2018 IEEE European Symposium on Security and Privacy

Forgotten Siblings: Unifying Attacks on Machine Learning and Digital Watermarking

Erwin Quiring, Daniel Arp and Konrad Rieck

Technische Universität Braunschweig
Brunswick, Germany

***Abstract*—Machine learning is increasingly used in security-critical applications, such as autonomous driving, face recognition, and malware detection. Most learning methods, however, have not been designed with security in mind and thus are vulnerable to different types of attacks. This problem has motivated the research field of *adversarial machine learning* that is concerned with attacking and defending learning methods. Concurrently, a separate line of research has tackled a very similar problem: In *digital watermarking*, a pattern is embedded in a signal in the presence of an adversary. As a consequence, this research field has also extensively studied techniques for attacking and defending watermarking methods.**

The two research communities have worked in parallel so far, unnoticeably developing similar attack and defense strategies. This paper is a first effort to bring these communities together. To this end, we present a unified notation of black-box attacks against machine learning and watermarking. To demonstrate its efficacy, we apply concepts from watermarking to machine learning and vice versa. We show that countermeasures from watermarking can mitigate recent model-extraction attacks and, similarly, that techniques for hardening machine learning can fend off oracle attacks against watermarks. We further demonstrate a novel threat for watermarking schemes based on recent deep learning attacks from adversarial learning. Our work provides a conceptual link between two research fields and thereby opens novel directions for improving the security of both, machine learning and digital watermarking.

1. Introduction

In the last years, machine learning has become the tool of choice in many areas of engineering. Learning methods are not only applied in classic settings, such as speech and handwriting recognition, but increasingly operate at the core of security-critical applications. For example, self-driving cars make use of deep learning for recognizing objects and street signs [e.g., 34, 70]. Similarly, systems for surveillance and access control often build on machine learning methods for identifying faces and persons [e.g. 50, 56]. Finally, several detection systems for malicious software integrate learning methods for analyzing data more effectively [e.g., 33, 35].

Machine learning, however, has originally not been designed with security in mind. Many learning methods suffer from vulnerabilities that enable an adversary to thwart their successful application—either during the training or prediction phase. This problem has motivated the research field of *adversarial machine learning* which is concerned with the theory and practice of learning in an adversarial environment [31, 45, 62]. This led to several attacks and defenses, e.g. for poisoning support vector machines [8, 9], crafting adversarial examples against neural networks [10, 43, 44] or stealing models from online services [59].

Concurrently to adversarial machine learning, a different line of research has faced very similar problems: In *digital watermarking* a pattern is embedded in a signal, such as an image, in the presence of an adversary [17]. This adversary seeks to extract or remove the information from the signal, thereby reversing the watermarking process and obtaining an unmarked copy of the signal, for example, for illegally distributing copyrighted content. As a consequence, methods for digital watermarking naturally operate in an adversarial environment and several types of attacks and defenses have been proposed for watermarking methods, such as sensitivity and oracle attacks [e.g., 1, 14, 16, 25].

Unfortunately, the two research communities have worked in parallel so far and unnoticeably developed similar attack and defense strategies. To illustrate this similarity, let us consider the simplified attacks shown in Figure 1: The middle plot corresponds to an *evasion attack* against a learning method, similar to the attacks proposed by Papernot et al. [43, 44]. A few pixels of the target image have been carefully manipulated, such that the digit 5 is misclassified as 8. By contrast, the right plot shows an *oracle attack* against a watermarking method, similar to the attacks developed by Westfeld [67] and Cox & Linnartz [16]. Again, a few pixels have been changed; this time, however, to mislead the watermark detection in the target image.

Target image with watermark. | Evasion attack (Misclassified as 8). | Oracle attack (Broken watermark).

Figure 1. Examples of attacks against machine learning and digital watermarking. Middle: the target is modified, such that it is misclassified as 8. Right: the target is modified, such that the watermark is destroyed.

DOI 10.1109/EuroSP.2018.00041

While both attacks address different goals, the underlying attack strategy is surprisingly similar. In fact, both attacks aim at minimally modifying the target, such that a decision boundary is crossed. In the case of machine learning, this boundary separates different classes, such as the digits. In the case of digital watermarking, the boundary discriminates watermarked from unmarked signals. Although the previous example illustrates only a single attack type, it becomes apparent that there is a conceptual similarity between learning and watermarking attacks.

In this paper, we strive for bringing these two research fields together and systematically study the similarities of black-box attacks against learning and watermarking methods. To this end, we introduce a unified notation for these attacks, which enables us to reason about their inner workings and abstract from the concrete attack setting. This unified view allows for transferring concepts from machine learning to digital watermarking and vice versa. As a result, we are able to apply defenses originally developed for watermarks to learning methods as well as transferring machine learning defenses to digital watermarking.

We empirically demonstrate the efficacy of this unified view in three case studies. First, we use deep learning concepts from adversarial learning [55] to attack the advanced watermarking scheme Broken Arrows [25]. Second, we show that techniques for hardening machine learning with classifier diversity [5] can be successfully applied to block oracle attacks against watermarks. Third, we show that stateful defenses from digital watermarking can effectively mitigate model-extraction attacks against decision trees [59]. In addition, we provide further examples of attacks and defenses, transferable between the research fields. By doing so, we establish several links between the two research fields and identify novel directions for improving the security of both, machine learning and digital watermarking.

In summary, we make the following contributions:

- *Machine learning meets digital watermarking.* We present a novel formal view on black-box attacks against learning and watermarking methods that exposes previously unknown similarities between both research fields.
- *Transfer of attacks and defenses.* Our unified view enables transferring concepts from machine learning to digital watermarking and vice versa, giving rise to novel attacks and defenses.
- *Three case studies.* Based on our unified view, we demonstrate a novel attack against watermarking schemes. Furthermore, we present two novel defenses derived from our unified view to hinder model-extraction attacks and oracle attacks.

The rest of this paper is organized as follows: In Section 2 we review the background of adversarial machine learning and digital watermarking. We introduce our unified view on black-box attacks in both research fields in Section 3 and present case studies in Section 4. We discuss the implications of our work in Section 5 and conclude in Section 6.

2. Background

Whenever machine learning or digital watermarking are applied in security-critical applications, one needs to account for the presence of an attacker. This adversary may try to attack the learning/watermarking process and thereby impact the confidentiality, integrity, and availability of the application. This section provides a basic introduction to the motivation and threat scenarios in *machine learning* and *digital watermarking*, before Section 3 systematizes them under a common notation. A reader familiar with one of the two fields may directly proceed to Section 3.

2.1. Adversarial Machine Learning

Machine learning has become an integral part of many applications in computer science and engineering, ranging from handwriting recognition to autonomous driving. The success of machine learning methods is rooted in its capability to automatically infer patterns and relations from large amounts of data [see 22, 30]. However, this inference is usually not robust against attacks and thus may be disrupted or deceived by an adversary. These attacks can be roughly categorized into three classes: *poisoning attacks*, *evasion attacks* and *model extraction*. The latter two attacks are the focus of our work, as they have concrete counterparts in the area of digital watermarking.

Evasion attacks. In this attack setting, the adversary attempts to thwart the prediction of a trained classifier and evade detection. To this end, the attacker carefully manipulates characteristics of the data provided to the classifier to change the predicted class. As a result, the attack impacts the *integrity* of the prediction. For example, in the case of spam filtering, the adversary may omit words from spam emails indicative for unsolicited content [37]. A common variant of this attack type are *mimicry attacks*, in which the adversary mimics characteristics of a particular class to hinder a correct prediction [23, 53]. Evasion and mimicry attacks have been successfully applied against different learning-based systems, for example in network intrusion detection [24, 53], malware detection [29, 54, 69] and face recognition [51].

Depending on the adversary's knowledge about the classifier, evasion attacks can be conducted in a *black-box* or *white-box* setting. In the black-box setting, no information about the learning method and its training data is available and the adversary needs to guide her attack along the predicted classes of the classifier [38, 43, 63]. With increasing knowledge of the method and data, the probability of a successful evasion rises [6]. In such a white-box setting, the adversary may exploit leaked training data to build a surrogate model and then determine what feature combinations have the most effect on the prediction.

Model extraction. In this attack setting, the adversary actively probes a learning method and analyzes the returned output to reconstruct the underlying learning model [38]. This attack, denoted as *model extraction* or *model stealing*,

impacts the *confidentiality* of the learning model. It may allow the adversary to gain insights on the training data as well as obtain a suitable surrogate model for preparing evasion attacks.

Depending on the output, the adversary operates in either a *black-box* or *gray-box* setting. If only the predicted classes are observable, extracting the learning model is more challenging, whereas if function values are returned or learning parameters are available, the adversary can more quickly approximate the learning model. For example, the recent attacks proposed by Tramèr et al. [59] enable reconstructing learning models from different publicly available machine learning services in both settings. Moreover, model extraction poses a serious risk to the privacy of users, as an attacker may derive private information from the reconstructed model [52].

2.2. Digital Watermarking

Digital watermarking allows for verifying the authenticity of digital media, like images, music or videos. Digital watermarks are frequently used for copyright protection and identifying illegally distributed content [66]. Technically, a watermark is attached to a medium by embedding a pattern into the signal of the medium, such that the pattern is *imperceptible* and *inseparable*. A particular challenge for this embedding is the robustness of the watermark, which should persist under common media processing, such as compression and denoising. There exist several approaches for creating robust watermarks and we refer the reader to the comprehensive overview provided by Cox et al. [17].

