Characterizing and evaluating adversarial examples for Offline Handwritten Signature Verification

Luiz G. Hafemann, Robert Sabourin, *Member, IEEE,* and Luiz S. Oliveira.

***Abstract*—The phenomenon of Adversarial Examples is attracting increasing interest from the Machine Learning community, due to its significant impact to the security of Machine Learning systems. Adversarial examples are similar (from a perceptual notion of similarity) to samples from the data distribution, that "fool" a machine learning classifier. For computer vision applications, these are images with carefully crafted but almost imperceptible changes, that are misclassified. In this work, we characterize this phenomenon under an existing taxonomy of threats to biometric systems, in particular identifying new attacks for Offline Handwritten Signature Verification systems. We conducted an extensive set of experiments on four widely used datasets: MCYT-75, CEDAR, GPDS-160 and the Brazilian PUC-PR, considering both a CNN-based system and a system using a handcrafted feature extractor (CLBP). We found that attacks that aim to get a genuine signature rejected are easy to generate, even in a limited knowledge scenario, where the attacker does not have access to the trained classifier nor the signatures used for training. Attacks that get a forgery to be accepted are harder to produce, and often require a higher level of noise - in most cases, no longer "imperceptible" as previous findings in object recognition. We also evaluated the impact of two countermeasures on the success rate of the attacks and the amount of noise required for generating successful attacks.**

***Index Terms*—Adversarial Machine Learning, Signature Verification, Biometrics**

I. Introduction

Biometric systems are extensively used to establish a person's identity in legal and administrative tasks [1]. They are commonly modeled as Pattern Recognition systems, in which biometric data from an individual is acquired (e.g. during an enrollment process), and stored as a "template" for future comparisons, or used to train a classifier that can discriminate if new samples belong to this user.

The reliability of these systems have security implications, and in the last decade these systems have been analyzed from an Adversarial Machine Learning perspective. From this viewpoint, we consider an active adversary, with its own goals (e.g. getting access to a system), knowledge (e.g. knowing the classifier parameters, or the learning algorithm) and capabilities (e.g. ability to manipulate the training data, or the inputs during test time). In particular, Ratha et al. [2] and later Biggio et al. [3] characterize the different components of a biometric system that can be attacked.

L. G. Hafemann and R. Sabourin are with the Laboratoire d'imagerie, de vision et d'intelligence artificielle, École de technologie supérieure, Université du Québec, Montreal, Canada. (e-mail: lghafemann@livia.etsmtl.ca, robert.sabourin@etsmtl.ca)

L. S. Oliveira is with the Department of Informatics, Federal University of Parana, Curitiba, Brazil (e-mail:lesoliveira@inf.ufpr.br)

This work was supported by the Fonds de recherche du Québec - Nature et technologies (FRQNT), the CNPq grant #206318/2014-6 and by the grant RGPIN-2015-04490 to Robert Sabourin from the NSERC of Canada.

However, an emerging issue of "Adversarial Examples" pose new security concerns for such systems. This issue refers to adversarial input perturbations specially crafted to induce misclassifications. Szegedy et al. [4] showed that very small perturbations on images (almost imperceptible) could be crafted to mislead a state-of-the-art CNN-based classifier. Moreover, attacks crafted for one model often transfer to other models, meaning that an attacker could train its own surrogate classifier to generate attacks, as long as it has access to data from the same data distribution. This issue has been analyzed in many recent papers [5], [6], [7], [8], [9], but the theoretical reasons are not fully understood, and most defenses are weak (i.e. they fail if the attacker knows about the defense).

We evaluate this new threat for biometric systems, by characterizing the potential new attacks under a taxonomy of threats to such systems [2], [3]. We consider particular attack scenarios to Offline Handwritten Signature Verification, identifying the attacker's goals, required knowledge and capabilities.

It is worth noting that attacking verification systems can present difficulties not present in classification problems. In particular, as new users join the system, they introduce a new *class*, not only unseen examples of existing classes. We present a refined version of the adversary's knowledge model that explicitly makes the distinction of whether access to data from a particular individual of interest is available to the attacker.

We conducted experiments on Writer-Dependent classifiers trained with a CNN-based representation (SigNet) and a handcrafted feature extractor (CLBP), considering four widely used Datasets: MCYT, CEDAR, GPDS-160 and the Brazilian PUC-PR. We defined a comprehensive set of experiments to evaluate such systems under different scenarios of the adversary's knowledge level and objectives, using four attack methods (gradient-based and gradient-free). Our main contributions are as follows:

- We characterize different attack scenarios for Offline Handwritten Signature Verification systems, focused on new threats introduced by Adversarial Examples.
- We identify that there is an asymmetry in the attacks, empirically showing that attacking genuine signatures (so that they are rejected) can be done with high success rate and a relatively low amount of noise, while attacking forgeries (so that they are accepted) is a much harder.
- Our experiments with different scenarios of attacker knowledge show that attacks can be done even with Limited Knowledge, where the attacker has no access to the signatures used to train the classifiers, showing that this transferability affects both CNN-based systems and systems based on handcrafted features. We also identify that attack transferability is greatly reduced if the CNN

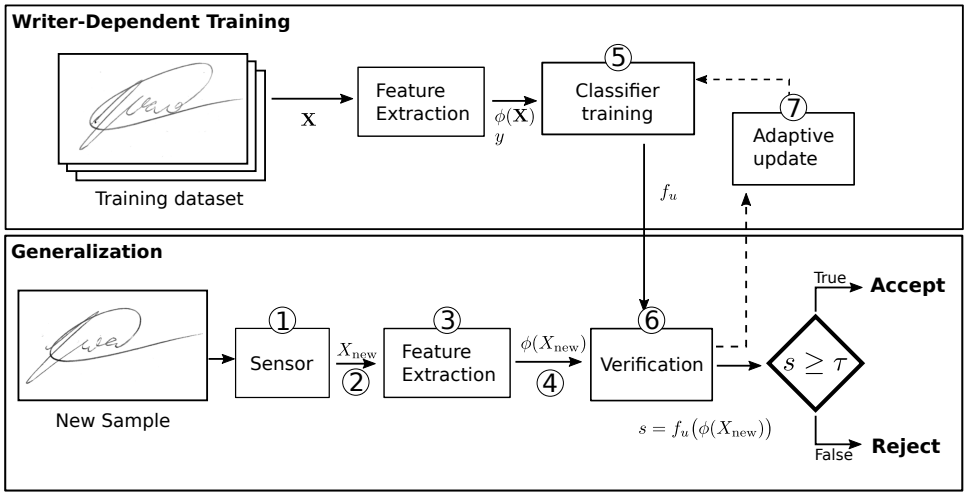


Fig. 1: A typical writer-dependent signature verification system, with annotated points of attack. On the training phase, a classifier f_u is trained for each user. During operations, for a new sample X_{new} we obtain a feature vector $\phi(X_{\text{new}})$, and use the classifier f_u to accept or reject the signature. For adaptive systems, an update rule select signatures for classifier adaptation.

is trained on a different subset of *users*, contrasting with previous findings that attacks transfer well if the CNN is trained on a different subset of samples from the same classes [4].

- Lastly, we evaluate the impact of countermeasures and find that the Madry defense [10] is effective in increasing the amount of noise necessary to make a sample adversarial, even when it is applied only to the feature learning phase, and not on training the WD classifiers. Code for reproducing the experiments will be made publicly available at https://github.com/luizgh/adversarial_signatures.

The paper is organized as follows: in section II we introduce the main concepts of security in biometric systems; in section III we present the issue of adversarial examples and in section IV we present particular attack scenarios for offline signature verification, and a refinement of the adversary's knowledge model. Section V describes the experimental protocol, and the results are discussed in section VI. Finally, our conclusions are listed in section VII.

II. Security in biometric systems

The security of machine learning systems have been widely studied in the past decade. Barreno et al. [11], [12] categorize attacks to such systems along three axes: (i) the influence of the attack, that can be causative (when training data is compromised) or exploratory (probing the learner to acquire information); (ii) the specificity of the attack: *targeted*, in which a particular point or a set of points is targeted or *indiscriminate*; and (iii) the security violation of the attack, that can seek an integrity violation (e.g. intrusion) or availability disruption (e.g. make the system unusable for legitimate users).

