On the Security of the One-and-a-Half-Class Classifier for SPAM Feature-Based Image Forensics

Benedikt Lorch, Franziska Schirrmacher, Anatol Maier, Christian Riess, *Senior Member, IEEE*

***Abstract*—Combining multiple classifiers is a promising approach to hardening forensic detectors against adversarial evasion attacks. The key idea is that an attacker must fool all individual classifiers to evade detection. The 1.5C classifier is one of these multiple-classifier detectors that is attack-agnostic, and thus even increases the difficulty for an omniscient attacker.**

Recent work evaluated the 1.5C classifier with SPAM features for image manipulation detection. Despite showing promising results, their security analysis leaves several aspects unresolved. Surprisingly, the results reveal that fooling only one component is often sufficient to evade detection. Additionally, the authors evaluate classifier robustness with only a black-box attack because, currently, there is no white-box attack against SPAM feature-based classifiers.

This paper addresses these shortcomings and complements the previous security analysis. First, we develop a novel white-box attack against SPAM feature-based detectors. The proposed attack produces adversarial images with lower distortion than the previous attack. Second, by analyzing the 1.5C classifier's acceptance region, we identify three pitfalls that explain why the current 1.5C classifier is less robust than a binary classifier in some settings. Third, we illustrate how to mitigate these pitfalls with a simple axis-aligned split classifier. Our experimental evaluation demonstrates the increased robustness of the proposed detector for SPAM feature-based image manipulation detection.

***Index Terms*—image forensics, counter-forensics, adversarial examples, one-and-a-half-class classifier**

I. Introduction

Machine learning classifiers are being increasingly deployed in security-related applications such as biometric identity recognition, forensic image authentication, intrusion detection and content-control filtering. In these scenarios, machine learning classifiers are exposed to malicious attackers whose goal is to evade the detection. Therefore, it is extremely important to harden classifiers against adversarial evasion attacks.

To date, no effective defense against evasion attacks has been found. However, there are several approaches to increasing classifier robustness against adversarial examples. One of these approaches combines multiple classifiers that an attacker must overcome. Even though this approach cannot fully prevent evasion attacks, it can at least increase the difficulty for an attacker to deceive the classifier. By forcing attackers to introduce larger amounts of distortion, the defender increases their chances of spotting adversarial input.

One of these multiple-classifier systems is the one-and-a-half-class (1.5C) classifier [1]. The 1.5C classifier consists of a binary classifier and two one-class classifiers, each of which tightly encloses one of the two classes. The outputs of these three classifiers are fused by a final one-class classifier, which makes the final decision. All components of the 1.5C classifier are instantiated with support vector machines. Compared to other defenses, such as adversarial training, the 1.5C classifier does not make assumptions about the nature of the adversarial perturbations and is therefore agnostic to different attacks. Moreover, in contrast to obfuscation-based defenses, the 1.5C classifier also increases the attack difficulty for an omniscient attacker who is aware of any protection techniques employed by the defender. This makes the 1.5C classifier a promising approach for safety-critical tasks, including malware detection [1], watermarking detection [2], and malicious scripting code detection [3].

In recent work, the 1.5C classifier was adopted to safeguard an image manipulation detector against adversarial examples [4]. In particular, the authors train the 1.5C classifier with subtractive pixel adjacency matrix (SPAM) features, which are well-known for their numerous applications in forensics and steganalysis. Protecting image forensics detectors against adversarial examples is particularly challenging because forensic image analysis often relies on weak traces. To date, the 1.5C classifier is one of few attack-agnostic defense mechanisms that has been evaluated in image forensics.

While the security analysis shows promising initial results [4], the authors leave several aspects open-ended. First, the authors only evaluate classifier robustness with a black-box attack because to the best of our knowledge, currently, there is no white-box attack against SPAM feature-based classifiers. Nevertheless, classifier robustness should also be studied in a white-box setting. Second, the results in [4] reveal that fooling only a single component of the 1.5C classifier is sufficient for evading detection, although the 1.5C classifier's key idea is that the attack must fool all components. Third, the authors only consider a single decision threshold. However, a conservative analyst may choose a tighter threshold to further increase classifier robustness against adversarial examples.