As an example, Figure 2 shows a simple watermarking scheme where a random pattern is added to the pixels of an image. The induced changes remain (almost) unnoticeable, yet the presence of the watermark can be detected by correlating the watermarked image with the original watermark. Appendix A illustrates this simple watermarking scheme in more detail.

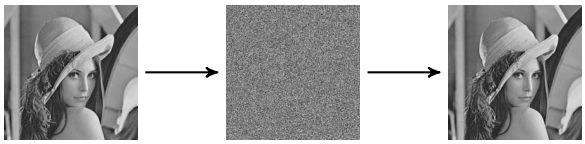

Figure 2. Example of a digital watermark. A random noise pattern is added to the image in the spatial domain. The pattern is not observable but detectable.

Similar to machine learning, watermarking methods need to account for the presence of an adversary and withstand different forms of attacks [16, 25]. While there exist several attacks based on information leaks and embedding artifacts that are unique to digital watermarking [e.g., 4, 17], we identify two attack classes that correspond to black-box evasion and model-extraction attacks.

Oracle attacks. In this attack scenario, the adversary has access to a watermark detector that can be used to check whether a given media sample contains a watermark [16]. Such a detector can be an online platform verifying the authenticity of images as well as a media player that implements digital rights management. Given this detector, the attacker can launch an *oracle attack* in which she iteratively modifies a watermarked medium until the watermark is undetectable. The attack thus impacts the *integrity* of the pattern embedded in the signal.

While it is trivial to destroy the pattern and the coupled signal, for example using massive changes to the medium, carefully removing the watermark while preserving the original signal is a notable challenge. As a consequence, a large variety of different attack strategies has been proposed [e.g., 14, 16, 18, 32]. A prominent example is the *Blind Newton Sensitivity Attack*, where no prior knowledge about the detector's decision function is required and which has been successfully applied against several watermarking schemes (see Appendix B).

Watermark estimation. In the second attack setting, the adversary also has access to a watermark detector, yet her goal is not only to remove the watermark from a target medium but to estimate its pattern [12, 41]. The attack thus impacts the *confidentiality* of the watermark and not only allows removing the pattern from the signal but also enables forging the watermark onto arbitrary other data. This *watermark estimation* therefore represents a considerable threat to watermarking methods, as it can undermine security mechanisms for copyright protection and access control.

3. Unifying Adversarial Learning and Digital Watermarking

It is evident from the previous section that attacks against learning and watermarking methods share some similarities—an observation that has surprisingly been overlooked by the two research communities [2]. Throughout this section, we systematically identify the similarities and show that it is possible to transfer knowledge about attacks and defenses from one field to the other. An overview of this systematization is presented in Figure 3. We guide our systematization of machine learning and digital watermarking along the following five concepts:

1) *Data Representation.* Machine learning and watermarking make use of similar data representations, which enables putting corresponding learning and detection methods into the same context (Section 3.1)
2) *Problem setting.* Watermarking can be seen as a special case of a binary classification. Consequently, binary classifiers and watermarking techniques tackle a similar problem (Section 3.2).
3) *Attacks.* Due to the similar representation and problem setting, attacks overlap between both fields, as we discuss for evasion attacks (Section 3.3) and model extraction (Section 3.4).

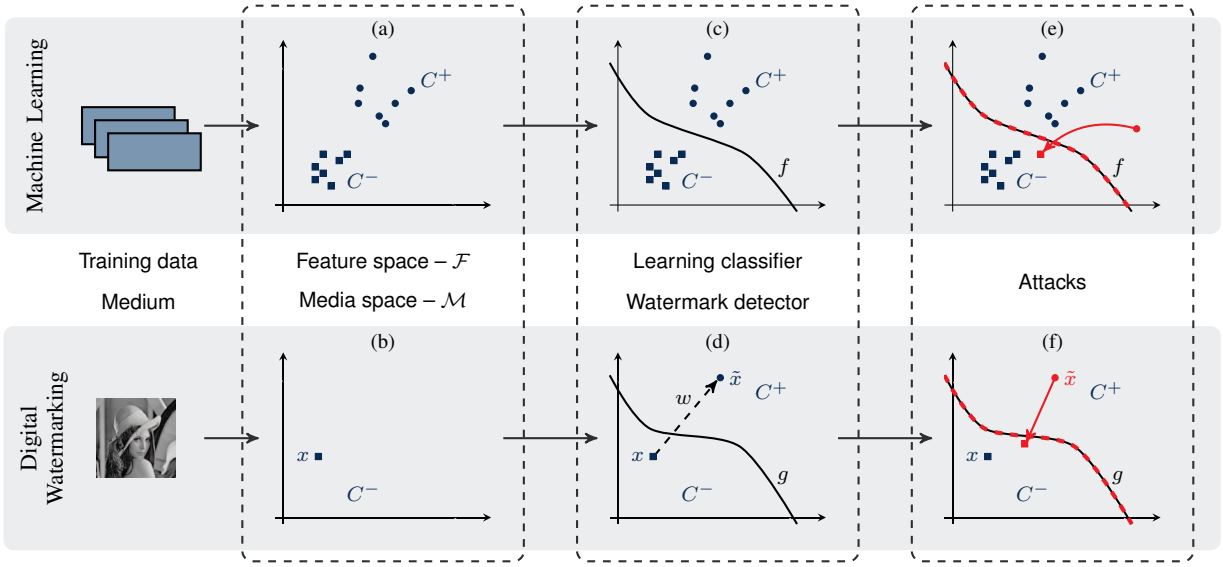

Figure 3. A unified view on machine learning and digital watermarking. Top: A machine learning setup including a feature space, a learning classifier and corresponding attacks. Bottom: A watermarking setup including the media space, the watermark detector and corresponding attacks. The red dashed line illustrates model extraction/watermark estimation, while the red arrow shows an evasion attack/oracle attack.

4) *Defenses.* Defenses developed in one research field often fit the corresponding attack in the other field and thus can be transferred due to the similar data representation and problem setting (Section 3.5).
5) *Differences.* Both fields naturally exhibit differences that—together with the similarities—yield a clear picture of both research fields (Section 3.6).

In the following, we discuss each of these concepts in more detail, where we first formalize the concept for machine learning and then proceed to digital watermarking.

3.1. Feature Space vs. Media Space

Machine learning. Learning methods typically operate on a so-called *feature space* $\mathcal{F}$ that captures the characteristics of the data to be analyzed and learned. These features usually correspond to vectors $x \in \mathbb{R}^N$ and in the case of *classification* are assigned to a class label y that needs to be learned and predicted, such as C^+ and C^- in Figure 3(a). Note that feature spaces in machine learning can also be constructed implicitly, for example using non-linear maps and kernel functions [22, 49]. Yet, a representation is often possible through vectors.

Digital watermarking. Similar to machine learning, watermarking methods operate on a signal available in some underlying *media space* $\mathcal{M}$, such as the pixels of an image or the audio samples of a recording. Without loss of generality, this signal can be described as a vector $x \in \mathbb{R}^N$ and thus the media space corresponds to the feature space used in machine learning: $\mathcal{F} \cong \mathcal{M}$. Note that advanced watermarking schemes often map the signal to other spaces, such as frequency or random subspace domains [17, 25]. Still, the mapped signals can be described as points in a vector space.

3.2. Classifier vs. Watermark Detector

Machine learning. After embedding the training data into a feature space, the actual learning process is performed using a learning method, such as a support vector machine or a neural network. In the case of classification, this learning method tries to infer functional dependencies from the training data to separate data points of different classes. These dependencies are described in a learning model θ that parameterizes a decision function $f(x)$. Given a vector x, the function $f(x)$ predicts a class label based on an underlying decision boundary in the vector space:

$$f : \mathcal{F} \longmapsto \{-1, 1\} \; . \tag{1}$$

We focus on binary classification for the sake of simplicity, but the discussed concepts are also applicable to multi-class classifiers.

Digital watermarking. The media space in watermarking is divided into two separate subspaces as depicted in Figure 3(d) where the marked and unmarked versions of the signal represent the two classes. Note that a robust watermark should ideally survive image processing steps, such as compression and denoising. Therefore, the watermark class implicitly contains variations as well, just as machine learning captures the variations of samples from a class through its generalization.

If we denote an unmarked signal as x and a watermarked signal as $\tilde{x}$, the relation between x and $\tilde{x}$ is given by a parameter w that defines the pattern of the watermark. As a

consequence, a watermark detector also employs a function $g(x)$ to determine which subspace a signal is in and thus whether it contains the watermark:

$$g: \mathcal{M} \longmapsto \{-1, 1\} . \quad (2)$$

Similar to machine learning, the function g may induce a linear or non-linear decision boundary, such as a polynomial [26] or fractalized surface [40].

