Adversarial Biometric Recognition: A Review on Biometric System Security from the Adversarial Machine Learning Perspective

Battista Biggio, Giorgio Fumera, Paolo Russu, Luca Didaci, and Fabio Roli

Abstract

In this article, we review previous work on biometric security under a recent framework proposed in the field of adversarial machine learning. This allows us to highlight novel insights on the security of biometric systems when operating in the presence of intelligent and adaptive attackers that manipulate data to compromise normal system operation. We show how this framework enables the categorization of known and novel vulnerabilities of biometric recognition systems, along with the corresponding attacks, countermeasures and defense mechanisms. We report two application examples, respectively showing how to fabricate a more effective face spoofing attack, and how to counter an attack that exploits an unknown vulnerability of an adaptive face recognition system to compromise its face templates.

INTRODUCTION

Adversarial machine learning is a novel research field that was born in response to the increasing use of pattern recognition and machine learning techniques, including signal processing ones, in security-related applications, like biometric identity recognition, spam and malware detection. In these applications, intelligent and adaptive *adversaries* are interested in subverting system operation; *e.g.*, non-authorized users may aim to gain access to a resource secured by a biometric identity recognition system. Despite pattern recognition and machine learning algorithms have enabled the development of more effective recognition systems, they have not been originally designed to operate in adversarial settings. In particular, their underlying assumption of data *stationarity* (*i.e.*, that training and testing data follow the same distribution) is likely to be violated in adversarial environments. As a consequence, these algorithms can

Department of Electrical and Electronic Engineering, University of Cagliari, Piazza d'Armi, 09123 Cagliari, Italy

B. Biggio: e-mail battista.biggio@diee.unica.it, phone +39 070 675 5776

G. Fumera: e-mail fumera@diee.unica.it, phone +39 070 675 5754

P. Russu: e-mail paolo.russu@diee.unica.it, phone +39 070 675 5776

L. Didaci: e-mail luca.didaci@diee.unica.it, phone +39 070 675 5776

F. Roli (corresponding author): e-mail roli@diee.unica.it, phone +39 070 675 5779, fax (shared) +39 070 675 5782

introduce additional, specific vulnerabilities, that can be exploited by carefully-crafted attacks to cause different security violations, including denial of service and missed detection of intrusive attempts. This may eventually compromise the whole system security. Even if countermeasures and novel algorithms have been proposed to improve security against these sophisticated attacks, they will not stop adversaries from developing novel ways of misleading such defense systems, engendering a long-lasting *arms race*.

To date, research efforts in adversarial machine learning have focused on identifying different kinds of potential attacks against machine learning and pattern recognition algorithms, and developing the corresponding countermeasures to improve robustness in adversarial settings [1]–[4]. According to the *security-by-design* paradigm, ongoing work is also addressing the issue of extending learning theory and methods to explicitly account for the presence of malicious adversaries that can undermine algorithm operation; *e.g.*, in [2], [3] traditional performance evaluation methods have been extended to allow for a systematic evaluation of the security of pattern classifiers in adversarial settings, providing a better understanding of the system performance both in the absence and in the presence of well-crafted attacks. The relevance of these issues is also witnessed by an increasing number of publications and events, *e.g.*, the NIPS Workshop on Machine Learning in Adversarial Environments for Computer Security [5], and the more recent Dagstuhl Perspectives Workshop on Machine Learning Methods for Computer Security [6].

Biometric identity recognition is a clear example of a widespread and still growing application field in which security is a key issue, and pattern recognition techniques play a major role. Different vulnerabilities of biometric systems, specific attacks that can exploit them, and corresponding countermeasures have been analyzed in the literature [7], [8] and in research projects. For instance, the recent EU FP7 Tabula Rasa project has carried out an extensive analysis on spoofing attacks (*i.e.*, attacks involving the submission of a fake biometric trait, like a gummy finger, to impersonate an authorized user), and on the development of possible countermeasures, like liveness detection techniques. Moreover, several approaches for the analysis and assessment of biometric systems security have been proposed.[1] However, all existing efforts disregarded the potential, specific vulnerabilities introduced by pattern recognition algorithms used in biometric systems, and thus the investigation of the corresponding attacks and countermeasures. We argue that looking at biometric system security from the perspective of adversarial machine learning not only provides an original categorization of existing attacks against such systems, but it also allows us to consider more sophisticated attacks targeting vulnerabilities of the learning algorithms used in these

[1]See, *e.g.*, the ISO/IEC 19792:2009 implementation specifics for the security evaluation of biometrics, and the NIST Common Criteria For Information Technology Security Evaluation.

systems, along with the countermeasures already proposed in the field of adversarial machine learning.

Based on the above motivations, in the following we first give a concise overview of adversarial machine learning, to introduce kindly the readers to this recent research field; we use popular attacks against biometric systems, such as spoofing attacks, as running examples to make our explanation clearer. We then review the security of biometric identity recognition systems by showing how recent theoretical results and systematization efforts from this field enable: (i) the definition of a more complete taxonomy of attacks against biometric systems, based on a formal attacker's model explicitly accounting for her knowledge and capability, which allows one to identify novel attack scenarios associated to specific vulnerabilities of machine learning and pattern recognition algorithms, besides encompassing known attacks; (ii) the design of the corresponding countermeasures, building on solutions proposed in adversarial machine learning, which can give rise to the design of novel, *secure-by-design* algorithms capable of improving adversarial biometric identity recognition. We finally discuss two application examples of the possible aforementioned achievements. We first show how a skilled attacker may fabricate more effective face spoofing attacks, and then highlight a new vulnerability of adaptive biometric systems, devising the corresponding attack and a possible countermeasure.

The main goal of this article is to provide the readers of this magazine, and researchers in biometrics, a gentle introduction to adversarial machine learning, and a well-structured review of the state of the art on biometric security, in light of the most recent findings in the area of adversarial machine learning.

ADVERSARIAL MACHINE LEARNING: AN OVERVIEW

During the last decades, the increasing variability and sophistication of attack threats, in response to the growing complexity and amount of vulnerable attack points in security systems, has favored the adoption of machine learning and pattern recognition techniques to timely detect variants of known and never-before-seen attacks. These techniques can however exhibit intrinsic vulnerabilities that can be exploited by skilled attackers, perpetuating their arms race against system designers. Adversarial machine learning aims at countering this phenomenon by focusing on vulnerabilities of learning algorithms. It attempts to *anticipate* the adversary's strategy by identifying novel threats and devising the corresponding countermeasures *before* system deployment. In practice, it follows a *proactive* rather than a *reactive* approach. The first step towards the above goal has been the proposal of a taxonomy categorizing attacks against learning algorithms along three axes [1], [2]: (i) the *attack influence*, which can be **exploratory**, if the adversary can only manipulate the testing data, or **causative**, if she can modify also the training data; (ii) the *attack specificity*, which ranges from **targeted** to **indiscriminate**, depending on whether

the classification of a set of specific samples or any of them is affected by the attack; (iii) the *security violation*, which can be an **integrity** violation, if the adversary is allowed to access a restricted service or resource (*e.g.*, an impostor gaining access to a genuine client's account [3], [9]); an **availability** violation, if legitimate users' are denied access or normal system operation is compromised (*e.g.*, misclassifying legitimate emails as spam); and a **privacy** violation, if the adversary is able to exploit confidential information about the system (*e.g.*, the clients' templates in a biometric recognition system [10]–[12]).