In this work, we address these shortcomings and complement the existing security analysis [4] of the 1.5C classifier for image manipulation detection.

As our first contribution, we develop the first white-box attack against SPAM feature-based detectors. This attack enables studying classifier robustness in the worst-case scenario of an omniscient attacker. Although hand-crafted features are increasingly replaced by learned features, we use SPAM features for comparison against prior work [4] and because SPAM features still offer greater robustness against evasion attacks, as we demonstrate in this paper.

We gratefully acknowledge support by the German Research Foundation (grants 393541319/GRK2475/1-2019 and 146371743/TRR 89), and the German Federal Ministry of Education and Research (grant 13N15319).
The authors are with the IT Security Infrastructures Lab, Friedrich-Alexander-Univerität Erlangen-Nürnberg, Germany. Email: benedikt.lorch@fau.de.

As our second contribution, we analyze the 1.5C classifier with several safety margins imposed by the analyst. Interestingly, we find the 1.5C classifier to be less robust than a binary classifier in some settings. By investigating this unexpected result in-depth, we identify three pitfalls in the implementation of the 1.5C classifier. These pitfalls concern the missing normalization, shape, and size of the final classifier's acceptance region. As our third contribution, we show that replacing the 1.5C classifier's final one-class classifier by axis-aligned splits prevents these pitfalls and achieves higher robustness against adversarial attacks than the current 1.5C classifier.

This paper is organized as follows: Section II reviews related work on counter-forensic attacks and defenses. Section III outlines adversarial attacks against SPAM feature-based classifiers. The 1.5C classifier and its shortcomings are presented in Sec. IV, followed by the experimental evaluation in Sec. V. Section VI concludes this paper.

II. Related Work

Counter-forensics subsumes techniques that hinder or mislead a forensic analysis. Most counter-forensic techniques aim at concealing traces of a manipulation such that a fake image passes forensic authentication. In addition to these so-called evasion attacks, attacks with other goals have also been developed, *e.g.*, implanting traces to mislead the analyst [5] or poisoning the training data to insert a backdoor into the detector [6]. However, in this work we only consider evasion attacks. The following section presents a selection of evasion attacks against forensic detectors as well as defenses for protection against these types of attacks. A more fine-grained categorization can be found in a recent survey [7].

A. Counter-Forensics

Counter-forensic attacks can be categorized by the amount of knowledge that they require on the forensic detector. Since forensic detectors often rely on relatively weak signals, traces of manipulations can sometimes be erased by simple post-processing. These laundering attacks require little to no prior knowledge about the detector. Typical laundering operations include JPEG compression [8], median filtering [9], or re-broadcasting [10] and pose a severe challenge to forensic detectors. Although laundering allows an attacker to erase many traces, an analyst can potentially detect these destructive laundering operations and refrain from any conclusion.

While laundering attacks alter the signal's properties, universal attacks aim at restoring the original signal's statistical properties that have been perturbed by a manipulation. Therefore, universal attacks are effective against any detector that builds on these statistical properties. For example, universal attacks have been developed to hide histogram modifications [11] and to conceal multiple JPEG compressions based on DCT histograms [12] and first significant digits [13], [14].

Given white-box access to the detector, attackers can craft adversarial examples with even less distortion by targeting weaknesses of the particular detector. More specifically, targeted attacks conceal traces that a particular detector searches for. Therefore, early works specifically target detectors for resampling by suppressing periodic artifacts during interpolation [15], they conceal contrast enhancement by smoothing out peaks and gaps in the intensity histogram [16], or they deceive compression detectors by restoring DCT histogram distributions and removing blocking artifacts [17]. Since these attacks were designed against a specific detector, we categorize them as targeted attacks analogous to Böhme and Kirchner [5]. However, it should be noted that these attacks can be effective against any detector with a feature set that captures the targeted signal properties. Other attacks also hide traces of median filtering [18] or copy-move forgeries [19].

With the advent of deep learning in multimedia forensics, attacks have been developed against convolutional neural networks (CNNs) for camera model identification [20], global image manipulation detection [21], and rebroadcasting detection [10]. Recent work also explored generative adversarial networks (GANs) as a tool for erasing traces from median filtering [22] and JPEG compression [23], as well as to deliberately falsify camera model traces [24], [25].