Although the functions f and g share similarities, the creation process of the underlying decision boundary fundamentally differs. In machine learning, the boundary needs to separate the training data as good as possible which restricts the boundary's shape to existing data. In contrast, the boundary in watermarking schemes can be created under more degree's of freedom as long as the underlying watermark is reliably detectable. After that the boundary is created, an attacker, however, faces the same situation in both fields. As Figures 3(c)–(d) highlight, a decision boundary divides the vector space into two—not necessarily the same—subspaces:

$$\mathcal{F} = \left\{x \in \mathbb{R}^N | f(x) = y^-\right\} \cup \left\{x \in \mathbb{R}^N | f(x) = y^+\right\} \quad (3)$$

$$\mathcal{M} = \left\{x \in \mathbb{R}^N | g(x) = y^-\right\} \cup \left\{x \in \mathbb{R}^N | g(x) = y^+\right\} \quad (4)$$

Consequently, black-box attacks that work through input-output observations are transferable between machine learning and digital watermarking. In the following sections, we discuss this similarity and provide a mapping between machine learning and watermarking attacks, which lays the ground for transferring defenses from one field to the other.

3.3. Evasion Attack vs. Oracle Attack

As the first attack mapping, we consider the pair of *evasion* and *oracle* attacks in a black-box setting. In this attack scenario, an adversary targets the integrity of the classifier's/detector's response by inducing a misclassification from an iteratively collected set of input-output pairs.

Machine learning. In an evasion attack, the adversary tries to manipulate a sample with minimal changes, such that it is misclassified by the decision function f. Formally, the attack can thus be described as an optimization problem,

$$\arg\min_t d(t) \text{ s.t. } f(x+t) = y^* , \quad (5)$$

where $d(t)$ reflects the necessary changes t on the original sample x to achieve the wanted prediction y^*.

Digital Watermarking. In an oracle attack, an adversary tries to disturb or even remove the watermark embedded in a medium. The attack setting is closely related to evasion. Formally, the underlying optimization problem is given by

$$\arg\min_t d(t) \text{ s.t. } g(\tilde{x}+t) = y^- , \quad (6)$$

where $d(t)$ reflects the changes t on the watermarked signal $\tilde{x}$ and y^- corresponds to no detection.

Machine learning $\leftrightarrow$ Digital Watermarking. The optimization problems in Eq. (5) and Eq. (6) are equivalent. In geometrical terms, this allows similar attack strategies in both fields whenever the adversary aims at crossing the decision boundary in the vector space towards the wanted class based on binary outputs only (see Figure 3(e)–(f)). Note that the black-box strategy generally does not depend on whether two or more classes are used. The attacker's objective is to cross one boundary towards a selected target class—which can be one of many.

We group the attack strategies into *direct attacks* and *transferability-based attacks*. In the first category, the attacker directly uses the binary classifier output to construct an evasive sample. The watermarking literature has extensively developed strategies to find samples in this way [e.g., 14, 16, 18, 32]. For example, the Blind Newton Sensitivity Attack [14] computes the decision boundary's first and second derivative by observing how the detector's output varies for minimal changes at a decision boundary location. This attack is straight applicable against learning classifiers when we replace the binary output g with f. Appendix B recaps the attack in more detail. Thus, researchers should reuse these existing watermark oracle attacks when attacking a classifier directly. In the context of adversarial learning, Dang et al. [20] introduced a novel strategy that allows the application of genetic programming to find evasive samples even if binary outputs are given only. This refinement is straight applicable to the watermarking field by replacing f with g and may foster novel attacks based on genetic programming.

The second type of attack strategy is based on the transferability property: an evasive sample that misleads a substitute model—calculated by the adversary—will probably mislead the original model as well [42, 43]. Due to the same attack objective and the same geometrical structure, such a strategy is also possible against watermarking schemes: An adversary learns a substitute model to approximate the watermark's decision function and performs a subsequent evasion attack on that model instead of the watermark detector. In this way, the adversary can exploit the full access to the model to apply white-box attacks. We demonstrate the efficacy of this novel attack against watermarking schemes in a practical case study in Section 4.

3.4. Model Extraction vs. Watermark Estimation

As the second attack mapping, we consider the pair of *model extraction* and *watermark estimation*. In the black-box scenario, the adversary aims at compromising the confidentiality of a learning model or digital watermark by sending specifically crafted objects to a given classifier/detector and observing the respective binary output over multiple iterations.

Machine learning. Model-extraction attacks center on an effective strategy for querying a classifier, such that the underlying model can be reconstructed with few queries. In contrast to evasion, the extraction of the learning model θ

enables the adversary to apply this model to arbitrary data. For instance, Tramèr et al. [59] have recently demonstrated this threat by stealing models from cloud platforms providing machine learning as a service. Geometrically, the adversary's goal can be described as finding a function $\hat{f}$ such that its decision boundary is as close as possible to the original one of f. Formally, we adapt the closely matching measure from Tramèr et al. [59] and describe the attack as

$$\frac{1}{|\mathcal{U}|} \sum_{x \in \mathcal{U}} d\left(f(x), \hat{f}(x)\right) \longrightarrow 0 \tag{7}$$

where d represents the 0-1 distance and $\mathcal{U}$ represents, for instance, a uniformly chosen set from $\mathcal{F}$.

Digital watermarking. Watermark estimation represents the counterpart to model extraction. In this attack scenario, the adversary seeks to reconstruct the watermark from a marked signal $\tilde{x}$. If successful, the adversary is not only capable of perfectly removing the watermark from the signal $\tilde{x}$, but also of embedding it in other signals, thereby effectively creating forgeries. We describe this attack again by reconstructing the decision boundary in the media space:

$$\frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} d\left(g(x), \hat{g}(x)\right) \longrightarrow 0 \tag{8}$$

where d represents the 0-1 distance and $\mathcal{X}$ can represent a uniformly chosen set from $\mathcal{M}$.

Machine learning $\leftrightarrow$ Digital Watermarking. While learning models and watermarks are conceptually very different, Eq. (7) and Eq. (8) emphasize that the extraction of an underlying decision boundary in a vector space represents a common adversarial goal in both research fields.

A commonly used attack strategy in both fields consists in localizing the decision boundary through a line search and then combining various gathered points to reconstruct the model or watermark precisely [12, 38, 41]. The extraction of non-linear classifiers such as decision trees also exploits localized boundary points for reconstruction [59]. We discuss the latter attack in more detail in Section 4 when presenting a novel defense against it, inspired by concepts from digital watermarking.

The second group of attacks builds on an approximation of the decision boundary and has been primarily investigated in adversarial machine learning. An attacker collects a number of input-output pairs with queries either scattered over the feature space or created adaptively [42, 43, 59]. These observations allow the adversary to learn a substitute model. As previously described, this approach can be part of an evasion attack. Our unified view underlines that an adversary can also substitute a watermarking scheme by a learning model that approximates the decision boundary of g. In this way, the attacker is able to remove or add digital watermarks by using white-box attack strategies from adversarial learning. Our case study in Section 4 examines this novel threat in more detail.

TABLE 1. TRANSFER OF DEFENSE TECHNIQUES INTRODUCED BY ADVERSARIAL LEARNING AND DIGITAL WATERMARKING.

	Defense Technique	Adv. Learning			Watermark.
Random.	Multiple Classifier/Detector	[7, 46, 63]			[11, 60]
	Rand. Boundary Region		◁		[25, 36]
	Union of Watermarks		◁		[25]
Complex.	Non-Linearity	[5, 48]	◁	▷	[40]
	Classifier Diversity	[5]		▷	(CS2)
	Snake Traps		◁		[25]
Stateful.	Closeness-To-The-Boundary	(CS3)	◁		[1, 57]
	Line Search Detection		◁		[1, 58]
	Locality-Sensitive Hashing		◁		[61]

◁ = Possible transfer from watermarking to machine learning;
▷ = Possible transfer from machine learning to watermarking;
CS2, CS3 = Defense transfer demonstrated as case study in Section 4;

3.5. Defenses

The communities of both research fields have extensively worked on developing defenses to fend off the attacks presented in the previous sections. However, it is usually much easier to create an attack that compromises a security goal, than devising a defense that effectively stops a class of attacks. As a result, several of the developed defenses only protect from very specific attacks and it is still an open question how learning methods and watermark detectors can generally be protected from the influence of an adversary. As the previous sections highlight, an attacker geometrically works at the decision boundary in both fields, so that various defense strategies are not restricted to one particular research field. In this section, we formally describe these similarities and outline the implications as means of novel research directions (see Table 1). We also include defenses from adversarial learning that were initially presented against informed adversaries, but also work when an adversary acts in a black-box setting.

Randomization. A simple yet effective strategy to impede attacks against classifiers and watermark detectors builds on the introduction of randomness. While this defense cannot rule out successful attacks, the induced indeterminism obstructs simple attack strategies and requires more thorough concepts for evasion or model extraction.

The application of *multiple learning classifiers or watermark detectors* represents a common implementation of this defense strategy. In machine learning, each classifier can be built from a random subset of the feature set [7, 46, 63]. The binary prediction is then retrieved from the numerical output of all classifiers F_i through some aggregation function E:

$$f(x) = E\left(F_1(x), F_2(x), \ldots, F_k(x)\right). \tag{9}$$

A corresponding strategy has been examined against oracle attacks. The binary prediction is obtained from the numerical

output of several detectors G_i where each is built from a random subset of pixels. The final detector output is then obtained from some aggregation function T, for instance the median, which yields [11, 60]:

$$g(x) = T\left(G_1(x), G_2(x), \ldots, G_k(x)\right). \tag{10}$$

A comparison of Eq. (9) and (10) reveals that both fields employ a similar defense strategy with the same intention: An adversary has to attack different classifiers/detectors at the same time and cannot be sure whether a specific feature/pixel influences the returned output.