Biggio et al. [3], [13] further expands this analysis for biometric systems, incorporating a model of the adversary that includes its *goals*, *knowledge* of the target system, and *capabilities* of manipulating the input data or system components. The goals of an attacker are mainly divided in: 1) *Denial of service*: preventing real users from using the system; 2) *Intrusion*: impersonating another user; 3) *Privacy violation*: stealing private information from an user (such as the biometric templates). The *knowledge* of the adversary refers to the information of the target system that is available to the adversary, such as perfect knowledge (e.g. knowledge of the feature extractor, type of classifier and model parameters) or limited (partial) knowledge of the system. The *capabilities* of the adversary refer to what it can *change* in the target system, such as changing the training set (poisoning attack), or the inputs to the system at test time (evasion attack).

Modeling the knowledge of the adversary was formalized by Biggio and Roli [14]. Let $\mathcal{X}$ and $\mathcal{Y}$ be the feature and label spaces, respectively, and $\mathcal{D}$ be a dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^{n}$ of n training samples. Let f be a training algorithm (classifier), and w be a collection of its parameters and hyper-parameters. The knowledge of the attacker can be formalized as a set θ, containing the components of the system that are known to the attacker. Perfect-Knowledge (PK) attacks consider full knowledge of the system, that is, $\theta_{PK} = (\mathcal{D}, \mathcal{X}, f, w)$. We can also consider Limited Knowledge (LK) attacks, in which some of the information is not available to the adversary. As an example, if the adversary does not have access to the learned weights of the model, but has access to the training data, a surrogate classifier f can be trained (learning parameters $\hat{w}$) and used to generate the attack. Similarly, if the training data is not available, the adversary may be able to collect another training set from the same data distribution and use it to train the surrogate classifier. In this last scenario, the knowledge of the attacker would be represented as $\theta_{LK} = (\hat{\mathcal{D}}, \mathcal{X}, f, \hat{w})$. The hat symbol ($\hat{}$) indicate limited knowledge of a component (such as getting a surrogate dataset from the same data distribution).

Biometric systems are composed of several components, such as the sensors capturing the biometric, and software to extract features, store templates and perform classification. Ratha et al [2] identified eight points of attack on biometric security systems, that were later grouped by Jain et al [15]

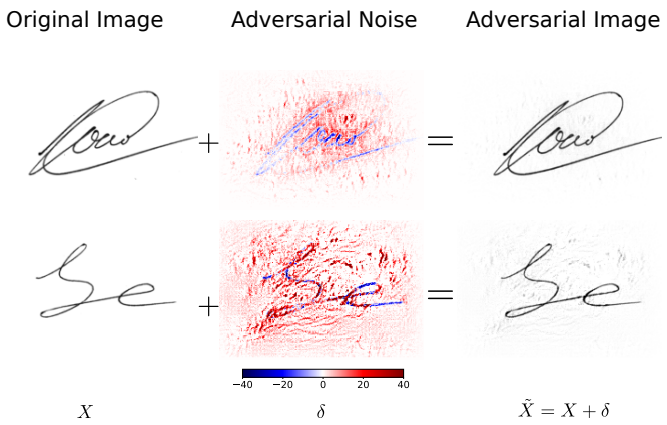


Fig. 2: Illustration of adversarial examples. An adversarial noise δ is added to original images X, such that the resulting image $\tilde{X}$ is misclassified. **Top**: Type-I attack: a genuine signature from user u_1 (left) is attacked to be classified as a forgery (right). **Bottom**: Type-II attack. The original image (left) is from user u_2 (i.e. a random forgery for u_1), and is attacked to be classified as a genuine (right).

and extended by Biggio et al [3] to include multi-modal systems and adaptive systems. The set of this attack points is considered the *attack surface* of the system. Figure 1 shows a typical User-Dependent classification system, with the main attack points. Below we discuss the main threats to the different points of attack.

The first point of attack (#1) in a biometric system is the user interface that collects the sample (e.g. a scanner capturing a document with a signature, or a mobile application taking a picture of a bank cheque). For many biometrics, attacks on this first point mainly consist of spoofing attacks, that normally use a fabricated fake biometric trait. Possible defenses for such attacks rely on liveness detection. On the signature verification task, simulated and traced forgeries can be considered attacks targeting this stage. A second set of attack points refer to attacks in the communication between different components of the system (#2, #4) (for example, intercepting and replacing the sensor input or the extracted features, that is input to the subsequent module). Defenses for such attacks involve encrypting the communication between the different modules. The software modules (#1, #3, #5, #6, #7) may present vulnerabilities in the code (such as buffer overflow) that can be exploited by a malicious user. The classifier training (#5) can be targeted for poisoning attacks (e.g. adding samples from another user in the training data for subsequent intrusion). For adaptive systems, the template update rule (#7) can be targeted to update the template database (e.g. for intrusion).

III. Adversarial Examples

Adversarial examples are samples similar to the true data distribution, but that fool a classifier. In computer vision, these are images $\tilde{X}$ that are visually similar to a "real" image X, but that fool a classifier (i.e. the classifier predicts an incorrect class for $\tilde{X}$: $\text{argmax}_y P(y|\tilde{X}) \neq \text{argmax}_y P(y|X)$).

Szegedy et al. [4] showed that for deep neural networks, we can run an optimization procedure to produce a small change δ to an image, such that $\tilde{X} = X + \delta$ is an adversarial example, as illustrated in Figure 2. Perhaps more surprisingly, they also discovered that an attack that is created to fool one network also fools other networks (trained on different subsets of data), meaning that attacks can be created even without full knowledge of the classifier under attack. It was later shown that such attacks can be done in the physical world [16], where adversarial images printed on paper and later captured with a camera also fooled a classifier. Lastly, although some defense strategies have been proposed [5], [6], [7], [10], most solutions are not robust to strong iterative attacks. Even *detecting* that an input is adversarial is a hard task [9].

Most of the recent research on this area concentrates on differentiable classifiers (usually Deep Learning models), creating attacks that use gradient information of the loss function with respect to the inputs. However, most feature extractors used in signature verification (such as LBP, HOG) are non-differentiable, and therefore attacks to systems using these features could not rely on gradient-based methods. Some methods proposed in the literature do not rely on gradient information, and could potentially be used for this task. Papernot et al. proposed *Substitute model training* [17], in which the attacker train a substitute (differentiable) model, and use it to generate the attack. Brendel et al. proposed a *Decision-based attack* [18], that relies only on the decision (prediction) of the model under attack. Its strategy is the opposite of most attacks: given an image X and an image $\tilde{X}^0$ that is from another class, the algorithm iteratively refines $\tilde{X}^k$ to be closer to $\tilde{X}$ (e.g. in L_2 norm). The image $\tilde{X}^0$ can be a random image (e.g. sampled at random until it is from the desired class), or an actual image from a target class. Chen et al. proposed a *Zeroth order optimization* method [19], where the gradient is estimated numerically. Doing so naively is impractical (due to the dimensionality of the input), so the authors employ techniques to reduce the computational complexity of this estimation (block coordinate descent, attack-space dimension reduction, hierarchical attacks and importance sampling). With all these techniques, the attack has shown to scale to imagenet ($299 \times 299 \times 3$ pixels), producing an attack in 20 minutes. This method requires the function to be smooth and Lipshitz continuous. Ilyas et al. proposed using *Natural Evolution Strategy (NES) gradient estimate* [20] - instead of using numerical methods to estimate the gradient (as above), the authors propose using Natural Evolution Strategies for the gradient estimate. These estimates are given by computing the loss function along random directions. The authors claim that this method require 1-2 orders of magnitude less computations of the loss function. Lastly, Ramanathan et al. [21] explored using *Simulated annealing* for creating adversarial examples for a system based on HOG features with a linear SVM classifier. In each iteration, a small perturbation is applied to the image, and the distance of the new image to the SVM hyperplane is used as a condition to accept the new point. With this approach the authors were able to craft adversarial images with imperceptible noise that fooled the HOG+SVM classifier.

A. Attacks considered in this paper

In this paper, we consider two gradient-based attacks (that can be used when the classifiers are differentiable with respect to the input), and two gradient-free attacks, that can be used even if the features and/or classifiers are non-differentiable. In this paper we are mostly interested in feature extractors widely used for signature verification, and chose the LBP descriptor, which is used in several studies [22], [23], [24]. Since LBP is highly discontinuous (due to the thresholding using the center pixel's value), methods that estimate the gradient are less interesting: the gradient should be very discontinuous (0 almost everywhere), since for each pixel, the transition from one pattern to the other is a step function. For this reason we selected two methods that do not rely on estimating the gradients: the decision-based attack [18] and the optimization using Simulated annealing. For the gradient-based attacks, we considered the Fast Gradient Method (FGM) [5] and the Carlini & Wagner attack [8].