A fundamental requirement of all these attacks is that they produce quantized images with valid pixel values. This is important because images are typically stored and transmitted with pixel values in the unsigned byte range. When intensity values need to be rounded to this range, naive rounding often interferes with adversarial perturbation. Even in the machine learning literature, only a few adversarial attacks produce quantized adversarial examples, *e.g.*, the decoupling direction and norm (DDN) [26] and the boundary projection [27] attacks. As a remedy, Bonnet *et al.* proposed a post-hoc quantization step that preserves the evasiveness while keeping the distortion low [28]. The authors also showed that integrating the quantization step into each attack iteration allows the attack to compensate for quantization errors, thereby producing adversarial images with even lower distortion.

For feature-based detectors, which are popular in forensics literature, adversarial attacks must be able to map feature modifications back into the pixel domain. When the relationship between the pixel and feature domains is invertible, *e.g.*, when the detector works in the DCT domain [12], the attack can operate in the feature domain and easily convert adversarial features into an adversarial image. For gradient-based attacks, the relationship between the pixel and feature domains only needs to be differentiable, such that the gradient signal can be backpropagated into the pixel domain.

However, if the relationship between pixel and feature domains is non-invertible, controlling the pixel-domain distortion becomes more challenging. Thus, Marra *et al.* proposed a two-stage approach, which first seeks for an adversarial example in the feature domain and then searches for pixels that produce these adversarial features [29]. As an alternative solution, Chen *et al.* approximated a pixel-domain gradient by probing the detector output after modifying individual pixels [30]. This black-box attack is effective even for non-invertible feature mappings, but re-computing the features and detector output for each pixel modification is expensive. Building on the work by Chen *et al.* [30], Tondi showed that this attack can also be used to craft quantized adversarial images against CNN detectors [31]. Despite being effective against black-box

classifiers, the attack cannot benefit from gradient information in a white-box setting. Instead, the attack requires a costly gradient approximation.

Recently, Athalye *et al*. showed that replacing non-differentiable transformations with differentiable approximations provides sufficient information for guiding gradient-based attacks [32]. This approach is called *backward pass differentiable approximation* (BPDA) and was originally proposed to overcome gradient shattering as a defense mechanism. In this work, we use this type of BPDA to obtain gradients against SPAM feature-based detectors. Compared to [33], we only use the approximation during the backward pass, whereas the forward pass yields the original SPAM features.

B. Anti-Counter-Forensics

Upon perturbing specific signal properties to deceive a forensic detector, adversarial attacks involuntarily alter other statistics as well. An analyst can use these alterations to identify traces of counter-forensics. For example, Kirchner and Chakraborty showed that restoring histograms of the first significant digits impacts the second significant digits [34]. Other related works also revealed telltale signs left by counter-forensic attacks against JPEG compression [35], median filtering [36] and chromatic aberration [37] detectors. An analyst can either search for these counter-forensic traces with an additional detector or harden the original detector by augmenting the original with these additional features. For example, related work included second-order statistics into the feature set of a contrast enhancement detector [38]. As a complementary technique, adversarial examples can be included in the training set to increase the detector's robustness in the presence of an adversary [10], [39].

In addition to improving the training data, another line of work focuses on more secure detector architectures. One direction is to select a random subset of the feature space, and rely on the attacker's lack of knowledge about the reduced feature space [40], [41]. Another promising direction is to combine multiple detectors. For example, Fontani *et al*. fused decisions from several forensic algorithms [42]. Similarly, Biggio *et al*. combined a binary classifier with two one-class classifiers [1]. The resulting one-and-a-half class (1.5C) classifier attains the accuracy of a binary classifier in the absence of an adversary while inheriting the rejection abilities from the one-class classifiers. In contrast to binary classifiers that merely partition the feature space, this architecture enables input rejection from regions with little training data support. The 1.5C classifier has been studied extensively in [4] for global image manipulation detection in an adversarial environment. The 1.5C classifier was also used to protect a watermarking detector against implausible signals [2]. While the 1.5C classifier architecture shows improved robustness against adversarial attacks, in this paper we demonstrate three potential pitfalls that can detrimentally affect its security. As a remedy, we propose a hardened variant of the 1.5C classifier.