However, the watermarking literature has already discussed weaknesses of this defense by creating a so-called *p-boundary* that acts as a surrogate boundary [13]. As a result, this attack also needs to be considered in the machine learning context if randomization defenses are used against black-box attacks.

The watermarking literature also provides further randomization defenses. For example, a detector may return arbitrary outputs within a randomized region around the decision boundary [25, 36]. However, this approach is also vulnerable to a surrogate boundary [13] and thus should be used in machine learning with great care. Moreover, the *Broken Arrows* watermarking scheme creates several watermarks that form a *union of watermarks*. During detection, only the watermark with the smallest distance to the current signal is applied [25]. This mitigates the risk that an adversary could compare multiple images with the same watermark. This defense has not been applied to learning methods yet. It would correspond to an ensemble of classifiers where the aggregation function E just chooses one classifier depending on the input sample.

In general, we conclude that existing and novel randomization strategies based on Eq. (9) and (10) are transferable between machine learning and watermarking. Moreover, such a unified view also allows the identification of weaknesses that researchers have already examined in the other field, for example the p-boundary against randomized regions.

Complexity. Another defense strategy consists in increasing the complexity of $\mathcal{F}$ or $\mathcal{M}$ such that an attacker has to invest more resources to exploit the decision boundary. However, this is not trivial, as a fine-grained boundary, for example, may enable new attacks [e.g. 44]. Both fields have focused on different strategies which provides an opportunity for transferring knowledge.

First, recent work on adversarial machine learning proposes to enclose the learned data tightly. In the case of malware detection, this implies that an evasion attack needs to contain plausible features of the benign class without losing the malicious functionality. Russu et al. implement this defense strategy using non-linear kernel functions [48], while Biggio et al. realize a tighter and more complex boundary through the combination of two-class and one-class models [5]. Although invented against informed attackers with a surrogate model, these countermeasures also tackle black-box attacks that need to probe the feature space with queries outside the training data distribution. We demonstrate in Section 4.2 that this strategy also addresses a watermark oracle attack, where an adversary may also probe the detector with artificial inputs [68].

Increasing the decision boundary's complexity represents another strategy. A linear boundary, for instance, can be replaced by a fractalized version along the previous boundary so that the boundary cannot be estimated with a finite number of known points [40]. In addition, Furon and Bas have introduced small indents called *snake traps* at the decision boundary in order to stop attacks based on random walks along the detection region [18, 25]. These defenses are applicable in machine learning as well, as they replace an existing boundary by a more complex version that lies along the previous boundary. In this way, the learned separation of the classifier is not changed and black-box attacks are obstructed.

Stateful analysis. If the learning method or watermark detector is outside of the attacker's control, an active defense strategy becomes possible, in which the defender seeks to identify sequences of malicious queries. For instance, a cloud service providing machine learning as a service may monitor incoming queries for patterns indicative of evasion and model-extraction attacks.

While this concept has not yet been examined in adversarial machine learning, stateful analysis of queries has been successfully applied in digital watermarking for detecting oracle and watermark-estimation attacks [1, 57, 58, 61]. These defenses exploit the fact that an adversary will typically follow a specific strategy to locate the decision boundary due to the inherent binary output restriction. For example, an adversary may use a line search to localize the boundary or perform several queries close to the boundary in order to exactly locate its position. Formally, we obtain a new detector that is based on a meta-detector m that works alongside the usual decision function $g(x_t)$:

$$g_m(x_t) = \Psi\left(g(x_t),\, m(x_t, x_{t-1}, \ldots, x_{t-l})\right). \tag{11}$$

The meta-detector does not influence $g(x_t)$ and analyzes the sequence of the current and prior inputs $x_t, x_{t-1}, \ldots, x_{t-l}$ in parallel to infer whether the system is subject to an attack. Then, the function Ψ either forwards the true decision value $g(x_t)$ or initiates another defense if m detects an attack. For instance, it may return misleading outputs or block further access.

Due to the similar problem setting, adversaries follow the same attack strategy in machine learning. Thus, the proposed defense strategies from digital watermarking are directly applicable to machine learning. The meta-detector m can be reused and just the detection function g needs to be replaced by f. We show in a case study in Section 4 that model-extraction attacks can be mitigated with the closeness-to-the-boundary concept. We note that stateful defenses have already been applied to watermarking schemes (see Table 1), providing the opportunity for constructing novel defenses for learning methods.

3.6. Differences

For successfully transferring concepts between machine learning and watermarking, however, researchers also need to account for the difference of both areas. First, as described in Section 3.2, the decision boundary in machine learning needs to be adjusted to existing training data in contrast to digital watermarking. Thus, defenses from watermarking that introduce a completely new decision boundary [e.g. 26, 27] are not necessarily applicable to machine learning. Second, the white-box setting from machine learning where an attacker knows internals such as the model or fractions of the training data is not directly transferable to digital watermarking. If the original image or the watermark are known, an adversary has already succeeded. Third, specific attacks are unique to the respective field. Reconstructing the watermark as a noise signal from a set of images, for example by averaging images, is unique to digital watermarking [e.g. 17]. The poisoning scenario known from adversarial machine learning where the attacker manipulates a fraction of the training data [19] is in turn not transferable to digital watermarking.

In summary, machine learning and digital watermarking have different goals and the learning process with real-world data differs to the artificial watermark embedding process. Nevertheless, both operate in a corresponding vector space and, although the decision boundary can be different, the black-box scenario leads to a common attack surface: An adversary tries to change the vector subspace or to estimate the boundary just from binary outputs. Therefore, similar attack strategies and defenses are usable.

4. Transfer of Attacks and Defenses

Equipped with a unified notation for black-box attacks, we are ready to study the transfer of concepts from one research field to the other in practical scenarios. In our first case study, we apply concepts from adversarial learning to attack a state-of-the-art watermarking scheme. As the second case study, we apply a concept for securing machine learning to a watermark detector and demonstrate that the resulting defense mitigates an oracle attack. In the third case study, we apply the concept of closeness-to-the-boundary to machine learning and show that it blocks recent model-extraction attacks.

4.1. Case Study 1: ML → DW

In our first case study, we apply concepts from the area of adversarial learning against the watermarking scheme Broken Arrows. In particular, we construct a substitute learning model to approximate the watermark detector and subsequently perform an evasion attack on this model to obtain an unwatermarked version of an input image. In this way, we demonstrate that strategies—originally examined to evade image or malware classifiers—also threaten watermark detectors.

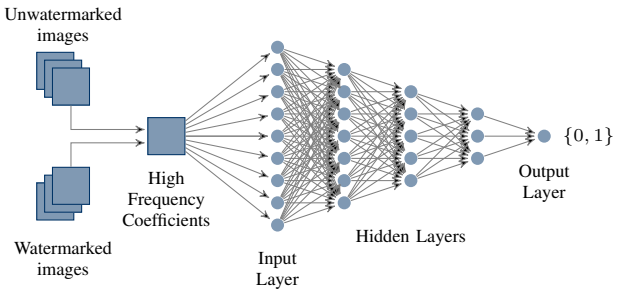

Figure 4. Transfer from machine learning to watermarking: A Deep Neural Network acts as a substitute model for a watermark detector.

We choose Broken Arrows from the second "Break Our Watermarking System" (BOWS) competition [25], as it represents a publicly available state-of-the-art watermarking scheme. In a nutshell, Broken Arrows first computes the high-frequency coefficients of an input image that are mapped to a secret 256-dimensional subspace. 30 watermark patterns are created in this subspace and the closest one to the input is further used. The selected watermark pattern is finally mapped back to the frequency space where it is added to the high-frequency coefficients of the original image, yielding the watermarked image.

We do not require the attacker to have detailed knowledge of the underlying watermarking scheme, except that the watermark is present in the high-frequency coefficients. Note that this assumption is not unusual in the watermarking context [68]. The attacker will consequently work with these coefficients, as an analysis is not distracted by the unused parts of an image. As the secret mapping induces a non-linear decision boundary in the frequency space, we choose a Deep Neural Network (DNN) for approximation, backed by the capability of DNN's to approximate various learning algorithms [42, 43].

The attacker's strategy is as follows: she first collects a number of images with the same watermarking key and their unwatermarked counterpart, for instance, by exploiting her oracle access to the watermark detector. The high-frequency coefficients of each image are used as training set so that the DNN finally learns to differentiate between watermarked and unwatermarked images. Figure 4 depicts this process schematically.