The decision-based attack [18] is an iterative method: given an image X from class y_i, the objective is to find an image $\tilde{X}^k$ that is classified as a different class, and minimizes the distance $D(X, \tilde{X}^k)$ for some distance measure. It starts with a sample $\tilde{X}^0$ from a class $y \neq y_i$. In each step, first the sample is projected in a random direction that is orthogonal to $(\tilde{X}^{k-1} - X)$ (i.e. orthogonal to a straight line to the sample X), and then takes a step in the direction of X. If the point is still from a class different than y_i, it is accepted as the next point $\tilde{X}^k$, otherwise a new point is searched in another random direction. This method therefore only requires the decision of the model (which class a sample $\tilde{X}^k$ belongs to).

The annealing method uses the well known simulated annealing method as a gradient-free optimization method. Starting from the image X, we add a small perturbation obtaining $\tilde{X}^k$. If the resulting image is closer to the decision boundary of the SVM (i.e the score decreases/increases depending on the type of attack), it is accepted as the next point. Otherwise, with a probability inversely proportional to the current step, it is still accepted as the next point. In the work from Ramanathan et al. [21], the authors consider as the objective function simply to reduce the distance to the SVM hyperplane, and stop optimization when the boundary is crossed. In our experiments, we found it necessary to include a penalty on the L_2 norm of the noise added to the image. This is further detailed in section V.

The FGM attack is a one-step gradient-based attack. In this paper we consider the version of this attack focused on the L_2 norm:

$$\tilde{X} = X + \epsilon \frac{\nabla J(x, y)}{\|\nabla J(x, y)\|_2} \tag{1}$$

Where X is the original image, $\nabla J(x, y)$ is the gradient of the loss function with respect to the input, and ϵ is a hyperparameter that controls the size of the update. The adversarial image is then clipped to the allowed range of the input (e.g. pixels between 0 and 255).

The Carlini & Wagner L_2 attack uses an iterative gradient attack, using a gradient descent method (the Adam optimizer). The objective to be minimized contains two terms: a term minimizing the noise δ and a term encouraging the model to misclassify the image:

$$\min_w \|\delta\|_2 + cf(X + \delta) \tag{2}$$

Where c trades-off between the two objectives, and is chosen with a binary search (the smallest c that still obtains a misclassified image). Instead of enforcing hard constraints on the adversarial image (to keep pixel values between 0 and 255), the authors propose a change of variable. First, they consider images normalized between 0 and 1. Then, to enforce that $X + \delta \in [0, 1]$ they consider the following change of variable:

$$\delta_i = \frac{1}{2}(\tanh(w_i) + 1) - x_i \tag{3}$$

Since $-1 \leq \tanh(w_i) \leq 1$, it follows that $0 \leq X_i + \delta_1 \leq 1$, satisfying the box constraints on the resulting image, but putting no constraints on the variable under optimization (w). As for the term that encourages the model to misclassify the image, they choose a term that seeks to increase the distance between the logits (pre-softmax activation) of the target class t and the class with maximum prediction (other than the target class):

$$f(X) = \max(\max_{i \neq t}(Z(X)_i) - Z(X)_t, -\kappa) \tag{4}$$

Where $Z(X)$ is the logit (pre-softmax activation) and κ is a constant that can be used to select how confident the model must be in the wrong class prediction. This loss function has no constraints, and can be solved by any gradient-based method.

B. Countermeasures

Under a paradigm of *Security by design*, systems should be designed to be secure from the ground up. In the case of Machine Learning, systems should be designed explicitly considering an adversary [14]. Dalvi et al [25] presented one of the first formulations of this problem, by considering a *game* between the classifier and the adversary. They propose a solution of this game for naive bayes classifiers, considering a classifier that performs as well as possible against an optimal adversary. This has some resemblance to recent approaches proposed for adversarial examples called *Adversarial Training* [5], [7], in which the training procedure is augmented with adversarial samples, with the objective of increasing robustness of the systems.

In this work we are concerned with the new vulnerabilities introduced by adversarial changes in the input images that induce misclassifications in Signature Verification systems. In this setting, some defenses become harder to implement - for instance, Biggio et al [26] propose learning the support ($P(X)$) and incorporating this knowledge on the classifier training. Learning this support when X is high dimensional (which is the case in signature images, eg. 150×200 pixels in this work) is a hard task, specially when just a few samples

per user are available. The problem of working with large models and input dimensions is explored in recent work in adversarial examples for deep neural networks. For instance by *Adversarial training* [5], [7]; *defensive distilation* (retraining a network with knowledge extract from a previous training) [6]; and techniques to add non-differentiable steps in the inference process (e.g. transforming the input with non-differentiable operations [27]). Most defenses, however, have been shown to fail when the attacker has knowledge of them. Tramer et al. [7] showed that Adversarial training is not robust to iterative attacks on a white-box (PK) scenario; Carlini and Wagner showed that distillation is also not effective in this scenario [8]. More recently, Athalye et al. showed that almost all defenses presented in recent ICLR and CVPR conferences can be bypassed [28], [29]. The only exception was the work of Madry et al. [10], that propose a framework that provides guarantees against attacks with a maximum L_∞ norm. However, as noted in [29], this defense is hard to scale (the authors only reported results on the CIFAR-10 dataset, which consists of small images of 32×32 pixels), and that resistance to L_∞ attacks does not guarantee resistance to other scenarios (e.g. when the attacker is limited by a maximum L_2 norm of the noise). This problem therefore remains as an open research question.

In this paper we focus our attention in defenses for the CNN-based models, in particular by evaluating two defenses: Ensemble Adversarial Training [7] and the Madry defense [10]. The first has demonstrated some robustness in Limited Knowledge scenarios, while the second is a proposed defense against perfect-knowledge attacks.

For the ensemble adversarial training, we first train M models on the task at hand. Then we train another model with the following loss function:

$$\tilde{J}(X, y, \theta) = \alpha J(X, y, \theta) + (1 - \alpha) J(\tilde{X}, y, \theta) \tag{5}$$

Where $J(X, y, \theta)$ is the cross-entropy loss function of a sample X with true label y, and $\tilde{X}$ is an adversarial sample generated using FGM (equation 1) either using the model being trained, or one of the M previously trained models.

The Madry defense involves a saddle point optimization problem, in which we optimize for the worst case:

$$\begin{aligned} &\min_{\theta} p(\theta) \\ \text{where} \quad &p(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[\max_{\delta\in\mathcal{S}} J(X + \delta, y, \theta)\big] \end{aligned} \tag{6}$$

Where $\mathcal{S}$ defines a feasible region of the attack (i.e. the attacker capability). For instance, to add robustness against attacks that minimize the L_2 norm of the attacks, we train the classifier with an adversary constrained to $\mathcal{S} = \{\delta : \|\delta\|_2 < \epsilon\}$, for a given maximum perturbation ϵ.

Lastly, we also consider a countermeasure using background removal. Handwritten data has an important difference compared to other vision tasks, such as object recognition, where we have a clear and simple separation of background and foreground. This is an important distinction because adversarial samples usually involve adding a crafted "noise" all around the image. To this end, we investigate the impact (on the attack success rate) of removing the background after the adversarial samples are generated.

IV. Attack scenarios for Offline Handwritten Signature Verification

We now consider the possible attacks to biometric systems based on adversarial examples $\tilde{X}$. In particular, we identify possible attack points, and provide specific scenarios for Offline Handwritten Signature Verification.

The attacks using adversarial examples involve changing the inputs to the classifier, and therefore we identify two potential areas of vulnerability: at the sensor level, or the template storage/update level. The most prominent aspect of adversarial examples is that they fool a machine learning system without fooling humans (i.e. $\tilde{X}$ being visually similar to X). This is an important difference to spoofing attacks (that also target the sensor level), since these fake biometric traits, such as a "gummy finger", are clearly identified as different from a real finger by a human. We identify the following new attacks on signature verification systems, along with possible goals of an attacker:

1) **Attacks on the data capture** (targets point #1). In this case the adversarial image is crafted before the image is collected for the system. That is, an adversary can craft adversarial images $\tilde{X}$, and present them to the system, for instance using a banking application that allows a customer to use a picture of a cheque to cash it; or by printing adversarial noise on a physical signature. We identify two types of attack:
   - **Type-I attack** (false rejection): Present a genuine signature that fools the system as being a forgery. This can be used for denial of service (preventing genuine users to accessing a system). We can also make a parallel to disguised signatures, where the user signs a document with the intent of later denying it (for example, the receiver of a check accepts it, but fails to cash it as the system classifies it is a forgery).
   - **Type-II attack** (false acceptance): Present a random forgery (i.e. a genuine signature from user y_i) that fools the system as being genuine for user y_j ($j \neq i$). At the same time, to a person, this sample can show no signs of being forgery (if it is not compared to a reference), since it is a genuine signature. The attacker can also use a skilled forgery as "starting point", creating noise to increase the likelihood of the forgery being accepted.
2) **Attacks on the templates** (targets point #5): If original images are stored as part of the system (e.g. for classifier re-training, or manual verification in case of system failure/rejection of a sample), the templates can be changed to still look like genuine signatures for human operators, but in a way that accept signatures from a different person as genuine.
3) **Attack on template update** (targets point #7): For adaptive systems, the attacker can craft changes on the user's signature, so that adversarial templates are

added to the gallery, to enable an intrusion later using a signature from another person. Similarly to the point above, the templates would appear as genuine to a person.