To date, several vulnerabilities and attacks against different learning algorithms (*e.g.*, support vector machines and neural networks) have been investigated, along with the proposal of possible countermeasures [13]–[15]. We will summarize the main existing countermeasures in the remainder of this paper, discussing how they can be exploited to improve the security of machine learning algorithms used in biometric systems. In particular, to anticipate the adversary's strategy, existing work in adversarial learning *simulates* attacks, based on more or less explicit *models* of the adversary. We recently formalized this approach within a general framework, proposing a formal model to characterize the adversary's behavior [3], [4]. We summarize our model below, and exploit it in the remainder of this article to characterize and understand security of biometric systems under an adversarial machine learning perspective.

Our model generalizes and encompasses other models proposed in the area of adversarial machine learning [1], [2], making explicit assumptions on the attacker's *goal*, *knowledge* of the targeted system, and *capabilities* of manipulating the input data or the system's components. The **goal** has to be defined according to the desired *security violation* and *attack specificity*. **Knowledge** of system components (*e.g.*, the kind of decision function and its parameters, or how a component operates) can be *perfect* or *limited*, and feedback on the classifier's decisions can also be exploited [13], [16], [17]. The **capability** is defined as the *attack influence*, based on how the adversary can affect training and testing data (*e.g.*, which features can be manipulated and how, according to application-specific constraints). An **attack strategy** can be then defined based on the previous elements, to implement the attack. In formal terms, we assume the attacker's goal to be quantified by a function $g(a, \boldsymbol{\theta})$ measuring the extent to which an attack strategy a from a feasible strategy set $\mathcal{A}$ fulfills the attacker's goal. The feasible set $\mathcal{A}$ has to be defined according to the attacker's capability of manipulating the input data and system components, while the attacker's knowledge is encoded by the parameter vector $\boldsymbol{\theta} \in \Theta$. Under this setting, the optimal attack strategy corresponds to the solution of the following optimization problem:

$$\max_{a \in \mathcal{A}} g(a; \boldsymbol{\theta}) \,. \tag{1}$$

Although this formulation may seem rather abstract at this stage, it enables us to consider trivial and sophisticated attacks under a consistent view, as shown in the remainder of this article.

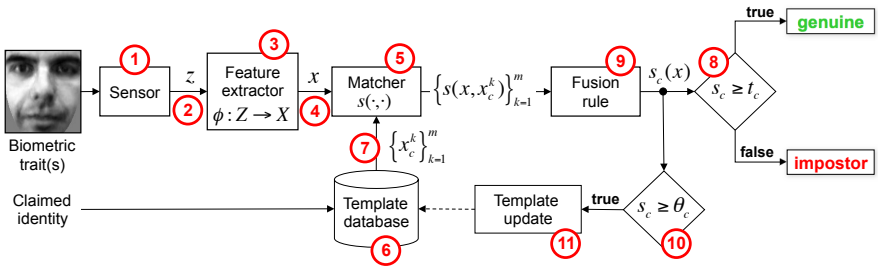


Fig. 1: Architecture of a biometric verification system and corresponding attack points, highlighted with red circled numbers. During verification, the image $\boldsymbol{z} \in \mathcal{Z}$ (*e.g.*, a face image) acquired by the sensor is processed by a feature extractor $\phi : \mathcal{Z} \mapsto \mathcal{X}$ to obtain a compact representation $\boldsymbol{x} \in \mathcal{X}$ (*e.g.*, a graph). The templates $\{\boldsymbol{x}_c^k\}_{k=1}^m$ of the claimed identity c are retrieved from the template database, and compared to $\boldsymbol{x}$ using a matching algorithm $s : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$. The resulting scores $\{s(\boldsymbol{x}, \boldsymbol{x}_c^k)\}_{k=1}^m$ are combined by a fusion rule, producing an aggregated score $s_c(\boldsymbol{x})$ that expresses the degree to which $\boldsymbol{x}$ is likely to belong to c. The score $s_c(\boldsymbol{x})$ is then compared with a decision threshold t_c to decide whether the claim is genuine or impostor. If template self-update is implemented, and $s_c(\boldsymbol{x})$ is not lower than a self-update threshold θ_c, one of the templates in $\{\boldsymbol{x}_c^k\}_{k=1}^m$ is updated depending on $\boldsymbol{x}$, according to a given policy.

BIOMETRIC RECOGNITION SYSTEMS UNDER ATTACK

Biometric recognition systems operate either in *enrollment* or in *recognition* mode [8]. During **enrollment**, each client provides his biometric traits and identity, in the presence of a human supervisor. A set of *reference templates* for each client is then stored in the *template database* along with the corresponding identity. During **recognition**, the biometric system is expected to recognize a previously-enrolled client by comparing the submitted traits with those stored in the template database. Biometric systems may operate either in an *verification* or in a *identification* setting. In *verification* settings, biometric systems are often used to control access to protected resources, including confidential information or services. A user aiming to access them has to provide his/her biometric trait and claim an identity. The system then verifies whether the claim is genuine (*i.e.*, the user's identity is the claimed one) or not (*i.e.*, the user is an impostor trying to impersonate another client), and allows access only in the former case. This procedure is illustrated in Fig. 1, which is general enough to also account for multibiometric systems; in this case, the fusion rule s_c should aggregate the matching scores coming from *all* biometric traits. In *identification* settings, instead, no identity claim is made: a user provides only the requested biometric trait $\boldsymbol{x}$, and the system is expected to correctly recognize the corresponding identity among those in the template database, by matching $\boldsymbol{x}$ against all the known clients' templates; the corresponding scores $s_c(\boldsymbol{x})$ are then sorted in descending order to provide a list of the most likely candidate identities.

To account for natural changes of biometric traits over time (*i.e.*, biometric *aging*), and changes in the environmental or acquisition conditions during verification, *adaptive* biometric systems have been proposed. They enable update of the stored templates automatically during verification [18], [19]. A popular technique is template self-update: if the submitted trait is sufficiently similar to the reference

TABLE I: Categorization of attacks and countermeasures for biometric systems. For each attack technique, we also report the targeted component (attack location) and the attack point(s), according to Fig. 1.

Attack Technique	Attack Location	Attack Point(s)	Defense
Spoofing	Sensor	1	Liveness Detection, Multibiometrics (Secure Fusion)
Replay	Interfaces / Channels	2, 4, 7	Encrypted Channel, Timestamp, Challenge-Response, Physical Isolation
Hill-Climbing	Interfaces / Channels	2, 4	Encrypted Channel, Timestamp, Challenge-Response, Physical Isolation, Score Quantization
Malware Infection	Modules / Algorithms	3, 5, 8-11	Secure Code, Specialized Hardware, Algorithmic Integrity
Template Theft, Substitution, and Deletion	Template Database	6	Template Encryption, Cancelable / Revokable Templates

templates of the claimed identity, one of such templates is updated by exploiting information coming from the submitted trait, according to a given policy. A simple update policy is called *nearest-out self-update*, as it replaces the most similar template to the submitted trait with the latter [20], [21].