III. Adversarial Attacks Against SPAM Feature-Based Detectors

This section describes two adversarial attacks against detectors that are based on SPAM features: the black-box attack by Chen *et al*. [30] and the proposed white-box attack, which combines the CNN-based feature extraction [33] with a BPDA gradient approximation [32] and the DDN attack [26]. We begin by summarizing the SPAM feature extraction.

A. SPAM Features

SPAM features are designed to capture local pixel dependencies. A large family of configurations was developed and evaluated originally for steganalysis [43] and later for image forensics [44]. This work focuses on one particular configuration of SPAM features, but can also be applied to similar models.

1) Feature Extraction: The SPAM feature extraction consists of the following steps: First, the scene content is suppressed by applying a high-pass filter with the third-order linear kernel, $\boldsymbol{k} = [1, -3, 3, -1]$, in the horizontal and vertical direction, resulting in two residuals, $\boldsymbol{r}^{(h)}$ and $\boldsymbol{r}^{(v)}$. Second, the residuals are quantized and truncated to reduce their complexity as follows:

$$\hat{r}_s^{(\cdot)} = \text{trunc}_{\tilde{T}}(\text{round}(r_s^{(\cdot)}/\tilde{q})) \ , \tag{1}$$

where s is the spatial position. We use the quantization step $\tilde{q} = 4.5$ and the truncation value $\tilde{T} = 1$, resulting in $L = 3$ quantization levels. In the third step, each quantized and truncated residual pixel is compared to its $N = 4$ neighbors to capture local patterns. For both $\hat{\boldsymbol{r}}^{(h)}$ and $\hat{\boldsymbol{r}}^{(v)}$, the resulting co-occurrence matrices are calculated in the horizontal and vertical directions. Each co-occurrence matrix is normalized so that its entries sum up to 1. Fourth, the four resulting co-occurrence matrices are then reduced by symmetry, and the horizontal-horizontal pair is added to the vertical-vertical pair. Similarly, the horizontal-vertical co-occurrence is added to the vertical-horizontal co-occurrence matrix. Finally, the two resulting matrices are flattened and concatenated to form a 50-dimensional statistical feature descriptor for each image.

2) CNN-based SPAM Feature Extraction: The SPAM feature extraction can also be implemented with a constrained CNN. We briefly summarize the CNN-based feature extraction, as shown in [33]. The image residuals are obtained through a convolutional layer with fixed weights. To calculate co-occurrences, the image residuals are stacked along the third dimension with three horizontally (or vertically) shifted versions. The quantization and truncation steps are implemented by matching the residual values to the $T = L^N$ pre-computed template vectors, one for each possible quantization outcome. The matching score, $m_{t,s}$, is the negative distance to the t-th template vector and is calculated using a convolutional layer. The residual values are then assigned to the best-matching template vector using a hardmax transformation as follows:

$$p_{t,s}^{\text{hard}} = \begin{cases} 1 & \text{argmax}_{i=1,\dots,T}\, m_{i,s} = t \\ 0 & \text{otherwise} \end{cases} \ . \tag{2}$$

This assignment $p_{t,s}^{\text{hard}}$ is stored as a one-hot encoded vector, whose length equals the number of co-occurrence entries. Symmetric entries in the assignment vector are summed with a fully-connected layer where the binary weights indicate which entries can be combined. The resulting vector is eventually normalized to unit sum by average pooling.

This architecture composes one of four blocks. The remaining three blocks vary the filter weight directions and the shift directions. The resulting four descriptors are combined as described above to form a 50-dimensional statistical feature descriptor for each image. This CNN-based implementation yields the same results as the original feature extraction.

B. Black-Box Evasion Attack Against SPAM-Based Detectors

The goal of an adversary is to minimally modify a fake image so that it passes forensic authentication. Most attacks compute the gradient of the forensic detector's output with respect to the image pixels and update the pixel values via gradient descent. In the case of SPAM features, however, the relationship between the pixels and features as detector input is not injective. Additionally, gradient descent updates can result in non-integer pixel values, which are cut off by most image formats. In the remainder of this section, we first summarize the gradient-based attack against a SPAM feature-based SVM by Chen *et al*. [30]. Second, we describe a gradient approximation that can be combined with state-of-the-art attacks, such as the DDN attack.