The attacks above require different capabilities from the part of the attacker. The first attack only affects the system at test time (evasion attack), and in many practical scenarios would require the creation of a *physical* attack, that is, the creation of an adversary signature in a piece of paper, for instance by printing adversarial noise on top of a handwritten signature. The second attack is a poisoning attack, that does not require a physical sample, as it impacts the stored templates of an user. However, it requires the capability of the attacker to update the template database, and can be categorized as an *insider attack* as per the terminology used by Biggio et al [3]. Note that this attack differ from simply adding another user's biometric to the templates, since a manual inspection of the templates would not reveal that the templates have been tampered with. The third attack can also be seen as a poisoning attack, affecting adaptive systems, that automatically add new samples to the set of user templates.

As for the knowledge required from the adversary, we can consider different scenarios, ranging from full knowledge of the system, to scenarios where only limited information is available to the attacker.

A. Refining the adversary's knowledge model

For biometric *verification* tasks, we identify an important refinement of the adversary's knowledge model. We argued in section II that an adversary that does not have access to the training set can collect its own data $\hat{\mathcal{D}}$ from the same data distribution, and train a surrogate classifier. For verification systems, each new user to the system effectively introduces a new class, and therefore it is important to make a distinction of accessing data for a particular individual of interest, and a "background class", that are negative examples for a given user (e.g. signatures from other users). We refer therefore to two data components: $\mathcal{D}_b$ - biometric data from the background class (i.e. not for the individual under attack), and $\mathcal{D}_u$ - biometric data from the targeted individual. This allows the definition of limited knowledge scenarios where the biometric sample of the user can be collected, or for scenarios where the adversary can only collect samples from a other users.

In our experiments, we consider three attack scenarios:

- **Perfect Knowledge** scenario: the attacker has knowledge of all components: $\theta_{PK} = (\mathcal{D}_b, \mathcal{D}_u, \mathcal{X}, f, w)$. This scenario serves as a tool to analyze the worst-case scenario (from the system's defense perspective).
- **Limited Knowledge #1**: we consider a scenario where the attacker does not have access to the dataset used for training the classifiers, but has access to all other components. We consider that the attacker is able to collect signatures from some users ($\hat{\mathcal{D}}_b$, that are from different users from those used to train the system), and some signatures from the user of interest, that were not used for training the system: $\hat{\mathcal{D}}_u$. In this case, $\theta_{LK1} = (\hat{\mathcal{D}}_b, \hat{\mathcal{D}}_u, \mathcal{X}, f, \hat{w})$.

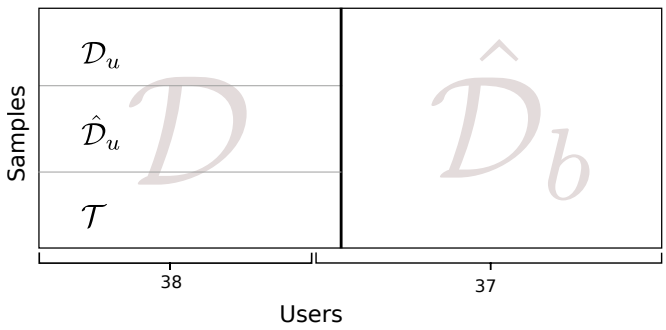


Fig. 3: Dataset separation for the MCYT dataset. The set $\mathcal{D}_u$ is used for training the classifiers under attack, and the sets $\hat{\mathcal{D}}_b$ and $\hat{\mathcal{D}}_u$ are used by the attacker to train surrogate classifiers.

- **Limited Knowledge #2**: similarly to the above, but we consider a scenario where the attacker does not have full access to the feature extraction function (that induces the space $\mathcal{X}$). In particular, we consider a scenario where the attacker does not have access to the CNN model that was used to extract the features, but trains its own CNN (with identical training procedure and architecture) on a different set of users. In this case, $\theta_{LK2} = (\hat{\mathcal{D}}_b, \hat{\mathcal{D}}_u, \hat{\mathcal{X}}, f, \hat{w})$.

V. Experimental Protocol

We conducted experiments using the datasets MCYT-75 [30] (with 75 users), CEDAR [31] (55 users), GPDS-160 [32] (160 users) and the Brazilian PUC-PR [33] (60 users).

In order to simulate the different attack scenarios we split the dataset into two parts of disjoint users, as illustrated in Figure 3. The set $\mathcal{D}$ refers to the users "enrolled in the system", that will be under attack. This dataset is divided in training (user signatures $\mathcal{D}_u$) and testing $\mathcal{T}$. For the limited knowledge scenarios, we consider a set $\hat{\mathcal{D}}_b$ that contains signatures from other users (not those being attacked), simulating the scenario of an attacker that acquired his own signature dataset in order to generate the attacks. We also consider that the attacker has access to some signatures from the user, $\hat{\mathcal{D}}_u$, that were not used for training the system (i.e. disjoint from $\mathcal{D}_u$ and $\mathcal{T}$).

The images were pre-processed in a similar way to [34]: Signatures were first centered in a blank canvas using their center of mass. We then resize the images to 150×220 pixels and invert the image such that the background pixels are zero-valued. Lastly, we run the OTSU algorithm to identify the optimal threshold that separates background and foreground. We set the pixels with intensity smaller than the threshold to intensity 0, leaving the remaining pixels in grayscale.

We consider Writer-Dependent classifiers, training an SVM (linear or with the RBF kernel) for each user. As feature extraction $\phi(X)$, we consider: (i) a CNN-based learned representation: SigNet [34], and (ii) the CLBP operator (Completed Local Binary Patterns)[35]. We train the SVMs with 5 genuine signatures from the user as positive samples, and 5 signatures from each other user as negative.

For the scenario LK2, we consider two CNN models with the same architecture and training procedures, but trained on a disjoint set of users. The CNN used by the model under attack

was trained on GPDS users 350-614, and the CNN used by the surrogate models (by the attacker) were trained with users 615-881. Training procedure followed the same as SigNet [34]. For the Ensemble Adversarial Learning evaluation, we first trained two models with different architectures (slight variations from SigNet, as described in the Supplemental Material) and then trained a model with the SigNet architecture and the loss function defined in equation 5, with $\epsilon = 5$. For the Madry defense, we also used the same architecture, and trained with $\mathcal{S} = \{\delta : \|\delta\|_2 < \epsilon\}$ with $\epsilon = 2$. We tried using larger values for ϵ and obtained worse classification performance during the CNN training, so these values represent a tradeoff between robustness and accuracy. In both cases, we trained the network with users 350-614, to enable evaluating the scenario LK2. In this scenario, we consider an attacker that trained a regular CNN (no adversarial training), with users 615-881.

After training the classifiers for each user, the SVMs implement the following decision functions:

$$s_{\text{Linear}} = \mathbf{w}^\intercal \phi(X) + b \tag{7}$$

$$s_{\text{RBF}} = \sum_{i \in \mathcal{S}} \alpha_i \exp(-\gamma \|\phi(X) - X_i\|) + b \tag{8}$$

Where s_{Linear} and s_{RBF} are the scores for the linear SVM and the SVM with the RBF kernel, respectively; w are the weights learned by the linear SVM, $\mathcal{S}$ is the set of support vectors, α_i and X_i are the coefficients and support vectors, γ is a hyperparameter for the RBF kernel and b is the bias. We can easily see that both functions are differentiable with respect to $\phi(X)$ [26]. For the classifier using a CNN-based model to extract the features, we can calculate the gradients of the scores w.r.t the inputs X, and apply gradient-based methods to generate the attacks. For non-differentiable feature extractors, we consider only the two gradient-free methods described in section III-A. When reporting the scores in Figures 2, 4 and 5, we consider a normalized loss as follows: $\tilde{s}(X) = s(X) - \tau$, where τ is the global threshold. This makes it easy to identify if a signature would be classified as genuine or as a forgery ($\tilde{s}(X) \geq 0$ indicates the prediction of X being a genuine signature).