The Attack Surface

Previous work has identified the main attack points and vulnerabilities of biometric recognition systems, along with the corresponding attacks [7], [8]. First, any system is subject to **intrinsic failures** not produced by adversarial attempts, *i.e.*, rejected genuine claims and accepted *zero-effort* impostors (*i.e.*, impostors that do not exert any special effort to intrude). Besides this, a number of **adversarial attacks** have been also considered in early work, leading to the identification of eight potentially vulnerable attack points highlighted by the red circled numbers 1–8 in Fig. 1 [7]. We additionally consider here points 9-11: they correspond to vulnerabilities of *adaptive* biometric systems that update clients' templates during operation, which we recently exploited to implement a *template poisoning* attack [20], [21]. The set of all attack points defines the *attack surface* of a biometric recognition system. The corresponding attacks can be categorized into four main groups, according to the targeted system component [8]: attacks to the sensor (point 1), to interfaces and channels connecting different modules (points 2, 4, 7), to processing modules and algorithms (points 3, 5, 8–11), and to the template database (point 6). We discuss them below, along with the corresponding countermeasures proposed so far, that are also summarized in Table I. It is also worth mentioning here a special category of attacks, known as *insider attacks*, where the attacker is colluded with a system administrator or exercises coercion to escalate privileges [8].

Spoofing attacks consist of fabricating a fake biometric trait to impersonate an enrolled client. They target the sensor (point 1), so they are also referred to as *direct attacks*. Current defenses are based on *liveness detection* methods, which aim to verify whether the submitted trait is "alive" or "fake" by looking at specific patterns (*e.g.*, perspiration patterns during fingerprint acquisition, or eye blinking during face

verification). Multibiometric systems have been also proposed as a defense; however, to avoid spoofing them by only using a single fake trait, the matching scores coming from the different traits should be properly combined, using a *secure* score-level fusion rule [9], [22].

Replay attacks can be staged at interfaces between modules by replaying a stolen image of the biometric trait of the targeted client to the feature extractor (point 2), or directly the corresponding feature values to the matcher (point 4). An attacker may even replay a signal to replace the features of a given template of the claimed identity (point 7). This attack can be clearly staged if the corresponding communication channels are *insecure*, but also over encrypted channels, as the encrypted signal can be stolen and replayed into the channel directly. This can be avoided by encrypting a timestamp into the signal, or using challenge-response mechanisms. Another possible countermeasure is *physical isolation*, to avoid sending data over insecure channels (*e.g.*, the Internet) subject to man-in-the-middle attacks. A popular example of physical isolation is the use of smart cards performing match-on-card operations. However, this technique has its own disadvantages, including limitations in terms of computational resources and memory, and the fact that the user should always use the smart card to be authenticated [8].

Hill-climbing attacks, similarly to replay ones, affect insecure communication channels between modules, and, in particular, point 2 and 4 in Fig. 1. Their goal is to reconstruct a template image by iteratively sending a bunch of slightly perturbed images to the feature extractor (point 2), or their features to the matcher (point 4), and retaining the one that maximizes the matching score $s_c(\boldsymbol{x})$, where $\boldsymbol{x}$ is the current image (or set of features) submitted by the attacker. In practice, it is a gradient-ascent technique that approximates the gradient of $s_c(\boldsymbol{x})$ numerically. In this case, the attacker is assumed to be able to observe $s_c(\boldsymbol{x})$ for any queried image, which may only be feasible if the system provides (or leaks) such information. In fact, besides the aforementioned channel protection schemes, an additional defense mechanism consists of quantizing the matching score to provide less accurate information to the attacker. However, attacks based on more sophisticated black-box optimization techniques, suited to quantized objective functions, can also be considered to make these countermeasures ineffective [10].

Malware Infection. The algorithmic implementations of the software modules (points 3, 5, 8–11) may exhibit vulnerabilities that can be exploited by a skilled attacker through well-known hacking techniques (*e.g.*, buffer overflow), to install malicious software, *i.e.*, *malware*, including worms, trojan horses, *etc.* This threat can be avoided or mitigated by exploiting well-known programming practices, like secure code programming, or using specialized hardware to perform some critical operations [8]. A secure programming practice is to check algorithmic integrity, *i.e.*, that each algorithm and function correctly handles any input parameter and never shows any unexpected behavior. For instance, if the matching

algorithm expects a vector $\boldsymbol{x} \in \mathbb{R}^d$ as input, and instead receives an input with a different format, is it going to crash or provide anyway an output? In the latter case, how is such an output handled by the subsequent modules? Does it lead to accepting by error the given claim as genuine or not?

Template Theft, Substitution, and Deletion attacks target the template database (point 6). If templates are not protected properly, one may be able to steal them, and use them to create a spoof (*i.e.*, a fake template), to perform a replay attack, or to impersonate the targeted client on a different system and perform other operations, *e.g.*, searching on protected databases (function creep) [8]. Another possibility is to replace a template to impersonate a client without requiring any sophisticated attack as spoofing or replay; *e.g.*, an attacker may add his own fingerprint template to the set of templates belonging to another client. Additionally, templates of a given client can be deleted to cause a denial of service, *i.e.*, to avoid the targeted client to be recognized successfully. Countermeasures include template encryption, and also the use of cancelable / revokable templates, which can be used only on a specific system, and reissued if stolen. The idea is to encode the templates using a key or pin code that can be changed to re-enroll the user and create a novel, different encrypted template [8].

ADVERSARIAL BIOMETRIC RECOGNITION

We analyze here biometric system security in terms of our previously-discussed framework, by making assumptions on the adversary's goal, knowledge, capability, and attack strategy, suited to biometric applications. Our aim is threefold: (i) to provide a well-structured categorization of the vulnerabilities of biometric systems and of the corresponding attacks, also through the definition of different, pertinent attack scenarios; (ii) to provide a formal characterization of existing attacks within our framework, and envision more sophisticated and effective attack strategies; and (iii) to identify suitable countermeasures and defenses inspired by previous work on adversarial machine learning.

Adversary's Goal. It is defined in terms of *security violation* and *attack specificity*. Biometric system security can be violated by an attacker that aims at impersonating a genuine user (**integrity** violation), at compromising the template galleries of genuine users to deny them access to the system, causing a denial of service (**availability** violation), or at violating the privacy of genuine users, *e.g.*, by inferring their templates through a hill-climbing attack (**privacy** violation). The attack specificity can be **targeted**, if the attack targets a specific set of clients, or **indiscriminate**, if any client may be affected.

Adversary's Knowledge. We define it by leveraging on the definition of the attack surface of biometric systems given in the previous section, by making specific assumptions on what the attacker knows of the system components and how they work. According to Fig. 1 and Table I, the attacker may know: (i)

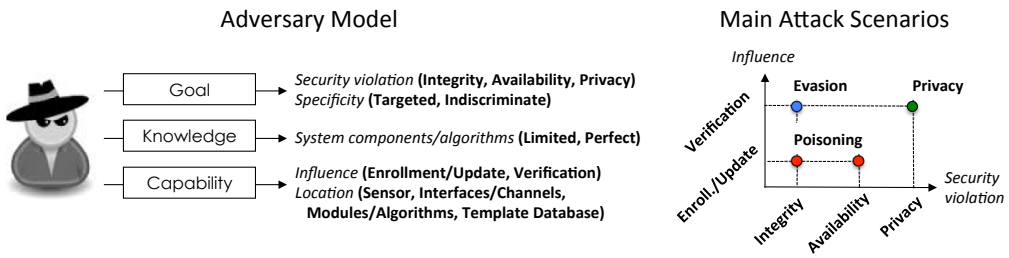


Fig. 2: A conceptual representation of the adversary model and of the main attack scenarios (given in terms of the corresponding security violation and attack specificity) according to our framework.

