Adversarial Machine Learning Against Digital Watermarking

Erwin Quiring and Konrad Rieck

Technische Universität Braunschweig
Brunswick, Germany

***Abstract*—Machine learning and digital watermarking are independent research areas. Their methods, however, are vulnerable to similar attacks if operated in an adversarial environment. Recent research has thus started to bring both fields together by introducing a unified view for black-box attacks and defenses between learning and watermarking methods. In this paper, we extend this work and examine a novel black-box attack against digital watermarking based on concepts from adversarial learning. With a set of marked images, we let a neural network approximate the watermark detection and use this network to remove the watermark. The attack does not require knowledge of the watermarking scheme.**

***Index Terms*—Digital Watermarking, Adversarial Examples**

I. Introduction

Machine learning is nowadays a substantial part in many applications of compute science, ranging from intrusion detection systems and spam filters to medical systems and autonomous cars. The success of machine learning is rooted in its capability to automatically infer patterns and relations from a large amount of data. This inference, however, is usually not robust against attacks and thus can be disrupted or deceived by an adversary. This problem has motivated the field of *adversarial machine learning* that is concerned with attacking and defending learning methods [13, 19]. Numerous attacks and defenses have been developed, e.g. adversarial examples that mislead a neural network with unnoticeably small perturbations [5]. The research field of *digital watermarking* also operates in a security-critical environment, where an adversary seeks to identify or remove a watermark embedded into a signal such as an image or audio file. Again, various attacks and defenses have been examined, such as the prominent Blind Newton Sensitivity Attack [7].

Although both fields use different methods to achieve their goals, they have a surprisingly similar attack surface in a black-box setting. An adversary, for instance, can exploit the limited access to a classifier or watermark detector to modify a target such that it is misclassified with the smallest possible changes. Recent work has thus started to systematically study the similarities of both fields by introducing a unified notation of black-box attacks and defenses—together with three case studies to practically demonstrate a possible transfer of knowledge [20]. The authors follow the goal of combining related research fields under a common category, *Adversarial Signal Processing* [2], so that researchers can bundle existing knowledge.

In this paper, we continue with this effort and extend a case study from Quiring et al. [20]. In particular, we examine a new black-box attack mechanism from adversarial learning against digital watermarking. We assume access to a set of marked images, either from some data leakage or from an available watermark detector that allows us to label new data points. Next, we learn a *substitute model* that approximates the watermark's detection function. For this purpose, we use a deep neural network that is capable to infer even highly-nonlinear patterns. Finally, we compute a perturbation for an image on this local model that also fools the original watermark detector. This attack does not require a detailed knowledge of the watermarking scheme. Furthermore, it shifts the black-box to a white-box setting, thereby evading defenses that operate around the decision boundary [e.g. 1, 14].

We empirically examine our novel attack against the watermarking scheme Broken Arrows [11]. We train a fully connected neural network to detect the watermark. Although this neural network just approximates the watermark, it allows us to remove the watermark in 100% of the images with an average PSNR of 35.60 dB and 38.79 dB on our two test sets. In summary, we demonstrate that concepts from adversarial learning can be successfully applied in other domains of signal processing and pose a threat to current watermarking schemes.

II. Background

This section introduces the threat scenario in machine learning and digital watermarking and discusses their unified view that lays the ground for our attack in Section III.

A. Adversarial Machine Learning

Attacks against learning-based systems can be roughly grouped into three categories: *poisoning attacks*, *evasion attacks*, and *model extraction*. We focus on the latter two attacks, as they have concrete counterparts in digital watermarking.

In the *evasion attack* scenario, the attacker tries to manipulate the prediction of an already trained classifier by carefully changing the characteristics of the input data. For instance, an adversary may slightly perturb the pixels such that the image is classified to the wrong class while a person does not recognize the changes [21]. In the context of spam filtering, an adversary can try to evade detection by omitting words from spam emails indicative for unsolicited content [15]. Depending on her knowledge, the adversary operates between a *black-box* and *white-box* setting. In the former case, no information

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

about the learning method or its training data is available. The adversary needs to work with the predicted classes of the classifier solely [16, 18]. In the white-box setting, she has more knowledge about the method or data and her chances of a successful evasion increase [4]. With access to the training data, for instance, the adversary is able to learn an own model that reveals the most promising features for evasion [4].