For the classifiers using LBP, we consider the the operator CLBP_S/M/C [35] (3D histogram of CLBP S, M and C), with the following parameters: $R = 1$ (radius of 1 pixel), $P = 8$ (eight neighbors) and rotation invariant uniform patterns ("riu2"). The feature vector has a total of 200 dimensions.

To simplify the generation of the attacks we considered a global threshold for the classifications, that obtained the Equal Error Rate on the set $\mathcal{D}$ (without any attacks).

After the classifiers are trained, we generate attacks using the four methods described in section III-A. We used the FGM method with $\epsilon = 1000$, and the Carlini & Wagner attack with $\kappa = 1$. For the Decision-based attack, we considered the implementation from the authors[1], running the attack for a maximum of 1000 iterations. For the Simulated Annealing method, we considered an open implementation of simulated annealing[2]. In each iteration, we change the state by adding gaussian noise ϵ ($\epsilon \sim \mathcal{N}(0, \sigma I)$, with $\sigma = 2$), and clipping the image between 0 and 255. We consider the energy to be a mixture of the SVM score and the L_2 norm of the adversarial noise δ: $E = s(X) + \lambda \|\delta\|_2$, with $\lambda = 0.001$ being a trade-off between changing the SVM score, and not deviating too far from the original image. We used an initial temperature $T_{\text{max}} = 1$ and final temperature $T_{\text{min}} = 0.001$. These values were chosen such that around 95% of the steps that would increase the energy are still accepted in the start of the procedure, and less than 5% were accepted in the end. We ran this procedure with at most 1000 steps, with early stopping (we stop optimization if the image is adversarial).

[1]https://github.com/bethgelab/foolbox

TABLE I: Results of WD classifiers using different feature sets (EER considering skilled forgeries)

Dataset	Features	EER global-τ		EER user-τ	
		Linear	RBF	Linear	RBF
MCYT-75	SigNet	7.12	7.03	7.39	5.68
	CLBP	26.49	27.03	27.21	26.85
CEDAR	SigNet	12.03	11.82	6.01	4.52
	CLBP	28.01	21.36	23.95	16.39
GPDS	Signet	7.70	6.80	4.62	4.14
	CLBP	26.74	24.58	21.79	22.37
Brazilian PUC-PR	SigNet	6.78	5.22	3.61	2.67
	CLBP	26.83	19.61	24.61	16.83

The experiments consisted in Type-I attacks (attempting to have a genuine signature rejected by the system) and Type-II attacks (attempting to have a forgery accepted by the system). For each user, we selected one genuine signature, one random forgery and one skilled forgery, such that all four classifiers correctly classified these samples. We then used the different attack methods to generate adversarial samples, and measured the attack success rate (number of misclassified images after the attack), and the average RMSE (root mean square error) of the adversarial noise on successful attacks. It is worth noting that we consider pixel values in the range $[0, 255]$, so the RMSE of the adversarial noise is also constrained in the same range. To summarize the experiments, we considered:

- Datasets: MCYT-75, CEDAR, GPDS-160, Brazilian PUC-PR
- Feature extractor: CLBP, SigNet
- SVM type: Linear, RBF
- Attack method: Decision-based, Simulated Annealing, FGM, Carlini
- Attacker's goal: Type-I (attacking Genuine signatures) and Type-II (attacking Random and Skilled forgeries),
- Attacker's knowledge: Perfect Knowledge, Limited Knowledge LK1 and LK2
- Defense: No defense, Ens. Adv. training, Madry

It is worth mentioning that in this work we did not consider the discretization of the generated adversarial images. We worked with images in float format, instead of discretized into integers between 0 and 255. This is discussed in section VI-F.

TABLE II: Success rate of Type-I attacks (% of attacks that transformed a genuine signature in a forgery)

Feature	Classifier	Attack Type FGM	Carlini	Anneal	Decision
CLBP	Linear	-	-	63.16	80.70
CLBP	RBF	-	-	100.00	100.00
SigNet	Linear	99.42	100.00	98.83	100.00
SigNet	RBF	98.25	100.00	98.83	100.00

TABLE III: Distortion (RMSE of the adversarial noise) for successful Type-I attacks

Feature	Classifier	Attack Type FGM	Carlini	Anneal	Decision
CLBP	Linear	-	-	0.40	1.57
CLBP	RBF	-	-	0.36	10^{-9}
SigNet	Linear	4.04	1.35	5.69	3.27
SigNet	RBF	4.06	1.40	5.17	3.02

VI. Results and discussion

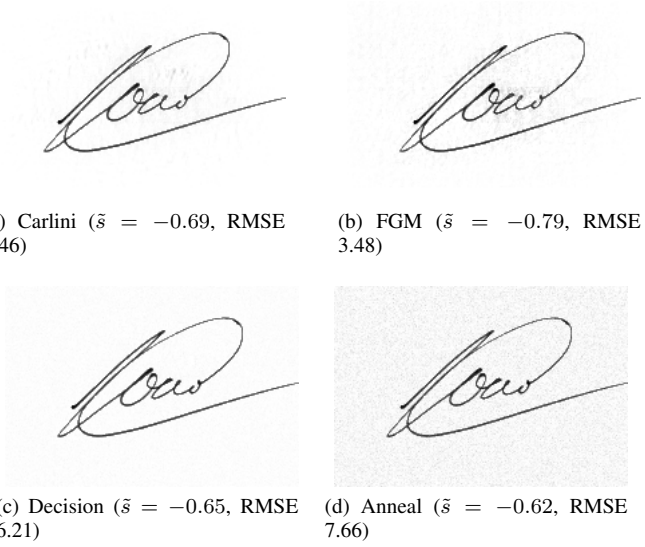
Before presenting the results of the attacks, we first validate the performance of the WD classifiers on the four datasets. Table I shows the EER obtained by using different features/classifiers, when trained with 5 reference signatures per user, with the protocol defined in section V. We observe a large variance in the results across different datasets, which suggests different degrees of difficulty on separating genuine signatures and forgeries in them. We also observe a large difference of performance between systems trained with the SigNet and CLBP features. In order to have a fair analysis of the adversarial examples against each classifier/feature extractor, we select the same set of images for the attacks on all classifiers, ensuring that the original images (before the attack) were correctly classified by them. Although the classifier performance varies across different datasets, the results for the adversarial attacks showed consistent trends across them. In this paper we report the consolidated results over the four datasets, and for completeness we include the results on individual datasets in the Supplementary Material.

A. Perfect Knowledge

We consider first a scenario of Perfect Knowledge, in which the adversary has full knowledge of all components of the system: $\theta_{PK} = (\mathcal{D}_b, \mathcal{D}_u, \mathcal{X}, f, w)$. The attacker can run his own copy of the system, and use one of the proposed attacks to generate adversarial images.

For Type-I attacks, given a genuine sample X_g, the objective is to obtain an adversarial $\tilde{X} = X_g + \delta$ that is classified as a forgery. Table II shows the success rate of attacks in this scenario (i.e. the percentage of attacks that found an adversary image), by attack type and classifier type. We see a high success rate for most attacks. Table III shows the average RMSE (root mean squared error) of the adversarial noise δ. We notice that the required amount of noise varies significantly with different classifiers and attack types. In general, gradient-based attacks find adversarial images with much less noise on the differentiable models. For the models with handcrafted features (where we do not have gradients), we noticed that even smaller changes on the image were enough to induce a misclassification. Figures 4 and 5 present examples of this type of attack.

[2]https://github.com/perrygeo/simanneal



(a) Carlini ($\tilde{s} = -0.69$, RMSE 2.46)

(b) FGM ($\tilde{s} = -0.79$, RMSE 3.48)

(c) Decision ($\tilde{s} = -0.65$, RMSE 6.21)

(d) Anneal ($\tilde{s} = -0.62$, RMSE 7.66)

Fig. 4: Example of Type-I attacks on the SVM model with RBF kernel and SigNet features. The original image is correctly classified as genuine by this model ($\tilde{s} = 0.13$).