the kind of **sensor** used (point 1), *e.g.*, an optical or capacitive fingerprint sensor; (ii) which **interfaces / channels** are used to implement connections (points 2, 4, 7), *e.g.*, if an insecure channel over the Internet is used to send the acquired images to the feature extractor (point 2); (iii) how the **modules / algorithms** work, and whether they are vulnerable or not (points 3, 5, 8-11), in particular, the feature mapping ϕ (point 3), the matching algorithm s (point 5), the decision threshold t_c (point 8), the fusion rule s_c (point 9), and, if template update is implemented, the self-update threshold θ_c (point 10) and the template update policy (point 11); (iv) some of the templates stored in the **template database** (point 6). The attacker may also be able to collect images of the same biometric traits using other techniques; *e.g.*, acquiring latent fingerprints, or collecting face images of the targeted clients from social networks. From a machine learning perspective, this amounts to having different levels of knowledge of the classifier's training data. In practice, it is worth noting that attackers typically have limited knowledge of the sensors and algorithms used, of the users' templates, and of any other system components (*e.g.*, communication channels, template encryption schemes, *etc.*). Several previous works have however considered vulnerabilities of biometric systems without clearly pointing out the underlying assumptions on the adversary's knowledge required to perform the corresponding attack. Under our framework, such assumptions become clearly explicit.

Adversary's Capability. It can be also defined in terms of the *attack location*: (i) the sensor (point 1); (ii) interfaces / channels (2,4,7); (iii) the internals or even the output of modules and algorithms (3,5,8–11), *e.g.*, through malware infection attacks; and (iv) the template database (6). In addition, one has to define the *attack influence*, *i.e.*, the capability of manipulating the input data (*e.g.*, using fake biometric traits), and how such data may be used to update the system (*e.g.*, in adaptive biometric systems the attacker can produce spoofing attacks that can subsequently poison the clients' templates [20], [21]). Accordingly, the attack can influence only **verification**, or also **enrollment / update**.

Attack Strategy. According to the adversary's goal (generally expressed in terms of an objective function $g(a;\boldsymbol{\theta})$), knowledge (given in terms of the parameter vector $\boldsymbol{\theta} \in \Theta$) and capability (which defines the feasible set of attack strategies $a \in \mathcal{A}$), an optimal attack strategy can be defined to implement the

TABLE II: Examples of categorization of previous work on biometric security according to the three main attack scenarios defined in adversarial machine learning: evasion, poisoning, and privacy attacks.

	Goal		**Knowledge**	**Capability**		**Attack Strategy**
	Violation	*Specificity*		*Influence*	*Location*	
Evasion attacks						
Matsumoto *et al.* [23], Rodrigues *et al.* [9], Johnson *et al.* [24]	Integrity	Targeted, Indiscriminate	Perfect (consensual fake)	Verification	Sensor	Spoofing
Poisoning attacks						
Biggio *et al.* [20], [21]	Integrity, Availability	Targeted, Indiscriminate	Perfect, Limited (unknown templates)	Enrollment / Update	Sensor	Spoofing (face)
Privacy attacks						
Adler [10], Galbally *et al.* [11], Martinez *et al.* [12]	Privacy	Targeted, Indiscriminate	Limited (unknown templates)	Verification	Matcher	Hill-climbing

attack, as explained before (see Eq. 1). For instance, assume that the attacker aims to impersonate an enrolled client (integrity targeted attack), and she is only able to acquire a latent fingerprint of the client, without having any other knowledge of the system components and algorithms. Then, the corresponding optimal attack strategy amounts to fabricating a fake fingerprint that is as similar as possible to the latent one, and using it to perform a spoofing attack. In this case, $g(a;\boldsymbol{\theta})$ can be regarded as a measure of the similarity between the fake and the latent fingerprint, as $\boldsymbol{\theta}$ only contains information related to the latent fingerprint, and a corresponds to the fake fingerprint. A more skilled attacker may however also know the matching algorithm s and the fusion rule s_c used by the system, and may be able to collect more than a single fingerprint image of the targeted client. As an application example of our framework, we will show that, under this setting, more sophisticated and effective spoofing attacks can be fabricated. As another application example, we will also consider a poisoning attack against an adaptive face verification system, and propose a novel countermeasure based on the *sanitization* of the client's templates.

In the following, we define a set of representative attack scenarios to categorize known attacks according to our framework. The framework and the considered attack scenarios are also represented in Fig. 2.

Categorization of Biometric Attack Scenarios

Previous work has categorized attacks to biometric systems and countermeasures simply in terms of the attack points of Fig. 1 (*e.g.*, spoofing attacks to the sensor, countered by liveness detection techniques, and attacks to a compromised channel, countered by channel encryption). Instead, looking at them from the broader perspective opened by our framework allows us to identify three main *attack scenarios*, described in the following, in which the attacks and countermeasures discussed in the previous sections play different roles. A few examples of known attack and defenses are categorized in Table II in terms of these attack scenarios. This also opens the way both to identify novel, more sophisticated attacks against

biometric systems, and to adapt the corresponding countermeasures from the adversarial learning field.

Evasion. The **goal** of this attack scenario is to impersonate a client (integrity, targeted/indiscriminate attack). To this end, **knowledge** of the client's biometric trait is required, *e.g.*, to create a fake trait or to carry out a replay attack. Attacks exploiting perfect knowledge of the targeted client's biometric trait include the so-called consensual method (in which the targeted client voluntarily provides the required biometric trait to the attacker), and template stealing. Conversely, exploiting a latent fingerprint is an example of limited knowledge, since the attacker may only partially know or observe the required biometric trait. A limited knowledge about the rest of the biometric system can also be sufficient. In this scenario the **capability** of the attacker consists of manipulating data during the verification step, whereas no influence on the enrollment/update step is assumed (in particular, she can not access the template database). Most frequently, the attack strategy corresponding to an evasion attack consists of submitting a fake trait (spoof) to the sensor (point 1), or of replaying the acquired image (point 2) into the system. In rarer cases (disregarded here), the biometric system can be infected by malware, potentially allowing the attacker to arbitrarily manipulate the functionality or the output of any system component.

Poisoning. This non-trivial attack scenario has been originally defined in the context of adaptive biometric systems in our previous work [20], [21], inspired by our adversarial learning framework, and by work on poisoning learning algorithms [1]–[4]. The **goal** of poisoning attacks can be either an integrity or availability security violation; it can be either targeted to a specific client, or indiscriminate (see below). The adversary's **knowledge** can be perfect or limited, depending on whether each of the system's components is known exactly to the attacker. More precisely, the attacker may have perfect (or limited) knowledge of each of the components discussed in Fig. 1, including the targeted clients' templates, the matching algorithm, the template update algorithm, and the decision and update thresholds. The attacker's **capability** consists of modifying the template database, either by directly manipulating it (*e.g.*, through malware infection), or, more realistically, by submitting fake traits that are erroneously used to update the template gallery of a given client. In terms of security violation, an integrity violation thus amounts to replacing a victim's template with an attacker's template, or to adding an attacker's template in the victim's gallery. This indeed allows the attacker to impersonate the victim without using any further spoofing or replay attack, but directly using her own biometric trait. The goal of an availability violation is to cause a denial of service, instead, by replacing or compromising the majority of templates in the victim's gallery. This will indeed deny the victim access to the system. Under this setting, the **attack strategy** amounts to compromising the template gallery either by introducing an attacker's template in the victim's gallery (*i.e.*, integrity violation), or by compromising the maximum number of victim's templates

(*i.e.*, availability violation). If the template database can not be compromised directly, the attacker can produce a well-crafted sequence of fake traits to gradually drift the victim's template gallery towards the desired set of templates, while minimizing the number of fake traits required to complete the attack. An example of such an attack will be given below.