In the *model extraction* scenario, the adversary reconstructs the underlying learning model by sending specifically crafted samples to the target system and analyzing the respective output [16, 23]. Tramèr et al. [23], for example, reconstructed various learning models from publicly available machine learning services by analyzing the returned binary or numerical function outputs. Such an attack can eventually serve as preceding step for an evasion attack.

B. Digital Watermarking

Digital watermarks are frequently used for copyright protection or to verify the authenticity of digital media, like images, music or videos. A robust 1-bit watermark is technically a random pattern that is added to the signal such that the embedding is *imperceptible* and *inseparable*, but detectable by knowing the watermark key [9]. Similar to machine learning, watermarking methods are also used in an adversarial environment and thus should not only survive unintentional changes like compression, but also targeted attacks. We focus on the following two attack classes that correspond to black-box evasion and model-extraction attacks [20].

In an *oracle attack*, an adversary exploits the access to a watermark detector that decides on the presence of a watermark in a given media sample [e.g. 7, 8]. This detector can be an online platform that verifies the authenticity of images or a media player that implements digital rights management. The attacker iteratively changes the signal by analyzing the respective binary responses until the watermark is not detected anymore. The necessary changes are minimized to preserve the signal. In the *watermark estimation* scenario, the adversary aims at recovering the watermark [e.g. 6]. In this way, she can embed or remove the watermark in a variety of new signals, thus undermining the use case of achieving copyright protection or authenticity.

C. Unified View

Digital watermarking and machine learning share a similar black-box attack surface. We shortly summarize the recently proposed unification of both fields and refer the reader to Quiring et al. [20] for a detailed discussion.

To begin with, both fields use a similar data representation. Learning methods typically operate in a so-called feature space $\mathcal{F}$ where the features can be described as a vector $x \in \mathbb{R}^N$. In the case of classification, a class label y gets assigned to each vector. Similarly, watermarking operates in a media space $\mathcal{M}$ which can be composed of pixels or audio samples. The marked and unmarked signals represent the classes. The signal can also be described as $x \in \mathbb{R}^N$. As a result, the feature and media space can correspond to each other: $\mathcal{F} \cong \mathcal{M}$.

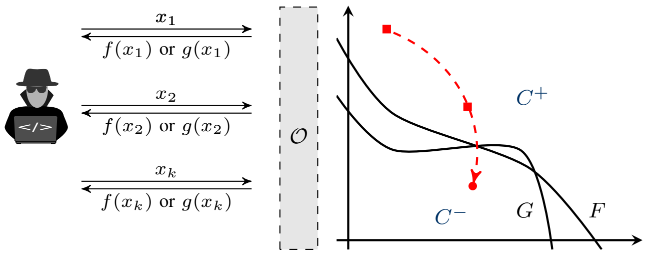

Fig. 1. An adversary tries to achieve a misclassification or to learn the detection boundary with only limited information such as the binary decisions. Internal calculations such as numerical function values are not accessible.

Moreover, a learning method such as a neural network infers the functional dependencies from the training data to separate the various data points. Internally, a prediction function $F(x)$ is learned that returns a score for a data point, e.g. its class probability. Geometrically, F separates the vector space through a decision boundary. In a black-box scenario, however, the adversary will typically have access to a function $f = \mathcal{O} \circ F$ where $\mathcal{O}$ deduces the class label from the numerical output F.