(a) Decision ($\tilde{s} = -0.39$, RMSE 0.99)

(b) Anneal ($\tilde{s} = -0.27$, RMSE 0.21)

Fig. 5: Example of Type-I attacks on the SVM model with Linear kernel and CLBP features. The original image is correctly classified as genuine by this model ($\tilde{s} = 1.60$).

We now consider Type-II attacks, in which we want to modify a forgery X_f, by creating an adversary $\tilde{X} = X_f + \delta$ that is classified as a genuine signature. Table IV shows the

TABLE IV: Success rate of Type-II attacks (% of attacks that transformed a forgery in a genuine signature)

Features	Classifier	Forgery Type	Attack Type FGM	Carlini	Anneal	Decision
CLBP	Linear	random	-	-	37.36	45.98
		skilled	-	-	38.73	46.24
CLBP	RBF	random	-	-	0.00	0.00
		skilled	-	-	0.00	0.00
SigNet	Linear	random	1.15	96.55	0.00	0.00
		skilled	28.90	99.42	2.31	3.47
SigNet	RBF	random	0.57	94.83	0.00	0.00
		skilled	19.65	100.00	1.73	1.73

TABLE V: Distortion (RMSE of the adversarial noise) for successful Type-II attacks

Features	Classifier	Forgery Type	Attack Type FGM	Carlini	Anneal	Decision
CLBP	Linear	random	-	-	0.39	1.17
		skilled	-	-	0.42	1.08
SigNet	Linear	random	4.11	6.07	-	-
		skilled	4.20	3.19	3.61	1.34
SigNet	RBF	random	4.70	6.55	-	-
		skilled	4.08	3.62	3.17	1.18

success rate of the different methods, and table V shows the level of noise required in the successful attacks. The results show that this attack is much harder to obtain compared to the Type-I attacks. For the models trained with CLBP features, we observed that the linear classifier could be attacked half of the time, while we could not generate any attack using the two gradient-free methods for the non-linear model. For the CNN-based models, a strong gradient-based method (Carlini) worked for almost all samples, while the gradient-free methods did not work in most cases - we only observed some success when using skilled forgeries as the starting point. Comparing tables III and V, we observe that for the CLBP-based classifiers, a similar amount of noise was required to create successful attacks. For the CNN-based methods, when starting from a random forgery a large amount of noise was required to create successful attacks, while when starting from a skilled forgery a lower amount of noise was required. We reiterate that the skilled forgeries selected for attack were correctly classified by the model (i.e. classified as forgeries), while in successful attacks the adversarial image is classified as a genuine.

It is worth noting that in the experiments with the strong gradient-based attack, we observed a much larger amount of noise required for misclassification compared to previous results reports on object recognition. For instance, in the classification task on ImageNet, successful attacks (using the same Carlini & Wagner method) are reported with much lower noise (RMSE of 0.004 for 100% success of targeted attacks on ImageNet [8]). While for object recognition the adversarial images are often perceptually identical to the original, for signatures we noted some distinguishable noise, specially on the Type-II attacks, as can be seen in figure 2 (where the Type-II attack has RMSE of 10.34).

B. Limited Knowledge #1

We now consider a limited knowledge scenario, where the attacker does not have access to the signatures used for training the system, but does obtain a surrogate dataset: $\theta_{LK1} = (\hat{\mathcal{D}}_b, \hat{\mathcal{D}}_u, \mathcal{X}, f, \hat{w})$. In this case, the signatures from the *background set* (used as negative samples during training) were from a different set of users than those used to train the system. We also consider that the attacker collected some signatures from the user of interest $\hat{\mathcal{D}}_u$, but that are also different from those used to train the system. This scenario also assumes that the attacker knows the feature extractor (i.e. full knowledge of the feature extractor, including all parameters), and the learning function (the WD classifier type, but not the learned parameters). In this scenario, the attacker uses the surrogate data to train their own version of the WD classifiers, and uses this classifier to generate the attacks. We then evaluate the success rate of these attacks on the actual system.

TABLE VI: Success rate of Type-I attacks (% of attacks that transformed a genuine signature in a forgery) (Limited Knowledge)

Feature	Classifier	Attack Type FGM	Carlini	Anneal	Decision
CLBP	Linear	-	-	42.69	43.86
CLBP	RBF	-	-	82.46	82.46
SigNet	Linear	97.08	80.12	50.88	40.35
SigNet	RBF	97.66	91.81	54.39	47.95

TABLE VII: Success rate of Type-II attacks (% of attacks that transformed a forgery in a genuine signature) (Limited Knowledge)

Features	Classifier	Forgery Type	Attack Type FGM	Carlini	Anneal	Decision
CLBP	Linear	random	-	-	24.71	28.74
		skilled	-	-	21.97	26.01
CLBP	RBF	random	-	-	0.00	0.00
		skilled	-	-	0.00	0.00
SigNet	Linear	random	0.00	46.55	0.00	0.00
		skilled	22.54	71.68	1.73	0.00
SigNet	RBF	random	0.00	74.71	0.00	0.00
		skilled	19.08	83.24	0.58	0.00

Table VI shows the success rate of the Type-I attacks. We observe a lower success rate compared to the perfect knowledge scenario, but still we find a high success rate against most models. This suggests that indeed there is a transferability of attacks across models (as observed before in CNNs [4]), and that this transferability also impacts systems trained with handcrafted features.

Table VII shows the success rate for Type-II attacks in a limited knowledge scenario. Again we see a drop in performance compared to the perfect knowledge scenario, but still the attacks that worked in the PK scenario also worked (to some extent) in the limited knowledge scenario.

C. Limited Knowledge #2

We now consider a limited knowledge scenario similar to the above, but where the attacker also does not have access to the CNN used to extract the features. In this case, we consider that the attacker trains a surrogate CNN using a disjoint set of users, which induces a new feature space $\hat{\mathcal{X}}$. We consider therefore $\theta_{LK2} = (\hat{\mathcal{D}}_b, \hat{\mathcal{D}}_u, \hat{\mathcal{X}}, f, \hat{w})$.

TABLE VIII: Success rate of Type-I attacks (% of attacks that transformed a genuine signature in a forgery) (Limited Knowledge #2)

Feature	Classifier	Attack Type FGM	Carlini	Anneal	Decision
SigNet	Linear	60.34	6.90	48.85	19.54
SigNet	RBF	64.37	9.20	51.15	18.97

TABLE IX: Success rate of Type-II attacks (% of attacks that transformed a forgery in a genuine signature) (Limited Knowledge #2)

Features	Classifier	Forgery Type	Attack Type FGM	Carlini	Anneal	Decision
SigNet	Linear	random	0.00	0.00	0.00	0.00
		skilled	2.30	2.30	0.57	0.57
SigNet	RBF	random	0.00	0.00	0.00	0.00
		skilled	1.72	1.72	1.15	0.00

Tables VIII IX show the success rate of the Type-I and Type-II attacks, respectively. We observe much lower success rates, especially for Type-II attacks, where no attacks were successful when starting from a random forgery, and starting with a skilled forgery the success was as low as 1-2%. For the Type-I attacks, we notice lower success rates compared to the PK and LK1 scenarios. Overall, these results suggest that transferability of the attacks is much worse when the models are trained with a different subset of users, that is, when the attacker does not have access to signatures from the same users that were used to train the CNN model. This contrasts with findings in object classification, where attacks trained on a subset of data transfer well to a model trained with another subset of data (different samples from the same classes) [4]. Also, it is worth noting that the strong Carlini attack (that achieves close to 100% success in the Perfect Knowledge scenario) drops in performance in the LK scenarios, confirming previous findings that such iterative attacks transfer less than single-step attacks such as FGM [36].

D. Evaluating countermeasures

We now consider the impact of two counter-measures for the CNN-based systems: Ensemble Adversarial Learning [7] and the Madry defense [10]. Tables X and XI show the success rate and distortion (RMSE) for Type-I attacks. We consider the three Knowledge scenarios discussed in section IV-A (Perfect Knowledge and two Limited Knowledge scenarios), and the two gradient-based attacks (FGM and Carlini). We notice that both defenses provide some robustness against the FGM attack in all knowledge scenarios. Considering the Carlini attack, we see that in a Perfect-Knowledge scenario the attack was always successful, but Table XI shows that the Madry defense greatly increase the amount of noise required to generate adversarial examples, going from a RMSE of 1.4 to around 3.3.