Privacy. In this case, the **goal** is to retrieve confidential information (*i.e.*, one or more templates) about either a given set of clients (targeted attack) or about any client (indiscriminate attack). This is typically a preliminary step before performing another kind of attack (evasion or poisoning), when no simpler way to retrieve information on the victims' templates exists (*e.g.*, acquiring a face image through a social network, or a latent fingerprint). To this end, the attacker can gain **knowledge** from the system's feedback, *e.g.*, the outcome of the verification decision (either accept or reject), or the score value $s_c(\boldsymbol{x})$ (as in hill-climbing attacks). In common settings (*i.e.*, disregarding cases like malware infection), the **capability** consists of sending a number of query images through a remote channel and observing the available feedback; *e.g.*, if the sensor and the matcher are remotely operating, and interconnected through the Internet, an attacker may perform a man-in-the-middle attack, and send replayed images through the channel. It is thus clear that the **attack strategy** in this case corresponds to an hill-climbing attack.

SECURE-BY-DESIGN BIOMETRIC SYSTEMS

The considered adversary model can be exploited not only to provide a different categorization of known defense mechanisms for biometric recognition systems, but also to identify novel countermeasures among those proposed in adversarial learning, that can help countering attacks against machine learning and pattern recognition algorithms used in biometric systems. We discuss them below, with reference to the three aforementioned attack scenarios.

Countering Evasion. In this scenario, the main attack strategies involve spoofing and replay. As reported in Table I, the pertinent defenses thus are: liveness detection, multibiometric systems with secure score-level fusion rules, encrypted channels and timestamp / challenge-response schemes, and physical isolation (*e.g.*, match-on-smart-cards). Novel defense mechanisms can also be devised, inspired by the adversarial machine learning field. In particular, to counter evasion attacks, one can consider *secure learning* techniques. They consist of modifying existing learning algorithms, and developing novel ones, that explicitly take into account a specific kind of adversarial data manipulation. They follow the paradigm of security by design, which advocates that a system should be designed from the ground up to be secure. In the context of biometric systems, secure learning techniques can be exploited to design trainable score-level fusion rules, such as those based on game theory, or on the framework of learning with invariances [14],

[25]–[27]. Investigating this issue would be an interesting research direction for future work.

Countering Poisoning. Spoofing and replay are the main attack strategies also under this scenario and, thus, the same defenses listed in Table I can be also exploited in this case. In addition, other countermeasures can be considered, among those proposed in adversarial learning, to improve the *security of the training phase* in the presence of poisoning, which may occur when the system is retrained on data collected during operation [15], [28]. These include *secure learning* (see above) and *data sanitization*. In a biometric system, the latter consists of detecting outlying template updates that may compromise the template gallery of a given client, *e.g.*, by adding an impostor's template to the targeted client's gallery, or by replacing some of the client's templates. We will give a concrete example of a novel defense based on *template sanitization* in the next section. We point out that these additional defenses can be considered complementary to those listed in Table I, like liveness detection and channel encryption.

Preserving Privacy. Known defenses that can be exploited against attacks targeting the template database are mainly based on template encryption schemes [8] (Table I). Score quantization has been also proposed to counter hill-climbing attacks, but it has already been shown to be ineffective [10]. Moreover, attacks proposed in adversarial learning have already been capable of reverse engineering the classifier by only exploiting only feedback on its decisions [16], [17]; thus, even by only looking at genuine or impostor classifications, an attacker may be able to successfully perform an hill-climbing attack. Among the proposed countermeasures that have not yet been considered for biometric systems, it would be worth investigating in future work the ones based on *randomization* and *disinformation*. They follow the paradigm of *security by obscurity*, aiming to improve system security by hiding information to the attacker. They have been suggested in adversarial learning to counter reverse-engineering attacks. This can be achieved by denying access to the actual classifier or training data, and randomizing the classifier's output to give imperfect feedback to the attacker [1]–[4], [29].

APPLICATION EXAMPLES

We consider here two application examples of our framework, respectively related to the development of sophisticated *spoofing* and *poisoning* attacks against face verification systems.

A. Improved Face Spoofing from Multiple Faces

Let us assume we are given a face verification system that authenticates clients by matching the acquired face image against the template gallery of the claimed identity (consisting of n images acquired during enrollment), and then thresholding the corresponding average score. According to the architecture depicted in Fig. 1, we assume that our system maps the submitted face image $\boldsymbol{z} \in \mathcal{Z}$ onto a reduced

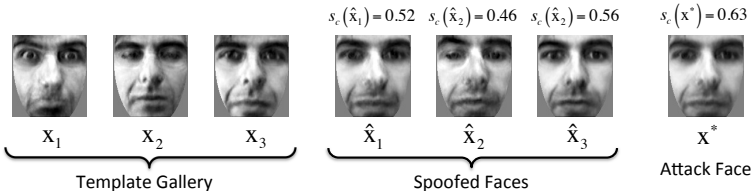


Fig. 3: Face spoofing from multiple images. The client's templates $\{\boldsymbol{x}_i\}_{i=1}^3$, the spoofed faces $\{\hat{\boldsymbol{x}}_j\}_{j=1}^3$, and the final attack face $\boldsymbol{x}^*$ (obtained solving Prob. 2) are shown, along with the corresponding s_c values.

vector space $\mathcal{X}$ using principal component analysis (PCA), and computes the matching score for client c as $s_c(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} s(\boldsymbol{x}, \boldsymbol{x}_i)$, where $s(\boldsymbol{x}, \boldsymbol{x}_i) = \exp\{-\|\boldsymbol{x} - \boldsymbol{x}_i\|\}$, and $\boldsymbol{x}_i$ is the *i-th* template of the claimed identity. We further assume the attack scenario detailed below.

Adversary's Goal. The attacker aims to impersonate a targeted client (*integrity, targeted* attack).

Adversary's Knowledge. She is assumed to know: (i) the feature extraction algorithm, (ii) the matching algorithm s, (iii) the fusion rule s_c, and (iv) a set of n face images $\{\hat{\boldsymbol{x}}_j\}_{j=1}^n$ of the targeted client, different from those in the client's template gallery (*e.g.*, potentially collected from a social network).

Adversary's Capability. She can only submit printed photos of faces to the sensor, during verification.

Adversary's Strategy. Under these assumptions, the attacker can approximate the score $s_c(\boldsymbol{x})$ computed by the targeted system for the claimed identity c using the collected face images of the victim, *i.e.*, she can compute an estimate $\hat{s}_c(\boldsymbol{x}) = \frac{1}{n}\sum_{j=1}^{n} s(\boldsymbol{x}, \hat{\boldsymbol{x}}_j)$. Accordingly, the optimal attack strategy is given by:

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} \hat{s}_c(\boldsymbol{x}) = \frac{1}{n}\sum_{j=1}^{n} s(\boldsymbol{x}, \hat{\boldsymbol{x}}_j)\,, \tag{2}$$

where $\hat{s}_c$ is the attacker's goal function $g(a, \boldsymbol{\theta})$ (Eq. 1), and $\boldsymbol{x}^*$ is the attack sample that maximizes the objective in the PCA-induced feature space. The above problem can be solved by a simple gradient-ascent algorithm, and the resulting attack sample $\boldsymbol{x}^*$ can then be projected back onto the space of face images $\mathcal{Z}$ (where each feature corresponds to the gray-level value of a pixel) by inverting the PCA-induced mapping. This is also possible if more sophisticated matching algorithms and feature representations are used, using well-crafted heuristics. We refer the reader to [21] for further details.