In a similar way, watermarking methods divide $\mathcal{M}$ into two separate subspaces for the marked and unmarked signals, respectively. We denote an unmarked signal by x, its marked counterpart by $\tilde{x}$, and the watermark key by w that defines the watermark pattern. Again, an adversary will typically only have access to a watermark detector's decision function g that is composed of the binary output $\mathcal{O}$ and a detection function $G(x)$ internally. In summary, we have a similar attack surface:

$$f : \mathcal{O} \circ F, \qquad g : \mathcal{O} \circ G, \tag{1}$$

$$\text{with } F : \mathcal{F} \longmapsto \mathbb{R}^M, \qquad G : \mathcal{M} \longmapsto \mathbb{R}^M, \tag{2}$$

$$\text{and } \mathcal{O} : \ \mathbb{R}^M \longmapsto \{-1, +1\} \ . \tag{3}$$

Figure 1 depicts the adversary's situation. Note that the shape of f's decision boundary needs to conform with the training data while the boundary of watermarking schemes can be created under more degree's of freedom. Yet, once the boundaries are defined, we observe the same attack surface in both fields: (a) A decision boundary, which can be different, separates the vector space into subspaces. (b) The adversary has only limited access to the classifier's/detector's response.

As a result, black-box attacks are transferable between machine learning and digital watermarking. As part of an *evasion/oracle attack*, the attacker seeks to cross the decision boundary from a—either given or iteratively queried—set of input-output pairs. In a *model extraction/watermark estimation attack*, the adversary tries to estimate the decision boundary.

In contrast to watermarking where evasive samples have been directly computed from binary outputs so far [e.g., 7], black-box attacks in adversarial learning are generally based on the *transferability-property*: an evasive sample that fools a substitute model—locally calculated by the attacker—will probably mislead the original model as well [17, 18]. To this end, the attacker adaptively collects a set of own training points, learns a local model, and conducts a white-box attack with this model. Interestingly, recent research has demonstrated that such a black-box attack can even be stronger than a white-box attack where the adversary has the original model. By using the substitute model, an adversary can overcome implemented defenses in the original model such as gradient masking which would prevent a gradient descent [19]. Note that this substitute model only needs to approximate the original model which is already enough to conduct an attack [18]. The similar attack surface motivates that such an attack is also possible against watermarking. We discuss in the next section how the concept of a substitute model—originally examined in adversarial machine learning—can be used against watermarking schemes.

III. Attack based on a Substitute Detector

In the following, we show a novel attack strategy against watermarking schemes that has been originally developed to attack image or malware classifiers. In a nutshell, the attack is based on learning a substitute model that approximates the watermark's detection function G. Then, the adversary makes use of this local model to remove or embed a watermark so that this evasive signal also transfers to the original watermark detector. Note that the learning and attacking phase require the same watermark key. The motivation behind this attack is that, first, no detailed knowledge about the watermarking scheme is necessary, since the learning process infers the pattern. Second, numerical outputs are usable instead of binary values—shifting the black-box to a white-box scenario. Third, similar to machine learning, the adversary may circumvent defenses applied on the original detector. For instance, the monitoring of a margin around the decision boundary to spot line searches [22] will not work, because the attack operates on the local model and does not need to work along the boundary.

Data Preparation. The attack requires a set of signals that are marked with the same watermark key w as well as another set of unmarked signals. Note that the adversary does not need to have the unmarked counterpart of each marked signal. Instead of feeding the signals into the learning process directly, each signal is transformed to its frequency representation where the low-frequency sub-bands are discarded. In this way, only the coefficients where the watermark is usually present are used and the learning process converges faster. We denote the high-frequency coefficients of a signal with and without watermark by $\tilde{z}$ or z, respectively.