Tables XII and XIII shows the results on Type-II attacks. In these experiments, we again observe that the Carlini attack finds attacks most of the time, and that the Madry defense showed to be effective in increasing the amount of noise required to obtain an adversarial example (e.g. the average RMSE is increased from 5.98 to 10.81 when starting with a random forgery, comparing the baseline and the Madry defense). It is worth noting that the RMSE values only consider the successful attacks, and therefore the results on the Limited Knowledge scenarios (where the success rate is very low) are likely skewed by a few forgeries that were already close to the decision boundary.

E. Impact of background removal

We now investigate the impact of simple noise-reduction techniques on the success of the attacks. Starting from the adversarial examples found in the experiments from the previous section, we applied the OTSU algorithm to remove noise with intensity lower than a threshold (as described in section V). We then evaluate if the resulting image remains adversarial.

Tables XIV and XV evaluate the impact of processing the adversarial images with OTSU on the success rate of the attacks, for Type-I and Type-II attacks, respectively. We noticed that this pre-processing was effective against the gradient-free attacks, and provided some reduction in the success rate using gradient-based attacks. A possible explanation for this difference is that on the gradient-free methods used in these experiments, only small changes to a random set of pixels in done in each iteration, while the gradient-based methods can select larger changes to a smaller set of pixels (the regions where we have a large gradient of the loss w.r.t to the pixels).

F. Limitations and practical considerations

In this work we evaluated different attack scenarios (knowledge and capabilities for the attacker), but we would like to highlight some practical aspects to take into consideration for actual attacks:

- **Discretization**: In this work, we use images in floating point representation, which is appropriate for the optimization methods. Images are commonly stored in 8-bits per channel (i.e. pixels intensities that are integer values $X_{ij} \in \{0, ..., 255\}$). Simply rounding the pixel intensities to the nearest integer degrades the quality of adversarial examples. An alternative is to conduct a greedy search (changing each pixel at a time and checking if the image is still adversarial). This solution is computationally intensive, but can solve the problem (Carlini et al. [8] reported success with this search - i.e. by using this method, the discretized version of an adversarial image is still adversarial, for all images). For figures 4 and 5 we used the discretized images (and reported the score and RMSE using the discretized version of the images), so this step mainly adds more computational complexity for the attacker.
- **Physical Attacks**: We considered only attacks using digital images (i.e. after the sensor acquisition) which are limited for scenarios where digital images are used: services where the client provides a digital image (e.g. an app where the user scans a picture of a bank cheque). It has been shown that physical attacks are possible [16], [37], where adversarial images were printed, subsequently captured using a camera, and still fooled classifiers. However, this often requires more noise to be added, to account to transformations such as slight rotations or translations of the image. Also, it is worth noting that, if noise is printed on top of a handwritten signature, the noise δ needs to be constrained to be positive. In some early experiments in this scenario, we found it to also require more noise (50% higher RMSE) than if δ does not have this constraint.

TABLE X: Success rate of Type-I attacks considering different defenses and attacker knowledge scenarios

		Attack Type and Knowledge scenario					
		FGM			Carlini		
Defense	Classifier	PK	LK1	LK2	PK	LK1	LK2
Baseline	Linear	100.00	95.40	60.34	100.00	78.16	6.90
	RBF	100.00	97.70	64.37	100.00	85.63	9.20
Ens. Adv	Linear	91.38	85.63	45.40	100.00	79.89	4.60
	RBF	90.23	83.91	46.55	100.00	90.23	5.75
Madry	Linear	91.38	83.33	22.99	100.00	74.71	1.72
	RBF	89.08	86.21	21.84	100.00	89.08	0.57

TABLE XI: Distortion (RMSE of the adversarial noise) for Type-I attacks, considering different defenses and attacker knowledge scenarios

		Attack Type and Knowledge scenario					
		FGM			Carlini		
Defense	Classifier	PK	LK1	LK2	PK	LK1	LK2
Baseline	Linear	4.17	4.19	4.30	1.31	1.33	1.37
	RBF	4.20	4.21	4.30	1.40	1.38	1.55
Ens. Adv.	SigNet & Linear	4.37	4.30	4.20	1.35	1.43	1.85
	RBF	4.36	4.32	4.20	1.44	1.43	1.63
Madry	SigNet & Linear	4.76	4.72	4.26	3.19	3.28	1.59
	RBF	4.77	4.74	4.27	3.48	3.52	2.19

TABLE XII: Success rate of Type-II attacks considering different defenses and attacker knowledge scenarios

			Attack Type and Knowledge scenario					
			FGM			Carlini		
Defense	Classifier	Forgery Type	PK	LK1	LK2	PK	LK1	LK2
Baseline	Linear	random	2.87	1.15	0.00	98.85	42.53	0.00
		skilled	40.80	29.31	2.30	100.00	66.67	2.30
	RBF	random	1.72	1.15	0.00	95.98	68.39	0.00
		skilled	34.48	27.59	1.72	100.00	83.91	1.72
Ens. Adv.	Linear	random	1.72	0.57	0.00	93.10	41.38	0.00
		skilled	29.31	14.94	1.15	100.00	64.37	3.45
	RBF	random	2.30	0.00	0.00	93.10	69.54	0.00
		skilled	22.99	17.24	1.15	100.00	83.91	2.30
Madry	Linear	random	1.72	0.57	0.00	98.28	45.98	0.00
		skilled	48.85	38.51	8.05	100.00	73.56	3.45
	RBF	random	2.30	0.57	0.00	97.70	75.86	0.00
		skilled	45.98	37.36	6.32	100.00	87.93	2.87

TABLE XIII: Distortion (RMSE of the adversarial noise) for Type-II attacks, considering different defenses and attacker knowledge scenarios

			Attack Type and Knowledge scenario					
			FGM			Carlini		
Defense	Classifier	Forgery Type	PK	LK1	LK2	PK	LK1	LK2
Baseline	Linear	random	3.97	3.75	-	5.98	5.71	-
		skilled	4.21	4.14	4.24	2.99	2.71	2.43
	RBF	random	3.84	3.83	-	6.27	6.03	-
		skilled	4.11	4.06	4.64	3.32	3.20	1.77
Ens. Adv.	Linear	random	4.51	4.82	-	8.61	8.83	-
		skilled	4.53	4.58	4.09	4.71	4.34	1.43
	RBF	random	4.40	-	-	9.45	9.31	-
		skilled	4.59	4.58	4.07	5.43	4.82	2.14
Madry	Linear	random	4.74	5.38	-	10.81	10.97	-
		skilled	4.90	4.93	4.15	6.18	5.87	1.94
	RBF	random	4.62	5.28	-	11.49	11.46	-
		skilled	4.91	4.88	4.16	7.00	6.71	2.40

TABLE XIV: Success of Type-I attacks in a PK scenario, with no pre-processing and with OTSU pre-processing

		Attack Type and Preprocessing							
		FGM		Carlini		Anneal		Decision	
Feature	Classifier	None	OTSU	None	OTSU	None	OTSU	None	OTSU
CLBP	Linear	-	-	-	-	63.16	9.36	80.70	3.51
	RBF	-	-	-	-	100.00	0.58	100.00	0.00
SigNet baseline	Linear	100.00	88.51	100.00	18.39	96.55	0.57	100.00	1.72
	RBF	100.00	86.21	100.00	22.41	98.28	0.00	98.85	0.57
SigNet Ens. Adv.	Linear	91.38	67.24	100.00	2.87	97.70	0.00	100.00	0.00
	RBF	90.23	65.52	100.00	1.72	98.28	0.00	100.00	0.00
SigNet Madry	Linear	91.38	87.93	100.00	77.01	87.93	0.00	99.43	6.90
	RBF	89.08	87.36	100.00	75.86	88.51	0.00	100.00	5.75

TABLE XV: Success of Type-II attacks in a PK scenario, with no pre-processing and with OTSU pre-processing

			Attack Type and Preprocessing							
			FGM		Carlini		Anneal		Decision	
Feature	Classifier	Forgery Type	None	OTSU	None	OTSU	None	OTSU	None	OTSU
CLBP	Linear	random	-	-	-	-	37.36	0.57	45.98	0.00
		skilled	-	-	-	-	38.73	1.73	46.24	1.16
	RBF	random	-	-	-	-	0.00	0.00	0.00	0.00
		skilled	-	-	-	-	0.00	0.00	0.00	0.00
SigNet Baseline	Linear	random	2.87	0.57	98.85	0.00	0.00	0.00	0.00	0.00
		skilled	40.80	31.03	100.00	12.07	1.15	0.00	1.15	0.00
	RBF	random	1.72	0.57	95.98	0.00	0.00	0.00	0.00	0.00
		skilled	34.48	24.14	100.00	14.37	1.72	0.00	1.72	0.00
SigNet Ens Adv.	Linear	random	1.72	1.15	93.10	7.47	0.00	0.00	1.72	0.00
		skilled	29.31	22.99	100.00	21.84	0.57	0.00	2.87	0.57
	RBF	random	2.30	1.15	93.10	12.64	0.00	0.00	0.00	0.00
		skilled	22.99	14.94	100.00	27.59	0.57	0.00	0.57	0.00
SigNet Madry	Linear	random	1.72	1.15	98.28	45.40	0.00	0.00	1.72	0.00
		skilled	48.85	43.10	100.00	77.01	0.57	0.00	2.87	0.57
	RBF	random	2.30	1.72	97.70	62.64	0.00	0.00	0.00	0.00
		skilled	45.98	40.23	100.00	84.48	0.57	0.00	0.57	0.00