An example of this improved spoofing technique on a simple case involving $n = 3$ templates is shown in Fig. 3, where we also report the values of s_c for each of the client's face images $\{\hat{\boldsymbol{x}}_j\}_{j=1}^n$. It can be seen that the final attack face $\boldsymbol{x}^*$ yields a higher probability (*i.e.*, s_c value) of successfully impersonating the victim than any of the available images $\{\hat{\boldsymbol{x}}_j\}_{j=1}^3$.

B. Poisoning Biometric Systems that Learn from Examples

We report now a different application example focusing on a *poisoning* attack against an *adaptive* face recognition system, and on the development of a corresponding defense. In recent work [20], [21], we

have shown that an attacker may exploit the system's adaptation mechanism to compromise the templates of a given client by presenting a well-crafted sequence of fake faces to the camera, with the goal of denying him access to the system. At the same time, if the attacker replaces the targeted client's templates with her templates, she may also impersonate the client without presenting any fake trait to the sensor.

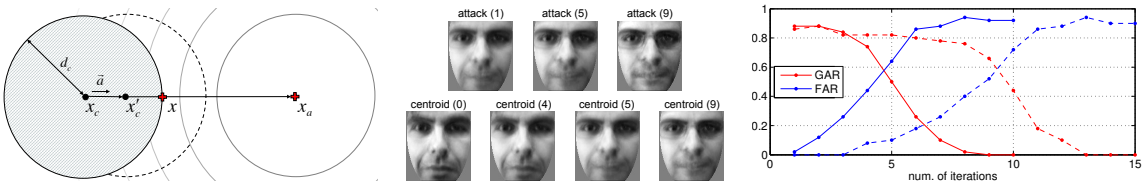
The face verification system considered in this example, as in [20], [21], authenticates clients based on matching the acquired face image with a stored *average* template, referred to as *centroid*. For each client, the centroid is updated using the self-update algorithm: if the submitted face image is similar enough to the centroid, the latter is updated incorporating the new image into the computation of the average face image. As in the previous case, we consider a PCA-based mapping to map face images from $\mathcal{Z}$ onto a reduced vector space $\mathcal{X}$. The matching score for client c is computed here as $s_c(\boldsymbol{x}) = \exp\{-\|\boldsymbol{x}-\boldsymbol{x}_c\|^2\}$, where $\boldsymbol{x}_c$ is the client's centroid. The centroid $\boldsymbol{x}_c$ is initially computed as the average of n templates and, when $s_c(\boldsymbol{x}) \geq \theta_c$, updated as $\boldsymbol{x}'_c = \boldsymbol{x}_c + \frac{1}{n}(\boldsymbol{x} - \boldsymbol{x}_c)$, *i.e.*, slightly drifted towards $\boldsymbol{x}$. Accordingly, in the PCA-based feature space, $\boldsymbol{x}_c$ is updated if the acquired image $\boldsymbol{x}$ is within an hypersphere centered on $\boldsymbol{x}_c$, with radius d_c dependent on the update threshold θ_c. The complete attack scenario is given below.

Adversary's goal. It is that of replacing the centroid $\boldsymbol{x}_c$ of a given client with an attacker's template $\boldsymbol{x}_a$, both to deny access to client c, and to allow the attacker to impersonate c using her own face (*i.e.*, a *targeted* attack violating both system *availability* and *integrity*).

Adversary's Knowledge. The attacker is assumed to know: (i) the feature extraction algorithm; (ii) the matching algorithm; (iii) the template update algorithm; and (iv) the decision and self-update threshold. In the case of *perfect* knowledge, she also knows the centroid $\boldsymbol{x}_c$ of the targeted client c, while when *limited* knowledge is considered, only a good estimate of $\boldsymbol{x}_c$ is available to the attacker, *e.g.*, a frontal face image of the victim collected from a social network.

Adversary's Capability. The attacker can modify the template database by presenting fake faces at the sensor that enable template self-update. The attack influence is thus over the enrollment / update phase.

Attack Strategy. Under these assumptions, the shortest sequence of fake traits required to replace the victim's centroid $\boldsymbol{x}_c$ with that of the attacker $\boldsymbol{x}_a$ can be found by solving the following optimization problem, for each sample $\boldsymbol{x}$ in the attack sequence: $\min_{\boldsymbol{x}} \|\boldsymbol{x} - \boldsymbol{x}_a\|^2$, subject to the update condition $\|\boldsymbol{x}-\boldsymbol{x}_c\| \leq d_c$ (see Fig. 4a). At each iteration, this amounts to finding the closest attack sample to $\boldsymbol{x}_a$ that enables update of $\boldsymbol{x}_c$. The solution is simply given as $\boldsymbol{x} = \boldsymbol{x}_c + d_c\vec{a}$, where $\vec{a} = \frac{\boldsymbol{x}_a - \boldsymbol{x}_c}{\|\boldsymbol{x}_a - \boldsymbol{x}_c\|}$ is the so-called *attack direction*. In practice, each attack sample $\boldsymbol{x}$ is found at the intersection between the hypersphere corresponding to the update condition, and the line connecting $\mathbf{x}_a$ and $\boldsymbol{x}_c$. As in the previous example, the face images for the attack sequence can be obtained by projecting the attack samples from the PCA-



(a) (b) (c)

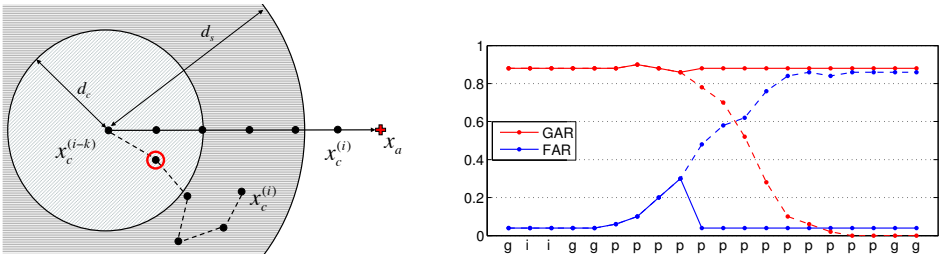
Fig. 4: (a): Poisoning attack with perfect knowledge. The circles centered on $\mathbf{x}_a$ represent the objective function $||\mathbf{x} - \mathbf{x}_a||$, minimized by the attack point $\mathbf{x}$ on the feasible domain $\|\boldsymbol{x} - \boldsymbol{x}_c\| \leq d_c$. The updated centroid $\mathbf{x}'_c$ and the feasible domain for the next attack iteration are also shown. (b) Attack samples and victim's centroids for poisoning with perfect knowledge, at different iterations. (c) GAR and FAR for poisoning with perfect (solid lines) and limited (dashed lines) knowledge, at different iterations.

induced space $\mathcal{X}$ onto the space of face images $\mathcal{Z}$. Then, the attacker can fabricate the corresponding fake faces (*e.g.*, by printing them on paper), and present them in the right order to the sensor.