The resulting learning model acts as a substitute for the watermark detector. It allows the attacker to perform a local evasion attack based on function values instead of binary outputs. Similar to Szegedy et al. [55], we solve the following problem:

$$\text{minimize } c\|\tilde{z}+t\|_2 + \hat{F}(\tilde{z}+t)\,, \tag{12}$$

where $\tilde{z}$ represents the frequency coefficients of the marked image $\tilde{x}$, t the changes and $\hat{F}(\tilde{z})$ the network's real-valued output. We perform a gradient descent until the DNN predicts the watermark's absence with high confidence. Various values for c are tested to find a suitable balance between both optimization terms. Finally, the attacker can exploit the oracle

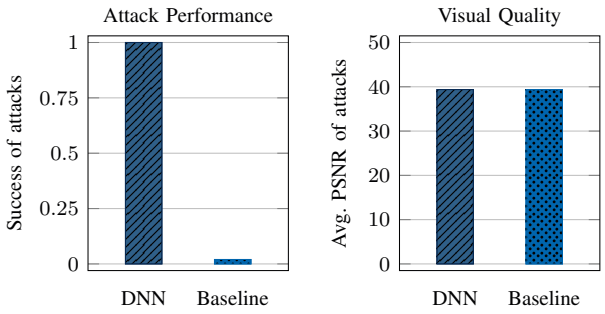

Figure 5. Attack Performance for our attack based on the substitute model (DNN) and the baseline.

access to the binary watermark detector to verify whether the resulting adversarial image leads to the wanted prediction and continue with the gradient descent on the substitute model if necessary.

Experimental Setup. As dataset for our evaluation, we consider the publicly available Raise Image Database [21] for our training and validation set and the Dresden Image Database [28] for our test set. All images are converted to grayscale, cropped to a common size of 96×96 with varying offsets, and marked with the same watermarking key. In total, our training set consists of 40,000 images, the validation set of 24,000 images and the test set of 30,000 images. Each set has the same number of marked and unmarked images.

A DNN is learned on the training set while the validation set is used to mitigate overfitting. We select 250 watermarked images from the test set where the attack solves Eq. (12) to find an unwatermarked version. We report results for our best combination of parameters. In particular, we report the number of successful watermark removals and the average Peak Signal to Noise Ratio (PSNR) between the original unwatermarked image and its adversarial counterpart as a visual quality metric. As a naive baseline, we perform an attack where random noise is added to the frequency coefficients of the same magnitude as the previously calculated gradient from Equation (12). We stop the baseline attack if the PSNR gets smaller than the DNN-based solution.

Attack Evaluation. Figure 5 presents the results of our attack. The adversary is able to make the watermark undetectable in 100% of the images, thereby demonstrating the efficacy of our attack. At the same time, the average PSNR is 39.38 dB with a standard deviation of 5.92 dB. These PSNR values are comparable to reported results during the 2nd BOWS contest [65]. Figure 6 additionally gives intuition about the resulting image quality. Furthermore, the baseline shows that random noise addition cannot destroy the watermark in the same order as our proposed attack does.

In our experiments, the best DNN architecture consists of 120, 30 and 5 neurons in the consecutive hidden layers. The attack can easily get stuck into local minima, so that the watermark is not removed or with a lower image quality than actually possible. A careful adjustment of various overfitting mechanisms and heuristics such as slightly varying starting positions reduce the likelihood of a local minimum.

Figure 6. The right plot shows the output of the attack where the watermark is not detected anymore. The PSNR between the original and the evasive sample is 33.88 dB.

Note that a manual attack against the watermarking scheme such as conducted by Westfeld or Bas [4] yields higher PSNR values. However, our attack is not limited to a specific watermarking scheme, as it automatically infers the underlying watermarking process. Overall, our attack demonstrates that an adversary with no background information and an oracle access is able to apply concepts from adversarial learning to attack a watermark detector successfully.

4.2. Case Study 2: ML → DW

In our second case study, we transfer a defense against evasion attacks from the area of machine learning. This defense increases the complexity of the decision boundary by combining a two-class and one-class classifier—a concept denoted as *1½-classifier* [5]. Instead of just discriminating objects into two classes, the defense additionally learns a one-class model for the underlying data distribution. The combined classifier discriminates two classes but also require all inputs to lie within the learned region of normality. As a result, evasion attacks become more difficult, as the adversary needs to stay within normal regions when locating and moving towards the decision boundary.

This simple yet effective idea has not been applied in the context of digital watermarking so far. While existing watermarking schemes provide an accurate detection of marked content, they ignore how signals are distributed in the media space and hence an adversary can explore the full space for exploring properties of the watermark. Broadly speaking, "the image does not have to look nice" in an attack [68] and thus attack points resemble distorted or implausible media. For example, many oracle attacks move along random directions or set pixels to constant values when locating the decision boundary [14] (see Figure 8).

We exploit this characteristic and introduce a *1½-detector* that identifies watermarks but additionally spots implausible signals, that is, inputs too far away from reasonable variations of the original signal. Our detector rests on the concept of Biggio et al. [5] and only provides a correct decision if the input lies within the learned region of normality. If signals outside the region are provided, the detector returns a *random decision*, thereby foiling attack strategies that move along random directions or use constant values. Figure 7 depicts this defense and the resulting combination of boundaries.

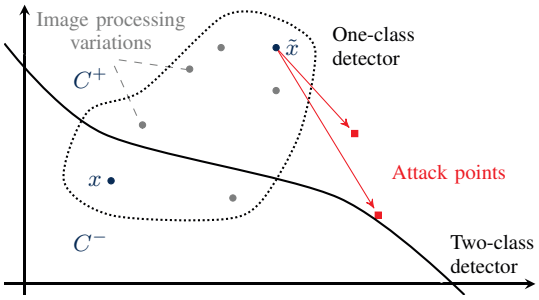

Figure 7. Transfer from machine learning to watermarking: A 1½-detector combining a one-class and two-class detection method.

To generate a suitable model of normality, the one-class model in the detector is trained with samples of common variations of the target signal. If the media space corresponds to images, different changes of brightness, scaling, contrast, compression and denoising can be applied to the target image $\tilde{x}$. Similarly, other plausible variations of the signal can be added into the one-class model. Figure 8 shows different variation of a target image that are correctly identified by the 1½-detector.

Experimental Setup. For our evaluation, we implement a 1½-detector using a linear watermarking scheme (see Appendix A) and a one-class model based on the neighborhood of a signal. Given an image $\tilde{x}$, this model computes the distance d to the k-nearest variation of $\tilde{x}$, that is,

$$d(\tilde{x}) = \frac{1}{D\,k} \sum_{z \in \mathcal{N}_{\tilde{x}}} \|z - \tilde{x}\| \tag{13}$$

where $\mathcal{N}_{\tilde{x}}$ are the k-nearest neighbors of $\tilde{x}$. For normalization purposes, we divide each distance by the maximum distance in the media space, D. The image is marked as implausible if the distance to its k-nearest variations reaches a given threshold δ. For our study, we simply fix $k = 3$.

As dataset for our evaluation, we consider 50 images from the Dresden Image Database [28]. All images are converted to grayscale and cropped to a common size of 128×128 pixels and tagged with a digital watermark. To obtain training data for the one-class model, we create different variations of the watermarked images by applying common image processing techniques, such as noise addition, denoising, JPEG compression and contrast/brightness variation.

To attack the marked images, we implement the Blind Newton Sensitivity Attack [3, 15], a state-of-the-art oracle attack that successfully defeats several existing defenses (see Appendix B). We launch the attack against the selected 50 images using different configurations of the 1½-detector and average results over the 50 runs, respectively.

Defense Evaluation. The results of this experiment are presented in Table 2. If no defense is deployed, the implemented oracle attack is capable of removing the watermark from all images, thereby demonstrating the efficacy of the Blind Newton Sensitivity Attack. However, if we enable the one-class model in our 1½-detector and pick a threshold below 0.31, the attack fails to remove the watermark in all cases. As our defense returns random decisions outside the normal regions, the attack is not able to compute the correct gradient and thus does not converge to the correct watermarking pattern. The correlation between the watermark extracted from the final attack outcome and the original watermark is thereby zero in all cases. A threshold $\delta \geq 0.31$ however enlarges the extent of the normal regions, so that the chances increase that the attack works on the decision boundary within the normal region without disturbance again.

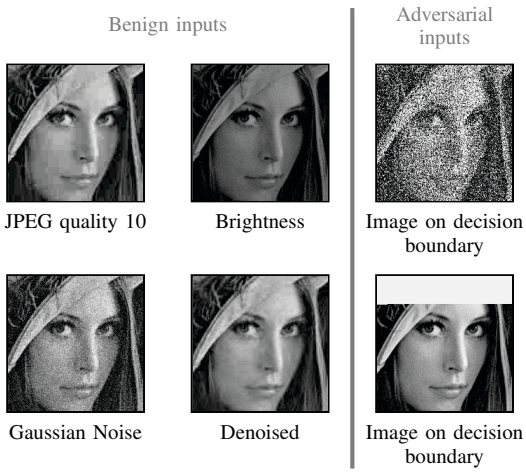

Figure 8. Distortions of the target image. The left four plots show plausible image distortions, whereas the right plots depict attack images.

TABLE 2. DETECTION PERFORMANCE OF THE 1½-DETECTOR.

Threshold	Success of attacks	False-positive rate
No defense	100%	—
$\delta = 0.46$	6%	0%
$\delta = 0.31$	6%	0%
$\delta = 0.23$	0%	0%
$\delta = 0.18$	0%	0%
$\delta = 0.12$	0%	0.14%
$\delta = 0.03$	0%	2.27%
$\delta = 0.02$	0%	4.47%

False Positives. We also inspect the false-positive rate induced by our detector. To this end, we use variations of the selected images that have not been used for training. If we pick a low threshold, the learned model is too restrictive and some of the generated variations lie outside the normal region. Starting with a threshold of 0.18, however, the defense does not identify any benign variations as attacks and thus allows us to separate legitimate variations of an image from malicious inputs generated by the Blind Newton Sensitivity Attack.