Learning Phase. The next step consists of learning a substitute model $\hat{F}$ that approximates the detector (see Figure 2). For instance, Broken Arrows starts by calculating the high-frequency wavelet coefficients for a given input image. The security is based on a subsequent projection to a secret 256-dimensional subspace where 30 watermark patterns are defined. The closest one is used for embedding and detection. We therefore consider deep neural networks due to their capability to learn highly non-linear dependencies which makes them promising candidates to attack non-linear watermarking schemes such as Broken Arrows. In this work, we use fully connected networks. $\hat{F}(z)$ maps an input to a 2-dimensional output $y \in \mathbb{R}^2$—in our case the watermark's presence and absence. Internally, the network is composed of various layers and a final softmax layer:

$$\hat{F}(z) = \text{softmax} \circ F_m \circ F_{m-1} \circ \ldots \circ F_1 \tag{4}$$

where

$$y_i = \text{softmax}(Z_i) = \frac{\exp(Z_i)}{\exp(Z_w) + \exp(Z_a)} \quad i = w, a \tag{5}$$

The output values of F_m—the last layer before the softmax—are the so-called logits. In our case, we have two final logits Z_w (watermark presence) and Z_a (watermark absence). The softmax layer converts both values into final probability outputs y_w and y_a by ensuring $y_w + y_a = 1$ with $y_w, y_a \in [0, 1]$.

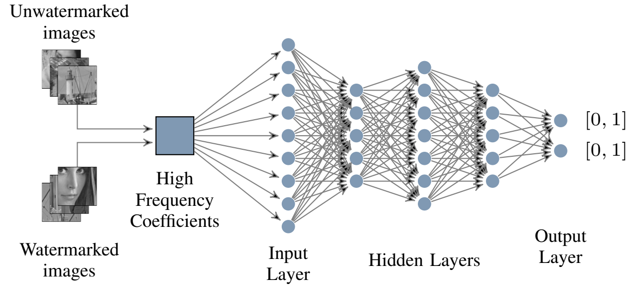

Fig. 2. A Fully Connected Neural Network acts as a substitute model for a watermark detector, trained with the frequency coefficients of marked and unmarked images.

Attack Phase. In the next step, the adversary performs a gradient descent towards the target class. We focus on removing the watermark from a marked signal in this work. The adversarial learning field has explored various strategies to find adversarial examples—outputs that have a particular misclassification with the smallest possible changes. We reuse these strategies to find an evasive sample[1].

In general, the generation of adversarial examples can be seen as an optimization problem:

$$\text{minimize } c \cdot \mathcal{L}_{\hat{F}}(\tilde{z} + \delta) + \mathcal{D} \, , \tag{6}$$

where $\mathcal{L}_{\hat{F}}$ measures the vicinity to the target class. For instance, we may set $\mathcal{L}_{\hat{F}} = y_w$ and measure the network's predicted probability of watermark presence. In addition, $\mathcal{D}$ penalizes the necessary perturbations. The parameter c represents the trade-off between both optimization terms.

Previous work has shown that the choice of $\mathcal{L}_{\hat{F}}$ has a significant impact on attack performance. We slightly adapt the state-of-the-art formulation from Carlini and Wagner that uses the logits instead of the softmax outputs [5]:

$$\mathcal{L}_{\hat{F}}(\tilde{z}') = Z_w(\tilde{z}') - Z_a(\tilde{z}') \, . \tag{7}$$

During the gradient descent, this formulation keeps the relative weight between both optimization terms in Eq. (6) more equal

[1] Note a subtle difference. In adversarial learning, the attack aims at fooling the neural network, for instance, by finding weak spots. In our case, the attacks uses the neural network as substitute to fool the watermark detector. As a result, weak spots in the substitute model do not result in a successful attack against the watermark detector.

than the probability values from the softmax layer do [5]. In addition, we try two formulations to measure the changes:

$$\mathcal{D}_1 = \|\delta\|_2 \ , \tag{8}$$

$$\mathcal{D}_2 = \|\tilde{z} + \delta\|_2 \ . \tag{9}$$

$\mathcal{D}_1$ is commonly used in adversarial learning and rewards positions that are closer to the starting position. $\mathcal{D}_2$ is a novel measure that exploits the frequency representation and penalizes the increase of high-frequency coefficients.

