Fig. 3 shows how the victim's template is updated by the attack under limited knowledge. Fig. 4 shows the behaviour of the False Acceptance Rate (FAR, the probability of the attacker accessing the system impersonating the victim) and the Genuine Acceptance Rate (GAR, the probability of the victim correctly accessing the system), for both perfect and limited knowledge. In the perfect knowledge case less queries are required to replace the victim's template with the attacker's

desired image, which is coherent with theoretical bounds [11]. In both cases, the attacker can violate the victim's account with high probability even when the template is only partially compromised, as shown by the significantly high FAR value after half of the queries. Notably, the GAR also quickly decreases, meaning that the victim can not correctly access the system: this is a side-effect, which can be mitigated by using multiple templates per client [4].

4 Where Do Adversaries Attack Next Time?

Attacks against pattern recognition systems emerged only recently as the application and popularity of these technologies generated sufficient incentives for attackers. Nowadays, we have many reported *spoofing* attacks against biometric recognition systems based on fake biometric traits, *e.g.*, a printed picture is used to fool a facial recognition system.[1] Besides face and fingerprint recognition, the European project TABULA RASA demonstrated successful spoofing attacks against systems using speech and gait.[2] Therefore, additional biometric systems could be the next targets soon. Another little-known type of attack likely to emerge in the near future is an evasion attack against biometric video surveillance systems used to recognize targeted individuals (*e.g.*, individuals on a watch-list). To date this avenue of attack has received little attention because evading a face recognition system is still quite easy (wearing hats or glasses is often sufficient to evade it). However, the arms race to evade these pattern recognition systems has already begun as is evident in the creative CV Dazzle project that proposes new facial makeup and hair styling to evade face recognition systems.[3] Finally, another potential class of attacks that may emerge in the near future involves data clustering, one of the key technologies for the commercial exploitation of massive volumes of both structured and unstructured data (now called *big data*). Clustering algorithms have been increasingly adopted in security applications to spot dangerous or illicit activities. However, they have not been originally devised to deal with deliberate attack attempts that may aim to subvert the clustering process itself. We have recently demonstrated that an attacker may significantly subvert the whole clustering process by adding a relatively small percentage of attack samples to the input data [8]. The market trend of *big data* makes very likely that clustering algorithms used in commercial and security applications will be soon the target of attacks.

5 Conclusions and Future Work

In this work we pointed out some of the issues related to the adoption of pattern recognition systems in security-sensitive settings, and advocated a proactive approach to security evaluation that can be exploited complementarily to the

[1] An example of a spoofing attack: `http://www.youtube.com/watch?v=2fKGXSgOFYc`
[2] `http://www.tabularasa-euproject.org`
[3] `http://cvdazzle.com`

well-known reactive paradigm to understand their security guarantees. Thinking proactively, we also discussed some novel potential sources of vulnerabilities, such as data clustering algorithms. For the same reason, one may also think of attackers that combine carefully crafted attacks against specific system components (*e.g.*, data clustering, feature selection, and classifier training) to develop more complex, stealthy attacks. These *multiple* attacks may be indeed more difficult to spot as they may only slightly affect each of the system's components involved, although eventually compromising the overall system security to a large extent. Finally, although the proactive security evaluation of pattern recognition systems advocated in this paper may suggest specific countermeasures, designing general-purpose *secure* classifiers remains an open problem that should be specifically addressed in the future.

Acknowledgments. This work has been partly supported by the project "Security of pattern recognition systems in future internet" (CRP-18293) funded by Regione Autonoma della Sardegna. The opinions expressed in this paper are solely those of the authors and do not necessarily reflect the opinions of any sponsor.

References

1. Attar, A., Rad, R.M., Atani, R.E.: A survey of image spamming and filtering techniques. Artif. Intell. Rev. 40(1), 71–105 (2013)
2. Barreno, M., Nelson, B., Sears, R., Joseph, A.D., Tygar, J.D.: Can machine learning be secure? In: Proc. of the 2006 ACM Symp. on Information, Computer and Comm. Sec., pp. 16–25. ACM, NY (2006)
3. Biggio, B., Akhtar, Z., Fumera, G., Marcialis, G.L., Roli, F.: Security evaluation of biometric authentication systems under real spoofing attacks. IET Biometrics 1(1), 11–24 (2012)
4. Biggio, B., Didaci, L., Fumera, G., Roli, F.: Poisoning attacks to compromise face templates. In: 6th IAPR Int'l Conf. on Biometrics, pp. 1–7 (2013)
5. Biggio, B., Fumera, G., Pillai, I., Roli, F.: A survey and experimental evaluation of image spam filtering techniques. Pattern Rec. Letters 32(10), 1436–1446 (2011)
6. Biggio, B., Fumera, G., Roli, F.: Security evaluation of pattern classifiers under attack. IEEE Trans. on Knowledge and Data Engineering 99(preprints), 1 (2013)
7. Biggio, B., Fumera, G., Roli, F., Didaci, L.: Poisoning adaptive biometric systems. In: Gimel'farb, G., Hancock, E., Imiya, A., Kuijper, A., Kudo, M., Omachi, S., Windeatt, T., Yamada, K. (eds.) SSPR&SPR 2012. LNCS, vol. 7626, pp. 417–425. Springer, Heidelberg (2012)
8. Biggio, B., Pillai, I., Rota Bulò, S., Ariu, D., Pelillo, M., Roli, F.: Is data clustering in adversarial settings secure? In: Proc. of the 2013 Artificial Intelligence and Security Workshop (2013)
9. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley-Interscience Publication (2000)
10. Huang, L., Joseph, A.D., Nelson, B., Rubinstein, B., Tygar, J.D.: Adversarial machine learning. In: 4th ACM Workshop on Artificial Intelligence and Security (AISec 2011), Chicago, IL, USA, pp. 43–57 (2011)
11. Kloft, M., Laskov, P.: Online anomaly detection under adversarial impact. In: Proc. of the 13th Int'l Conf. on Artificial Intelligence and Statistics, pp. 405–412 (2010)