In summary, we identify a range of suitable thresholds where the detector does not misclassify benign variations and is successfully able to obstruct the watermark removal in all cases. The proposed defense is generally applicable by other watermarking schemes, because the objective is to spot

adversely crafted images without changing the underlying watermark detection process. Moreover, as our defense already impedes the initial boundary localization process which is not unique to the Blind Newton Sensitivity Attack, other oracle attacks [e.g. 12, 32] are affected as well.

4.3. Case Study 3: DW → ML

In our third case study, we transfer the concept of closeness-to-the-boundary from the area of digital watermarking to machine learning. In particular, we demonstrate that this defense effectively mitigates the risk of model extraction by identifying sequences of malicious queries to a learning method.

Before presenting this defense, we shortly summarize the tree-extraction attack proposed by Tramèr et al. [59]. The attack reconstructs decision trees by performing targeted queries on the APIs provided by the BigML service. The attack is possible, since the service does not only return the class label for a submitted query but also a confidence score for a particular leaf node. This enables an adversary to distinguish between the leaves. For each leaf and for each of its features, a recursive binary search locates the leaf's decision boundary in that direction. As the binary search covers the whole feature range, other leaf regions are discovered as well and extracted subsequently. In this way, an adversary can extract all possible paths of the decision tree. Note that the attack needs to fix all features except for the one of interest, as otherwise the attack may miss a leaf during the binary search.

As a countermeasure to this attack, we devise a defense that observes the closeness of queries to the decision boundary, as already used in digital watermarking [1, 57]. In this scenario, the detector does not only check for the presence of a watermark, but simultaneously counts the number of queries falling inside a margin surrounding the boundary. An attacker conducting an oracle attack—thereby working around the boundary necessarily—creates an unusually large number of queries inside this margin. As a result, the analysis of the input sequences allows the identification of unusual activity. The exact parameters of the security margin are derived from statistical properties of the decision function [1].

Although this defense strategy has been initially designed to protect watermark detectors, we demonstrate that it can be extended to secure decision trees as well. Figure 9 illustrates the transferred concept where margins are added to all boundaries of a decision tree. The width of these margins is determined for each region separately depending on the statistical distribution of the data. Overall, this security margin is defined alongside the original decision tree and does not require changes to its implementation. Appendix C provides more information on the margin's creation process.

When the decision tree returns the predicted class for a query, our defense checks whether the query falls inside the security margin. To determine whether the tree is subject to an attack, we keep track of a history of queries and compute the ratio between points inside and outside the margin for each leaf. The averaged ratio over all leaves,

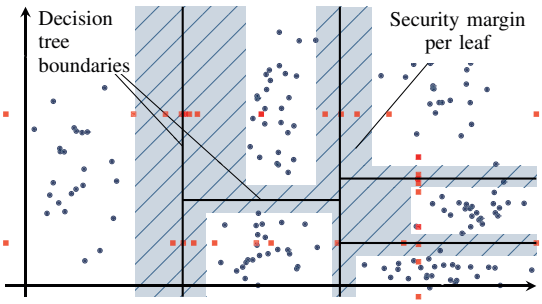

Figure 9. Transfer from watermarking to machine learning: A stateful defense using the closeness-to-the-boundary concept.

φ, is an indicator for the plausibility of the current input sequence. As an example, Figure 9 shows a typical query sequence from the tree extraction algorithm (red squared). The adversary has to work within the margin to localize the decision boundary, in contrast to the distribution of benign queries (blue circles).

Experimental setup. To evaluate this defense in practice, we use the publicly available tree-extraction implementation by Tramèr et al. [59]. Table 3 summarizes our used datasets. We divide each dataset into a training set (50%) and test set (50%), where we use the first for learning a decision tree and calibrating the security margins. The detector identifies an attack if the query ratio φ exceeds the threshold $\tau = 0.3$.

We make use of the test set to simulate the queries of an honest user. In this way, we can determine the risk of false positives, that is, declaring an honest input sequence as malicious. Next, we run the tree-stealing attack against the learned tree without and with the security margin defense. In the latter case, we consider two reactions after the detection of an attack sequence: (a) the tree blocks further access and (b) the tree returns random decisions. We stop an attack after 1 Million queries (denoted by *). We repeat each experiment 5 times and present aggregated results in the following.

Defense Evaluation. To determine the knowledge gain by the adversary, Table 4 reports the fraction of successfully extracted leaves p together with the required number of queries Q. Without any defense, the original attack extracts the whole tree ($p = 1$). In contrast, the blocking strategy based on the margin defense allows the tree to block the tree extraction at the very beginning. With random decisions, the attack's binary search recursively locates an exponential

TABLE 3. DATASET FOR EVALUATION. THE NUMBER OF LEAVES FROM THE LEARNED DECISION TREE ARE AVERAGED OVER THE REPETITIONS.

Dataset	Samples	Features	∅ Leaves
Iris	150	4	4.6
Carseats	400	8	13.2
College	777	17	18.8
Orange Juice	1,070	11	59.0
Wine Quality	1,599	11	89.4

TABLE 4. EFFECTIVENESS OF THE TRANSFERRED CLOSENESS-TO-THE-BOUNDARY DEFENSE FOR DIFFERENT ATTACK VARIATIONS AND POSSIBLE REACTIONS AFTER DETECTING THE ATTACK.

Dataset	Original Attack		Blocking Defense		Random Resp. Defense		Adapted Attack	
	Q	p	Q	p	Q	p	Q	p
Iris	108	1.00	38	0.09	*	0.09	4,412	1.00
Carseats	871	1.00	148	0.20	*	0.20	15,156	0.46
College	2,216	1.00	244	0.10	*	0.10	8,974	0.08
Orange Juice	4,804	1.00	846	0.20	*	0.20	86,354	0.48
Wine Quality	9,615	1.00	978	0.11	*	0.11	37,406	0.11

number of boundaries erroneously, without any improvement regarding the extraction. At the same time, the final query ratio φ after submitting an honest sequence was not higher than 0.2 in all datasets, so that the tree does not mark a benign query sequence as an attack by mistake. As a result, the tree can effectively separate legitimate from malicious input sequences.

Adapted Attack. In practice, an adversary will adapt the attack strategy to the particular defense, so that we examine possible attack variations in the following. We let the attacker create *cover queries* outside the margin by selecting random values in the range of each feature. The intention is to keep the query ratio below the threshold. Table 4 shows the performance of this adapted attack where an adversary sends 40 cover queries for each tree extraction query. Despite this substantial increase in queries, the whole tree can still not be extracted. Only half of the leaves are recovered before the tree spots the attack and blocks further access.

There are two practical problems that explain the attack's failure. Without knowledge of the training data distribution, the adversary cannot know where a decision boundary could be located and thus where the margin could be. Another problem is that the attacker needs to control the ratio in almost each leaf. It is not sufficient to send just one fixed well-chosen cover query all the time since this query would only affect one leaf. These problems make the smart selection of cover queries challenging since the attacker has to perform initial queries to localize a first set of leaves. Thus, our defense can spot the attack before the adversary collects more information to formulate smarter cover queries.

Well-Informed Attack. We finally consider the situation where an attacker may even have access to parts of the training data. This makes a defense clearly challenging since the attacker can already make assumptions about a possible learning model. We let the attacker create cover queries from the leaked training data. Table 5 summarizes the fraction of extracted leaves p for varying amounts of known training data and cover queries. The defense can still block an adversary even if training data is leaked partly. If just 10% of the data is known, even 40 cover queries between each attack query do not suffice to extract the whole tree. However, if the adversary knows more data points, the cover queries spread over all leaves more equally and the attack chances increase.

Overall, our evaluation demonstrates that our transferred defense can effectively obstruct model-extraction attacks.

TABLE 5. FRACTION OF EXTRACTED LEAVES WITH AN INFORMED ATTACKER KNOWING A CERTAIN FRACTION OF THE TRAINING DATA.

Dataset	Cover Queries	Fraction training data				
		10	20	30	40	50
Iris	1x	0.17	0.21	0.21	0.21	0.22
	5x	0.64	0.85	0.89	0.92	0.94
	40x	0.76	0.91	0.94	0.97	1.00
Carseats	1x	0.28	0.29	0.28	0.29	0.30
	5x	0.39	0.60	0.69	0.82	0.89
	40x	0.50	0.87	0.97	1.00	1.00
College	1x	0.12	0.12	0.12	0.12	0.12
	5x	0.17	0.26	0.28	0.29	0.32
	40x	0.29	0.64	0.85	0.94	1.00
Orange Juice	1x	0.28	0.29	0.29	0.29	0.29
	5x	0.39	0.63	0.88	0.98	0.99
	40x	0.46	0.92	1.00	1.00	1.00
Wine Quality	1x	0.20	0.22	0.22	0.23	0.24
	5x	0.33	0.55	0.88	0.98	1.00
	40x	0.43	0.91	1.00	1.00	1.00

It is not limited to a decision tree and can be applied to models, such as an SVM, where an attacker tries to locate the decision boundary through queries. As our defense can be implemented alongside an existing classifier, online services such as BigML can easily deploy our defense in practice. To motivate further research in this direction, we make our implementation and dataset publicly available[1].