We perform a gradient descent from an input signal $\tilde{z}$ until the substitute model predicts the watermark's absence with high confidence. We found empirically that we often do not get stuck into local minima by taking the sign of the gradient, at the expense of more changes due to a non-optimal path. Note that a high confidence is necessary, since the substitute model approximates the detector. Thus, the neural network's decision and the detector's decision may not match in a certain transition interval. We finally exploit the oracle access to check if the watermark is present and continue with the gradient descent if necessary.

IV. Evaluation

We examine the previously described attack in two steps. First, we test that an adversary can learn a substitute model to approximate the watermark detector. Second, we demonstrate that this model can be used to remove the watermark from images.

Experimental Setup. We use Broken Arrows from the second "Break Our Watermarking System" (BOWS) competition [11], as it represents an advanced, publicly available watermarking scheme. The training and test set are drawn from different sources to lower the probability of similar image content. We work with the Raise Image Database [10] as training and validation set. Images from the Dresden Image Database [12] and another set of manually selected images serve as test set. The latter set was drawn from standard images like Lena or Barbara. These images are generally highly-textured and ensure that we use an image set with meaningful content despite the cropping process. All images were converted to grayscale, cropped to a common size of 128×128 with varying offsets, and marked with the same watermark key. We excluded images with a too weak watermark embedding[2]. The embedding PSNR is adjusted to 43 dB. Our training set finally consists of 13,050 marked and 13,050 *other* unmarked images, and the validation set of 5,593 images of each class. The test set from the Dresden database has 19,489 images of each class, and the manually selected set has 444 images of each class.

For the attack, we select 225 watermarked images from each test set. We repeat the attack process for both distance measures $\mathcal{D}_1$ and $\mathcal{D}_2$. We further vary the trade-off parameter c and the gradient descent's step size, and slightly change the starting positions to mitigate local minima. For each image, we report the respective solution with the smallest changes. We measure the number of successful watermark removals and the average Peak Signal to Noise Ratio (PSNR) between the original marked image and its adversarial counterpart. As a simple baseline, we use a median denoising filter where we increase the window size until the watermark is not detected anymore.

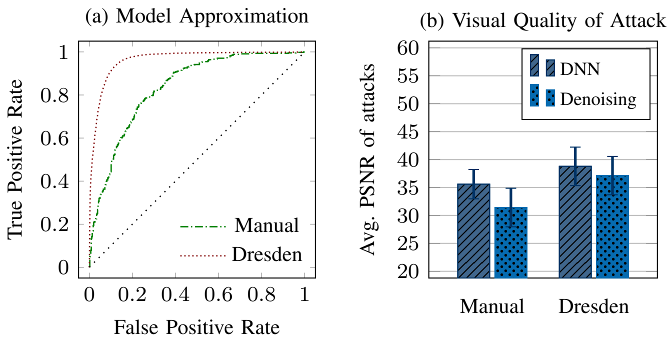

Fig. 3. Attack Evaluation. (a) the ROC curve to measure the model's capability to separate marked from unmarked images. (b) the PSNR as measure for the achieved quality after removing the watermark.

Substitute Model. Our final neural network architecture consists of 40, 60, 40 neurons in the consecutive hidden layers. Figure 3(a) depicts the ROC curve for both test sets with y_w as discrimination threshold. The y-axis shows the number of correctly detected watermarks. With a simple cutoff value at $y_w = 0.5$, the network's accuracy is 86.82% for the Dresden Image Database images, and 75.45% for the manually selected images. The difference in accuracy is explainable by the fact that the cropped images from the Dresden Image Database have smoother content than the manually selected set. Thus, the watermark is embedded slightly stronger and its inference is less disturbed by image content. In addition, we verified that the substitute model does not learn to differentiate that more noise means watermark presence. We marked each image from the previously unmarked set with another watermark key with no impact on the accuracy. Overall, the results highlight that the network only learns an approximation of the watermark pattern, which is enough for a subsequent attack.