- **Knowledge of noise-removal**: In section VI-E, we considered a pre-processing step to remove noise, that is effective (to some extent) in many scenarios. We note, however, that this cannot be considered a robust defense, and that if the adversary is aware of it, it can use this information as part of generating the adversarial images (e.g. knowing that a threshold τ is used, consider adding only pixels with intensity larger than τ). This still increases the difficulty for gradient-based methods, since the problem becomes discontinuous (the pixel intensities can be 0 or greater than τ).

VII. Conclusion

In this paper we investigated the impact of adversarial examples on biometric systems, in particular by identifying threats to Offline Handwritten Signature Verification under the point of view of Adversarial Machine Learning. Our experiments indicate that the issue of adversarial examples present new threats to such systems in several scenarios, including both systems using handcrafted feature extractors and systems that learn directly from image pixels. In particular, we identify that Type-I attacks (changing a genuine signature so that it is rejected by the system) were successful is all systems investigated, even in a limited knowledge scenario, where the attacker does not have access to the signatures used for training the writer-dependent classifiers. The results in this scenario confirm previous findings that attacks transfer across different CNN classifiers [4], and show that this transferability is also present on attacks on systems using a handcrafted feature extractor (CLBP). We found, however, that transferability is greatly reduced when the CNN is trained with a different set of *users* (rather than a dijsoint set of *samples* from the same classes, as investigated in [4]). We identified that Type-II attacks (changing a forgery to be accepted as genuine) are much harder to craft, obtaining lower success rates overall, and requiring larger amounts of noise for the strong gradient-based method. This contrasts with results in object recognition literature, where successful attacks (even in a targeted setting) are reported with much lower noise (less than 3 orders of magnitude), that are commonly visually imperceptible. [8]

Lastly, we investigated some countermeasures for this problem, and confirmed previous findings that the Madry defense [10] increase the amount of noise necessary to generate adversarial images. In this paper, we show that this defense is effective even when only applied on the feature learning phase, with no changes to the subsequent WD classifier training. We do note, however, that in spite of the increased amount of noise required, a strong attack (Carlini) is able to find adversarial examples most of the time. Our experiments with noise reduction show that this can reduce the success rate of attacks when the attacker is not aware of the defense, although we reiterate that this cannot be considered a robust defense

(the adversary can incorporate this knowledge on the attack generation process). A definitive solution for this issue is yet an open research problem. Exploring the nature of the signal (a pen trajectory in 2D space) as part of the defense can be a promising direction for defenses. Another interesting area for future work is analyzing the impact of physical attacks (e.g. by printing adversarial noise on top of a signature).

REFERENCES

[1] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 1, pp. 4–20, 2004.

[2] N. K. Ratha, J. H. Connell, and R. M. Bolle, "An analysis of minutiae matching strength," in *International Conference on Audio-and Video-Based Biometric Person Authentication*. Springer, 2001, pp. 223–228.

[3] B. Biggio, g. fumera, P. Russu, L. Didaci, and F. Roli, "Adversarial Biometric Recognition : A review on biometric system security from the adversarial machine-learning perspective," *IEEE Signal Processing Magazine*, vol. 32, no. 5, pp. 31–41, Sep. 2015.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in *International Conference on Learning Representations*, 2015.

[6] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Security and Privacy, IEEE Symposium on*. IEEE, 2016, pp. 582–597.

[7] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. McDaniel, "Ensemble Adversarial Training: Attacks and Defenses," in *International Conference on Learning Representations*, 2018.

[8] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 2017, pp. 39–57.

[9] ——, "Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017.

[10] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," *International Conference on Learning Representations*, 2018.

[11] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*. ACM, 2006, pp. 16–25.

[12] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," *Machine Learning*, vol. 81, no. 2, pp. 121–148, 2010.

[13] B. Biggio, G. Fumera, and F. Roli, "Security evaluation of pattern classifiers under attack," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 4, pp. 984–996, 2014.

[14] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, Dec. 2018.

[15] A. K. Jain, K. Nandakumar, and A. Nagar, "Biometric Template Security," *EURASIP J. Adv. Signal Process*, vol. 2008, pp. 113:1–113:17, Jan. 2008.

[16] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *International Conference on Learning Representations (workshop track)*, 2017.

[17] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical Black-Box Attacks Against Machine Learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ser. ASIA CCS. ACM, 2017, pp. 506–519.

[18] W. Brendel, J. Rauber, and M. Bethge, "Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models," in *International Conference on Learning Representations*, 2018.

[19] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks Without Training Substitute Models," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017, pp. 15–26.

[20] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box Adversarial Attacks with Limited Queries and Information," in *International Conference on Learning Representations*, 2018.

[21] A. Ramanathan, L. Pullum, Z. Husein, S. Raj, N. Torosdagli, S. Pattanaik, and S. K. Jha, "Adversarial attacks on computer vision algorithms using natural perturbations," in *Contemporary Computing (IC3), 2017 Tenth International Conference on*. IEEE, 2017, pp. 1–6.

[22] J. F. Vargas, M. A. Ferrer, C. M. Travieso, and J. B. Alonso, "Off-line signature verification based on grey level information using texture features," *Pattern Recognition*, vol. 44, no. 2, pp. 375–385, Feb. 2011.

[23] J. Hu and Y. Chen, "Offline Signature Verification Using Real Adaboost Classifier Combination of Pseudo-dynamic Features," in *Document Analysis and Recognition, 12th International Conference on*, Aug. 2013, pp. 1345–1349.

[24] M. B. Yılmaz and B. Yanıkoğlu, "Score level fusion of classifiers in off-line signature verification," *Information Fusion*, vol. 32, Part B, pp. 109–119, Nov. 2016.

[25] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma, "Adversarial Classification," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '04. New York, NY, USA: ACM, 2004, pp. 99–108.

[26] 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 *Machine Learning and Knowledge Discovery in Databases*, 2013, pp. 387–402.

[27] C. Guo, M. Rana, M. Cissé, and L. van der Maaten, "Countering Adversarial Images using Input Transformations," in *International Conference on Learning Representations*, 2018.

[28] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples," *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[29] A. Athalye and N. Carlini, "On the Robustness of the CVPR 2018 White-Box Adversarial Example Defenses," in *The Bright and Dark Sides of Computer Vision: Challenges and Opportunities for Privacy and Security (CVPR workshop)*, 2018.

[30] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J.-J. Igarza, C. Vivaracho, and others, "MCYT baseline corpus: a bimodal biometric database," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 150, no. 6, pp. 395–401, 2003.

[31] M. K. Kalera, S. Srihari, and A. Xu, "Offline signature verification and identification using distance statistics," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 07, pp. 1339–1360, Nov. 2004.

[32] J. Vargas, M. Ferrer, C. Travieso, and J. Alonso, "Off-line Handwritten Signature GPDS-960 Corpus," in *Document Analysis and Recognition, 9th International Conference on*, vol. 2, Sep. 2007, pp. 764–768.

[33] C. Freitas, M. Morita, L. Oliveira, E. Justino, A. Yacoubi, E. Lethelier, F. Bortolozzi, and R. Sabourin, "Bases de dados de cheques bancarios brasileiros," in *XXVI Conferencia Latinoamericana de Informatica*, 2000.

[34] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Learning features for offline handwritten signature verification using deep convolutional neural networks," *Pattern Recognition*, vol. 70, pp. 163–176, Oct. 2017.

[35] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *Image Processing, IEEE Transactions on*, vol. 19, no. 6, pp. 1657–1663, 2010.

[36] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial Machine Learning at Scale," in *International Conference on Learning Representations*, 2017.

[37] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing Robust Adversarial Examples," in *International Conference on Machine Learning*, 2018.