5. Discussion

Adversarial machine learning and digital watermarking are vivid research fields that have established a broad range of methods and concepts. The presented unified view demonstrates that black-box attacks and corresponding defenses in machine learning and digital watermarking should be addressed together. Section 3 discloses that multiple attacks and defenses are transferable between both fields. For instance, attacks such as the BNSA which directly computes an evasive sample without a substitute model can be exploited against learning-based classifiers. Moreover, lessons learned from the application of an attack or defense technique can serve as guidance for the other research

1. The implementation and datasets are available under https://www.tu-braunschweig.de/sec/research/data/mldw

field in this way. The watermarking field, for example, studied several randomization techniques such as adding noise to the detector's output or rendering the decision boundary more complex by fractalizing it. However, the broad conclusion was that these defense techniques mitigate, but do not prevent an attack and researchers continued with stateful defenses [1, 13]. The other way round, the adversarial learning community concluded that an attacker, for instance, can learn a local substitute model through input-output queries so that she can bypass defenses on the original model such as gradient masking due to the transferability property [45]. Our case study reveals that substitute learning models represent a similar risk to watermark detectors.

Furthermore, the identified similarities between machine learning and digital watermarking can be seen as part of a bigger problem: *Adversarial Signal Processing* [2]. More fields such as multimedia forensics also deal with an adversary and a common understanding across various fields could eventually help to combine knowledge.

Finally, we refer to various contests in both fields. The BOWS contests from digital watermarking [25, 47] or the adversarial learning contests organized at the NIPS 2017 [64] or by Madry et al. [39] have let researchers work as an attacker in a real scenario without perfect knowledge—revealing previously unknown questions and insights [e.g. 4, 67]. We thus encourage the organization of a *regular contest for adversarial machine learning*, covering different learning methods and applications.

6. Conclusion

Developing analysis methods for an adversarial environment is a challenging task: First, these methods need to provide correct results even if parts of their input are manipulated and, second, these methods should protect from known as well as future attacks. The research fields of adversarial learning and digital watermarking both have tackled these challenges and developed a remarkable set of defenses for operating in an adversarial environment.

By means of a systematization of black-box attacks and defenses, we show in this paper that both lines of research share similarities which have been overlooked by previous work and enable transferring concepts from one field to the other. In three case studies, we empirically demonstrate the benefit of such a unified view. First, we learn a Deep Neural Network as substitute model to attack a watermarking scheme. Second, the transferred concept of classifier diversity successfully prevents the Blind Newton Sensitivity Attack from removing a watermark in marked images. Last but not least, a stateful defense from digital watermarking also blocks model-extraction attacks against decision trees.

As part of our unification, we identify interesting directions of future research that enable the two communities to learn from each other and combine the "best of both worlds".

Acknowledgements

The authors acknowledge funding from Deutsche Forschungsgemeinschaft (DFG) under the project RI 2469/3-1.

References

[1] M. Barni, P. Comesaña-Alfaro, F. Pérez-González, and B. Tondi, "Are you threatening me?: Towards smart detectors in watermarking," *Proceedings of SPIE*, vol. 9028, 2014.

[2] M. Barni and F. Pérez-González, "Coping with the enemy: Advances in adversary-aware signal processing," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 8682–8686.

[3] M. Barni, F. Pérez-González, P. Comesaña, and G. Bartoli, "Putting reproducible signal processing into practice: A case study in watermarking." in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2007, pp. 1261–1264.

[4] P. Bas and A. Westfeld, "Two key estimation techniques for the broken arrows watermarking scheme," in *Proc. of ACM Workshop on Multimedia and Security*, 2009, pp. 1–8.

[5] B. Biggio, I. Corona, Z. He, P. P. K. Chan, G. Giacinto, D. S. Yeung, and F. Roli, "One-and-a-half-class multiple classifier systems for secure learning against evasion attacks at test time," in *Proc. of International Workshop on Multiple Classifier Systems (MCS)*, 2015.

[6] 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*. Springer, 2013, pp. 387–402.

[7] B. Biggio, G. Fumera, and F. Roli, "Adversarial pattern classification using multiple classifiers and randomisation," in *Structural, Syntactic, and Statistical Pattern Recognition*. Springer, 2008, pp. 500–509.

[8] B. Biggio, B. Nelson, and P. Laskov, "Support vector machines under adversarial label noise," in *Proc. of Asian Conference on Machine Learning (ACML)*, 2011, pp. 97–112.

[9] ——, "Poisoning attacks against support vector machines," in *Proc. of International Conference on Machine Learning (ICML)*, 2012.

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

[11] M. E. Choubassi and P. Moulin, "On the fundamental tradeoff between watermark detection performance and robustness against sensitivity analysis attacks," *Proceedings of SPIE*, vol. 6072, pp. 1–12, 2006.

[12] ——, "Noniterative algorithms for sensitivity analysis attacks," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 2, pp. 113–126, 2007.

[13] ——, "On reliability and security of randomized detectors against sensitivity analysis attacks," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, pp. 273–283, 2009.

[14] P. Comesaña, L. Pérez-Freire, and F. Pérez-González, "Blind newton sensitivity attack," *IEE Proceedings – Information Security*, vol. 153, no. 3, pp. 115–125, 2006.

[15] P. Comesaña and F. Pérez-González, "Breaking the BOWS watermarking system: Key guessing and sensitivity attacks," *EURASIP Journal on Information Security*, vol. 2007, no. 1, 2007.

[16] I. J. Cox and J.-P. M. G. Linnartz, "Public watermarks and resistance to tampering," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, 1997, pp. 26–29.

[17] I. J. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital watermarking and steganography*. Morgan Kaufmann Publishers, 2002.

[18] S. Craver and J. Yu, "Reverse-engineering a detector with false alarms," *Proceedings of SPIE*, vol. 6505, p. 65050C, 2007.

[19] N. N. Dalvi, P. Domingos, Mausam, S. K. Sanghai, and D. Verma, "Adversarial classification," in *Proc. of International Conference on Knowledge Discovery and Data Mining (KDD)*, 2004, pp. 99–108.

[20] H. Dang, Y. Huang, and E.-C. Chang, "Evading classifiers by morphing in the dark." in *Proc. of ACM Conference on Computer and Communications Security (CCS)*, 2017, pp. 119–133.

[21] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "RAISE: a raw images dataset for digital image forensics," in *6th ACM Multimedia Systems Conference*, 2015, pp. 219–224.

[22] R. Duda, P.E.Hart, and D.G.Stork, *Pattern classification*, 2nd ed. John Wiley & Sons, 2001.

[23] P. Fogla and W. Lee, "Evading network anomaly detection systems: Formal reasoning and practical techniques," in *Proc. of ACM Conference on Computer and Communications Security (CCS)*, 2006, pp. 59–68.

[24] P. Fogla, M. Sharif, R. Perdisci, O. Kolesnikov, and W. Lee, "Polymorphic blending attacks," in *Proc. of USENIX Security Symposium*, 2006, pp. 241–256.

[25] T. Furon and P. Bas, "Broken arrows," *EURASIP Journal on Information Security*, vol. 2008, pp. 1–13, 2008.

[26] T. Furon, B. Macq, N. Hurley, and G. Silvestre, "JANIS: Just another N-order side-informed watermarking scheme," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, vol. 3, 2002, pp. 153–156.

[27] T. Furon, I. Venturini, and P. Duhamel, "Unified approach of asymmetric watermarking schemes," *Proceedings of SPIE*, vol. 4314, pp. 269–279, 2001.

[28] T. Gloe and R. Böhme, "The Dresden Image Database for benchmarking digital image forensics," *Journal of Digital Forensic Practice*, vol. 3, no. 2–4, pp. 150–159, 2010.

[29] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial perturbations against deep neural networks for malware classification," Computing Research Repository (CoRR), Tech. Rep. abs/1606.04435, 2016.

[30] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: data mining, inference and prediction*, ser. Springer series in statistics. New York, N.Y.: Springer, 2001.

[31] L. Huang, A. D. Joseph, B. Nelson, B. I. P. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proc. of ACM Workshop on Artificial Intelligence and Security (AISEC)*, 2011, pp. 43–58.

[32] T. Kalker, J.-P. M. G. Linnartz, and M. van Dijk, "Watermark estimation through detector analysis," in *Proc. of IEEE International Conference on Image Processing (ICIP)*, 1998, pp. 425–429.

[33] A. Kapravelos, Y. Shoshitaishvili, M. Cova, C. Kruegel, and G. Vigna, "Revolver: An automated approach to the detection of evasive web-based malware," in *Proc. of USENIX Security Symposium*, Aug. 2013, pp. 637–651.

[34] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. M. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *Proc. of IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 163–168.

[35] X. Liao, K. Yuan, X. Wang, Z. Li, L. Xing, and R. A. Beyah, "Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence." in *Proc. of ACM Conference on Computer and Communications Security (CCS)*, 2016, pp. 755–766.