Attack Evaluation. The adversary is able to make the watermark undetectable in 100% of the images with the DNN-based and baseline attack. Figure 3(b) presents the corresponding PSNR values. Our DNN-based attack achieves an average PSNR of 35.60 dB with a standard deviation of 2.62 dB on the manually selected set, and 38.79 dB (standard deviation: 3.44 dB) on the Dresden Image Database images. The magnitude of these PSNR values is comparable to reported results during the 2nd BOWS contest [24]. Figure 4 gives an additional intuition about the resulting image quality. The evasive images preserve image details like edges. On the contrary, the denoising process removes substantial details.

Comparing the distances from Eq. (8) and Eq. (9), we found that $\mathcal{D}_1$ yields higher PSNR values in 80% of the manual

[2] We exclude images with a cos value less than 0.45 which Broken Arrows internally computes.

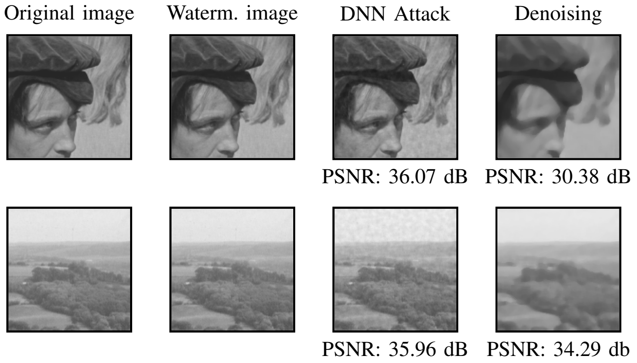

Fig. 4. Comparison of the achieved quality for two adversarial samples from the manually selected test set. The original watermarked image serves as reference for the PSNR calculation. The DNN-based attack removes the watermark while retaining image details, e.g. the face in the upper row.

test images and 53.78% of the Dresden images. However, $\mathcal{D}_2$ retains a considerable performance in particular cases where $\mathcal{D}_1$ would lead to substantially smaller PSNR values. Thus, both distances complement each other.

Discussion. Our attack demonstrates that an adversary with no background information and a set of marked images is able to apply concepts from adversarial learning to attack a watermark detector successfully. While specialized attacks against Broken Arrows [e.g. 3] yield higher PSNR values, we present a novel, generic solution that attains a considerable attack performance without any knowledge of the watermarking scheme. We credit our attack performance to the learned sensitivity of each frequency component with respect to the watermark. Future work may improve the approximation quality or reduce the number of training images by using data augmentation techniques or adaptive relearning strategies.

V. Conclusion

This paper strives to narrow the gap between adversarial machine learning and digital watermarking by extending a previous case study from Quiring et al. [20]. We enhance the experimental setup by using larger and less images for training, an additional test set, less neurons, and a denoising baseline. The attack only requires a set of marked images without their unmarked counterpart. We revise the optimization problem to generate evasive samples so that we accelerate their generation and improve their quality.

Overall, we demonstrate that the attack strategy to learn and exploit a substitute model from adversarial learning also threatens digital watermarks. We are able to approximate the watermarking scheme Broken Arrows with a fully connected neural network. While this is only an approximation, we can yet apply concepts to find evasive examples in machine learning to remove the watermark from images. To motivate further research in this direction, we make our implementation and dataset publicly available[3].

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

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] 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.

[4] 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.

[5] 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.

[6] M. E. Choubassi and P. Moulin, “Noniterative algorithms for sensitivity analysis attacks,” *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 2, pp. 113–126, 2007.

[7] 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.

[8] 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.

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

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

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

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

[17] 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.

[18] 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.

[19] 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)*, Apr. 2018.

[20] E. Quiring, D. Arp, and K. Rieck, “Forgotten siblings: Unifying attacks on machine learning and digital watermarking,” in *Proc. of IEEE European Symposium on Security and Privacy (EuroS&P)*, Apr. 2018.

[21] 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.

[22] 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.

[23] 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.

[24] Website, “BOWS-2 Web page,” http://bows2.ec-lille.fr/, 2008, last visited August 2017.