An example of how the victim's centroid is gradually updated by the corresponding sequence of attack faces, under the assumption of *perfect knowledge*, and for $n = 5$, is given in Fig. 4b. In Fig. 4c, we show how the Genuine and False Acceptance Rate (GAR and FAR) vary as the attack proceeds, under perfect and limited knowledge of the victim's template. Note how the probability of authenticating the attacker (without presenting any fake face) as the victim (*i.e.*, the FAR) increases, while the probability of correctly authenticating the victim as a genuine user (*i.e.*, the GAR) decreases, since the victim's template is gradually *morphed* towards the attacker's face during the attack progress. Results for the perfect and limited knowledge attacks are similar, despite the latter case requires more iterations (*i.e.*, submitting more fake faces) to compensate the lack of knowledge of the victim's template. The exact number of iterations required to complete both attacks can also be analytically computed [21].

Template Sanitization. In the remainder of this section, we present a novel countermeasure based on the idea of *sanitizing* the template gallery (*i.e.*, identifying *anomalous* template updates), inspired by the countermeasures proposed in adversarial machine learning against poisoning attacks [1]–[4], [28].

The underlying idea is to analyze whether the sequence of the most recent k updated centroids $\boldsymbol{x}_c^{(i-k)}, \ldots, \boldsymbol{x}_c^{(i)}$ (where i denotes the current iteration) falls within a given region of the feature space, called *sanitization hypersphere* (see Fig. 5a). If the current centroid $\boldsymbol{x}_c^{(i)}$ falls within the sanitization hypersphere, *i.e.*, if $\|\boldsymbol{x}_c^{(i)} - \boldsymbol{x}_c^{(i-k)}\| \leq d_s$, then the center of the sanitization hypersphere is updated to the next centroid in the sequence, *i.e.*, $\boldsymbol{x}_c^{(i-k+1)}$; otherwise, the current center of the sanitization hypersphere $\boldsymbol{x}_c^{(i-k)}$ is restored as the *current* centroid. In this case, an alert may be also (or alternatively) raised to the system administrator to report the *anomalous* update. The rationale of this approach is to identity sequences of centroid updates that consistently drift the centroid towards a given, biased direction, within a small number of iterations (as it happens in the presence of a poisoning attack), assuming that



(a) (b)

Fig. 5: (a) Template sanitization. If the current centroid $\boldsymbol{x}_c^{(i)}$ falls outside the sanitization hypersphere (dark grey area), as for the poisoning attack sequence (solid line), the centroid $\boldsymbol{x}_c^{(i-k)}$ is restored; otherwise, as for the hypothesized genuine update sequence (dashed line), the center of the sanitization hypersphere is updated to $\boldsymbol{x}_c^{(i-k+1)}$ (red circled point). (b) GAR and FAR values in the presence (solid lines) and in the absence (dashed lines) of template sanitization, after different centroid updates, including genuine ('g') and impostor ('i') attempts, and poisoning attacks ('p') with perfect knowledge.

genuine updates exhibit a different (*i.e.*, less biased and more random) behavior. The parameter k and the hypersphere radius d_s of the proposed approach should thus be chosen such that $d_s \leq k\frac{d_c}{n}$, otherwise poisoning attacks will not be detected, as they drift the centroid of an amount equal to $\frac{d_c}{n}$ at each iteration.

In Fig. 5b, we report an example using the same attacker and victim pair considered in the previous case. Initially, we simulate a number of random accesses to the system, including genuine and impostor attempts, which do not significantly affect GAR and FAR. Then, the attacker launches a poisoning attack, and, as in the previous case, the GAR decreases and the FAR increases. If template sanitization is not implemented, the attack succeeds, and system integrity and availability are compromised. Conversely, in the presence of template sanitization, the attack is detected after four iterations, and a previous centroid is restored. This avoids the attack to succeed, preserving normal system operation and security. Although this simple countermeasure can be misled by a poisoning attack in which the attack samples are closer to the current centroid (instead of lying exactly at the boundary of the feasible domain), this would require the attacker to perform a significantly higher number of iterations to complete the attack. We can thus conclude that the proposed sanitization technique helps improving system security.

Summary and Open Problems

In the last decade, the use of electronic devices in our daily life has become increasingly pervasive, providing several advantages in managing our tasks, and communicating with other people. However, with such a huge number of powerful devices connected to the so-called Internet of Things (*e.g.*, smartphones, smart TVs, *etc.*), the number of potential attack points and vulnerabilities has significantly increased, as well as chances for attackers to compromise the corresponding devices and systems. In addition, the level of sophistication of attacks has become increasingly higher over the years, witnessing the presence of very skilled attackers and strong economic incentives behind. Biometrics can be considered a potential

tool for improving security of such systems in the digital era. However, besides having a strong deterrent effect, they should be really designed to be intrinsically secure, in order to successfully resist to very sophisticated attacks that may be incurred during operation. In this article, we have provided an overview of the current state of the art on biometric system security, under a perspective inspired by the field of adversarial machine learning. We have discussed how this novel perspective may not only inspire the simulation of more sophisticated attack scenarios, but also how, based on such scenarios, more effective countermeasures can be proactively developed. As concrete application examples, we have considered a sophisticated spoofing attack, and a poisoning attack against an adaptive face verification system, in which the attacker gradually compromises the template gallery of a given client by presenting a well-crafted sequence of fake faces. We have also proposed a novel countermeasure based on *template sanitization*. Another example of how the proposed perspective based on adversarial machine learning may depict novel attack scenarios and inspire potential countermeasures is related to recent work in adversarial learning, which has shown that clustering algorithms may be significantly vulnerable to well-crafted attacks [30]. In the biometric setting, similar attacks may target systems that perform template selection and update exploiting clustering algorithms. Although investigating and developing secure clustering algorithms against adaptive and intelligent attackers is still an open issue in the adversarial machine learning field, the corresponding results may also inspire countermeasures that can be adapted to improve the security of template selection and update procedures in biometric systems.

About the Authors

All authors are affiliated with the Dept. of Electrical and Electronic Engineering at the University of Cagliari, Italy. They gave seminal contributions to the field of adversarial machine learning [3], [30] and co-organized the first Dagstuhl Perspectives Workshop on "Machine Learning Methods for Computer Security" [29]. They are part of the AISec program committee, the leading workshop on adversarial machine learning. They published conference and journal papers on biometric system security [21], [22], and created the successful series of the LivDet context for fingerprint liveness detection methods.

Battista Biggio is a Postdoctoral Researcher, working on adversarial learning, multiple classifier systems, kernel methods, biometric authentication, spam and malware detection. He serves as a reviewer for the main international conferences and journals in this field. He is a member of the IEEE.

Luca Didaci is an Assistant Professor of Computer Engineering, working on methodologies and applications of statistical pattern recognition, adversarial machine learning and biometrics. He is a reviewer for top conferences and journals in these areas, and a member of the IEEE.

Paolo Russu is a PhD Student, working on adversarial learning, biometrics, and computer security.

Giorgio Fumera is an Associate Professor of Computer Engineering, working on methodologies and applications of statistical pattern recognition, adversarial classification and document categorization. He has published more than sixty papers in international journals and conferences. He acts as reviewer for the main international conferences and journals in this field, and he is a member of the IEEE.

Fabio Roli is a Full Professor of Computer Engineering. His research over the past twenty years addressed the design of pattern recognition systems in real applications. He played a leading role for the research field of multiple classifier systems. He is Fellow of the IEEE and Fellow of the IAPR.

ACKNOWLEDGEMENTS

This work has been partly supported by the TABULA RASA project, 7th Framework Research Programme of the European Union (EU), grant agreement number: 257289; and by the project CRP-18293 funded by Regione Autonoma della Sardegna, L.R. 7/2007, Bando 2009.