[36] J.-P. M. G. Linnartz and M. van Dijk, "Analysis of the sensitivity attack against electronic watermarks in images," in *Proc. of Information Hiding Conference*, vol. 1525, 1998, pp. 258–272.

[37] D. Lowd and C. Meek, "Good word attacks on statistical spam filters," in *Conference on Email and Anti-Spam*, 2005.

[38] ——, "Adversarial learning," in *Proc. of ACM SIGKDD Conference on Knowledge Discovery in Data Mining (KDD)*, 2005, pp. 641–647.

[39] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "MNIST Adversarial Examples Challenge," https://github.com/MadryLab/mnist_challenge, last visited February 2018.

[40] M. F. Mansour and A. H. Tewfik, "Improving the security of watermark public detectors," in *Proc. of International Conference on Digital Signal Processing (DSP)*, 2002, pp. 59–66.

[41] ——, "LMS-based attack on watermark public detectors," in *Proc. of International Conference on Image Processing (ICIP)*, 2002, pp. 649–652.

[42] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," arXiv:1605.07277, Tech. Rep., 2016.

[43] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. Berkay Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. of ACM Asia Conference on Computer Computer and Communications Security (ASIA CCS)*, 2017, pp. 506–519.

[44] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. of IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016.

[45] N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman, "SoK: Security and privacy in machine learning," in *Proc. of IEEE European Symposium on Security and Privacy (EuroS&P)*, 2018.

[46] R. Perdisci, G. Gu, and W. Lee, "Using an ensemble of one-class SVM classifiers to harden payload-based anomaly detection systems," in *Proc. of International Conference on Data Mining (ICDM)*, 2006, pp. 488–498.

[47] A. Piva and M. Barni, "Design and analysis of the first BOWS contest," *EURASIP Journal on Information Security*, vol. 2007, pp. 3:1–3:7, 2007.

[48] P. Russu, A. Demontis, B. Biggio, G. Fumera, and F. Roli, "Secure kernel machines against evasion attacks," in *Proc. of ACM Workshop on Artificial Intelligence and Security (AISEC)*, 2016, pp. 59–69.

[49] B. Schölkopf and A. J. Smola, *Learning with Kernels*. Cambridge, MA: MIT Press, 2002.

[50] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.

[51] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition." in *Proc. of ACM Conference on Computer and Communications Security (CCS)*, 2016, pp. 1528–1540.

[52] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. of IEEE Symposium on Security and Privacy (S&P)*, 2017.

[53] Y. Song, M. E. Locasto, A. Stavrou, and S. J. Stolfo, "On the infeasibility of modeling polymorphic shellcode," in *Proc. of ACM Conference on Computer and Communications Security (CCS)*, 2007, pp. 541–551.

[54] N. Srndic and P. Laskov, "Practical evasion of a learning-based classifier: A case study." in *Proc. of IEEE Symposium on Security and Privacy*, 2014, pp. 197–211.

[55] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," Computing Research Repository (CoRR), Tech. Rep. abs/1312.6199, 2013.

[56] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[57] B. Tondi, P. Comesaña-Alfaro, F. Pérez-González, and M. Barni, "On the effectiveness of meta-detection for countering oracle attacks in watermarking," in *Workshop on Information Forensics and Security (WIFS)*, 2015, pp. 1–6.

[58] ——, "Smart detection of line-search oracle attacks," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 3, pp. 588–603, 2017.

[59] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *Proc. of USENIX Security Symposium*, 2016, pp. 601–618.

[60] R. Venkatesan and M. H. Jakubowski, "Randomized detection for spread-spectrum watermarking: Defending against sensitivity and other attacks," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, 2005, pp. 9–12.

[61] I. Venturini, "Oracle attacks and covert channels," in *Proc. of International Workshop on Digital Watermarking*, vol. 3710, 2005, pp. 171–185.

[62] N. Šrndić and P. Laskov, "Practical evasion of a learning-based classifier: A case study," in *Proc. of IEEE Symposium on Security and Privacy*, 2014.

[63] K. Wang, J. J. Parekh, and S. J. Stolfo, "Anagram: A content anomaly detector resistant to mimicry attack," in *Proc. of International Symposium on Recent Advances in Intrusion Detection (RAID)*, 2006, pp. 226–248.

[64] Website, "Adversarial Attacks and Defences," https://www.kaggle.com/nips-2017-adversarial-learning-competition, last visited January 2018.

[65] ——, "BOWS-2 Web page," http://bows2.ec-lille.fr/, 2008, last visited August 2017.

[66] ——, "Pornographic films on BitTorrent: Flava Works gets huge damages," http://www.bbc.co.uk/news/technology-20178171, 2012, last visited August 2017.

[67] A. Westfeld, "A workbench for the BOWS contest," *EURASIP Journal on Information Security*, vol. 2007, no. 1, p. 064521, 2008.
[68] ——, "Fast determination of sensitivity in the presence of countermeasures in BOWS-2," in *International Workshop on Information Hiding*. Springer, 2009, pp. 89–101.
[69] W. Xu, Y. Qi, and D. Evans, "Automatically evading classifiers: A case study on pdf malware classifiers." in *Proc. of Network and Distributed System Security Symposium (NDSS)*, 2016.
[70] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Appendix A. Watermark Detector

This section exemplifies the watermarking process with a simple, yet commonly studied watermarking scheme. The process can be divided into two phases: embedding and detection. For the former phase, we use the *additive spread spectrum technique*. In this scheme, the watermarking parameter w consists of a pseudorandom pattern $v \in \mathbb{R}^N$ and a threshold η. The watermarked version $\tilde{x}$ of a signal x is then created by adding the watermarking vector v onto x element-wise, that is,

$$\tilde{x} = x + v \; . \tag{14}$$

In order to decide whether a signal contains the particular watermark, a *linear correlation detector* can be employed that uses the following decision function:

$$g(\tilde{x}) = \tilde{x}^\intercal v \; \gtrless \eta \; = \; \{1, -1\} \; . \tag{15}$$

The function computes a weighted sum between $\tilde{x}$ and the watermark v. If watermark and signal match, the correlation exceeds a pre-defined threshold η and a positive label is returned. Geometrically, each signal corresponds to a point in a vector space where the watermark induces a decision boundary. The result are two subspaces, one for the watermark's presence, one for its absence. The detection thus works by determining which subspace an input signal is currently in.

Appendix B. Blind Newton Sensitivity Attack

This section briefly recaps the Blind Newton Sensitivity Attack [14] (BNSA) that solves Eq. (6). As the watermark parameter w is secret, the adversary has not access to the real-valued output that $g(\tilde{x})$ internally computes before returning the binary decision. Therefore, Comesaña et al. rewrite the optimization problem from Eq. (6) into an unconstrained version:

$$\arg\min_{t \in \mathbb{R}^N} d(h(t)). \tag{16}$$

The function $h(t)$ reflects the prior constraint to find a solution in the other subspace. As a position on the boundary is sufficient, $h(t)$ maps each input to the boundary. To this end, a bisection algorithm can be used to find a scalar α such that αt lies on the decision boundary. However, $h(t)$ has to map each input vector to the boundary explicitly by running the bisection algorithm each time. Thus, a closed form to solve the problem is not applicable. Therefore, numeric iterative methods such as Newton's method or gradient descent have to be used as Figure 10 exemplifies.

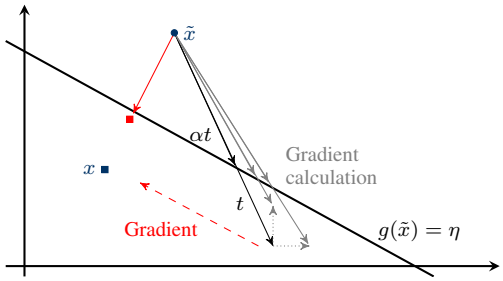

Figure 10. Blind Newton Sensitivity Attack. Queries around a boundary position reveal the function's gradient at this position to minimize the distance between the manipulated sample and the original one.

The attack starts with a random direction to locate the decision boundary. After calculating an image at the boundary, it slightly changes the vector at one position, maps the vector to the boundary again and records the distance through this change. By repeating this procedure for each direction in the vector space, the attack is able to calculate the gradient at this boundary position. In this way, the attack is able to locate a boundary position that is closer to $\tilde{x}$. The distance term d becomes smaller. In summary, the attack does not require a priori knowledge about the detector's decision function and works only with a binary output. The optimal solution is guaranteed for convex boundaries, but suitable results are also reported for non-linear watermarking schemes—with e.g. polynomial or fractalized decision boundaries—by following the boundary's envelope [14, 15].

Appendix C. Security Margin Construction

The security margin's construction works as follows: First, we choose a tree region and select the training data that fall inside this particular region. Next, we estimate the distribution of the selected training data at each dimension through a kernel-density estimation. In this way, no a priori assumptions about their distribution are required. Finally, the distribution in each dimension is used to define the margin at the boundary in this dimension. To this end, we set the margin to the feature value where the probability of occurrence is smaller than a certain threshold. In Figure 9, for example, the top right tree regions has a smaller security margin, since more training data are near the boundary. On the contrary, the most left region exhibits fewer training samples near the boundary, so that a larger margin can be defined. By defining the security margin in this statistical way, we can control the false alarm rate that a honest query falls inside the margin. We repeat the process for each tree region.