REFERENCES

[1] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proc. ACM Symp. Information, Computer and Comm. Sec.*, ser. ASIACCS '06. New York, NY, USA: ACM, 2006, pp. 16–25.

[2] L. Huang, A. D. Joseph, B. Nelson, B. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *4th ACM Workshop Artificial Intell. and Sec.*, Chicago, IL, USA, 2011, pp. 43–57.

[3] B. Biggio, G. Fumera, and F. Roli, "Security evaluation of pattern classifiers under attack," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 4, pp. 984–996, 2014.

[4] B. Biggio, G. Fumera, and F. Roli, "Pattern recognition systems under attack: Design issues and research challenges," *Int'l J. Patt. Recogn. Artif. Intell.*, vol. 28, no. 7, p. 1460002, 2014.

[5] P. Laskov and R. Lippmann, Eds., *NIPS Workshop on Mach. Learn. in Adv. Environments for Computer Security*, 2007.

[6] A. D. Joseph, P. Laskov, F. Roli, and D. Tygar, Eds., *Dagstuhl Persp. W. Mach. Learn. Methods for Computer Sec.*, 2012.

[7] N. K. Ratha, J. H. Connell, and R. M. Bolle, "An analysis of minutiae matching strength," in *AVBPA*, ser. LNCS, J. Bigün and F. Smeraldi, Eds., vol. 2091. Springer, 2001, pp. 223–228.

[8] A. K. Jain, K. Nandakumar, and A. Nagar, "Biometric template security," *J. Adv. Sign. Proc.*, vol. 2008, pp. 1–17, 2008.

[9] R. N. Rodrigues, L. L. Ling, and V. Govindaraju, "Robustness of multimodal biometric fusion methods against spoof attacks," *J. Vis. Lang. Comput.*, vol. 20, no. 3, pp. 169–179, 2009.

[10] A. Adler, "Vulnerabilities in biometric encryption systems," in *5th Int'l Conf. Audio- and Video-Based Biometric Person Auth.*, ser. LNCS, T. Kanade et al. Eds., vol. 3546. Hilton Rye Town, NY, USA: Springer, July 20-22 2005, pp. 1100–1109.

[11] J. Galbally, C. McCool, J. Fierrez, S. Marcel, and J. Ortega-Garcia, "On the vulnerability of face verification systems to hill-climbing attacks," *Pattern Recogn.*, vol. 43, no. 3, pp. 1027–1038, 2010.

[12] M. Martinez-Diaz, J. Fierrez, J. Galbally, and J. Ortega-Garcia, "An evaluation of indirect attacks and countermeasures in fingerprint verification systems," *Patt. Rec. Letters*, vol. 32, no. 12, pp. 1643 – 1651, 2011.

[13] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *European Conf. Mach. Learn. and Principles and Practice of Knowledge Disc. in Databases, Part III*, ser. LNCS, H. Blockeel et al., Eds., vol. 8190. Springer Berlin, 2013, pp. 387–402.

[14] A. Globerson and S. T. Roweis, "Nightmare at test time: robust learning by feature deletion," in *Proceedings of the 23rd Int'l Conf. on Mach. Learn.*, W. W. Cohen and A. Moore, Eds., vol. 148. ACM, 2006, pp. 353–360.

[15] B. I. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S.-h. Lau, S. Rao, N. Taft, and J. D. Tygar, "Antidote: understanding and defending against poisoning of anomaly detectors," in *Proc. 9th ACM SIGCOMM Internet Meas. Conf.*, ser. IMC '09. New York, NY, USA: ACM, 2009, pp. 1–14.

[16] D. Lowd and C. Meek, "Adversarial learning," in *Proc. 11th ACM SIGKDD Int'l Conf. Knowl. Disc. Data Mining*. Chicago, IL, USA: ACM Press, 2005, pp. 641–647.

[17] B. Nelson, B. I. Rubinstein, L. Huang, A. D. Joseph, S. J. Lee, S. Rao, and J. D. Tygar, "Query strategies for evading convex-inducing classifiers," *J. Mach. Learn. Res.*, vol. 13, pp. 1293–1332, 2012.

[18] U. Uludag, A. Ross, and A. K. Jain, "Biometric template selection and update: a case study in fingerprints," *Patt. Rec.*, vol. 37, no. 7, pp. 1533–1542, 2004.

[19] C. Ryu, H. Kim, and A. K. Jain, "Template adaptation based fingerprint verification," in *Proceedings of the 18th Int'l Conf. on Pattern Recognition - Volume 04*, ser. ICPR '06. Washington, DC, USA: IEEE CS, 2006, pp. 582–585.

[20] B. Biggio, G. Fumera, F. Roli, and L. Didaci, "Poisoning adaptive biometric systems," in *Structural, Syntactic, and Statistical Pattern Recognition*, ser. LNCS, G. Gimel'farb et al., Eds. Springer, 2012, vol. 7626, pp. 417–425.

[21] B. Biggio, L. Didaci, G. Fumera, and F. Roli, "Poisoning attacks to compromise face templates," in *6th IAPR Int'l Conf. Biometrics*, Madrid, Spain, 2013, pp. 1–7.

[22] B. Biggio, Z. Akhtar, G. Fumera, G. L. Marcialis, and F. Roli, "Security evaluation of biometric authentication systems under real spoofing attacks," *IET Biometrics*, vol. 1, no. 1, pp. 11–24, 2012.

[23] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino, "Impact of artificial "gummy" fingers on fingerprint systems," *Datenschutz und Datensicherheit*, vol. 26, no. 8, 2002.

[24] P. Johnson, B. Tan, and S. Schuckers, "Multimodal fusion vulnerability to non-zero effort (spoof) imposters," in *IEEE Int'l Workshop on Information Forensics and Security*, 2010, pp. 1–5.

[25] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma, "Adversarial classification," in *10th ACM SIGKDD Int'l Conf. on Knowl. Disc. Data Mining*, Seattle, 2004, pp. 99–108.

[26] M. Brückner, C. Kanzow, and T. Scheffer, "Static prediction games for adversarial learning problems," *J. Mach. Learn. Res.*, vol. 13, pp. 2617–2654, 2012.

[27] C. H. Teo, A. Globerson, S. Roweis, and A. Smola, "Convex learning with invariances," in *NIPS 20*, J. Platt et al., Eds. Cambridge, MA: MIT Press, 2008, pp. 1489–1496.

[28] G. F. Cretu, A. Stavrou, M. E. Locasto, S. J. Stolfo, and A. D. Keromytis, "Casting out demons: Sanitizing training data for anomaly sensors," in *IEEE Symp. Security and Privacy*. Los Alamitos, CA, USA: IEEE CS, 2008, pp. 81–95.

[29] A. D. Joseph, P. Laskov, F. Roli, J. D. Tygar, and B. Nelson, "Machine Learning Methods for Computer Security (Dagstuhl Perspectives Workshop 12371)," *Dagstuhl Manifestos*, vol. 3, no. 1, pp. 1–30, 2013.

[30] B. Biggio, I. Pillai, S. R. Bulò, D. Ariu, M. Pelillo, and F. Roli, "Is data clustering in adversarial settings secure?" in *Proc. ACM Workshop on Artificial Intell. and Sec.*, ser. AISec '13. New York, NY, USA: ACM, 2013, pp. 87–98.