Chen *et al*. used a finite differences approximation of the gradient direction to attack an SVM that was trained on SPAM features [30]. Therefore, we refer to this attack as a finite differences (FD) attack. To estimate the gradient direction, the attack modifies a single pixel by an increment of 1 and evaluates the resulting difference in the detector's decision function. This procedure is repeated for each pixel location to obtain a gradient matrix for all of the pixels. Instead of modifying all of the pixels, the authors found that modifying a fraction of the pixels per attack iteration yields adversarial images with less distortion. Therefore, the pixels are ordered by their gradient magnitude. After obtaining the gradient direction, the attack searches for the minimum number of pixels to update via a line search until the objective function is fulfilled. If the attack has not succeeded after updating $K = 20\%$ of all the pixels, a fraction of the pixels is updated and the attack proceeds to the next iteration.

Note that this finite differences approximation leads to a coarse estimate of the gradient. Estimating the gradient direction also requires re-computing the SPAM feature descriptor and detector output for each pixel location. This computational effort can be reduced by caching the residual image and the co-occurrence matrices so that only a few entries of the co-occurrence matrices need to be updated. Nevertheless, iterating over all pixel locations is still expensive, especially for high-resolution images.

C. Proposed White-Box Attack With BPDA

The attack that is proposed in this work combines the CNN-based feature extraction [33] described above with a BPDA [32] and the DDN attack [26]. Therefore, we refer to this attack as DDN-BPDA. During the forward pass, the CNN-based feature extraction yields the original SPAM features. During the backward pass, we replace the non-differentiable hard assignment of each pixel to the closest template vector with a soft assignment implemented with a softmax transformation. As a result, pixels are partially assigned to several cells of the co-occurrence matrix. Cozzolino *et al*. explored this variant with the goal of improving the original SPAM features [33]. In contrast, we use this soft assignment to obtain a gradient approximation. The balance between a soft and hard assignment can be controlled by scaling the softmax steepness with a hyper-parameter α,

$$p_{t,s}^{\text{soft}} = \frac{\exp(\alpha\, m_{t,s})}{\sum_{i=1}^{T} \exp(\alpha\, m_{i,s})} \quad . \tag{3}$$

A high α corresponds to a hard assignment as in the original formulation, but also leads to vanishing gradients. Conversely, a small α corresponds to a soft assignment with more useful gradients but a worse approximation of the original features. This hyper-parameter α is subject to an ablation study in Sec. V-E3.

After obtaining the gradient approximation, we search for adversarial examples with the *decoupling direction and norm (DDN)* attack [26]. We use the DDN attack because it produces quantized adversarial images and its computational cost is low compared to that of more sophisticated attacks. Note that the proposed gradient approximation can readily be used with off-the-shelf attacks available in modern libraries [45] and benefits from the GPU acceleration of deep learning frameworks.

DDN is a gradient-based iterative attack that aims to find low-distortion adversarial examples by optimizing a perturbation budget ϵ [26]. In our scenario, the attack maximizes the detector output by taking a step in the direction of its normalized gradient. The step size starts at 1 and is reduced to 0.01 by cosine annealing with the number of iterations. The resulting adversarial noise is projected onto an ϵ-sphere around the original example. The radius ϵ is defined by the current perturbation budget. If the previous attack iteration leads to an adversarial example, then the perturbation budget is decreased to $(1-\gamma)\epsilon$, where γ is a hyper-parameter. If the previous step is unsuccessful, the perturbation budget is increased to $(1+\gamma)\epsilon$. Each iteration is concluded by quantizing the adversarial noise with naive rounding. We also experimented with Lagrangian quantization at the end of each iteration [28]. However, given enough attack iterations, we did not observe a considerable benefit compared to naive rounding.

IV. One-and-a-Half-Class Classifier

Binary classifiers achieve high classification performance by partitioning the input space. Nevertheless, they typically make decisions with unreasonably high confidence for test samples that are far away from the training data. Though, one-class classifiers are designed to detect anomalous examples

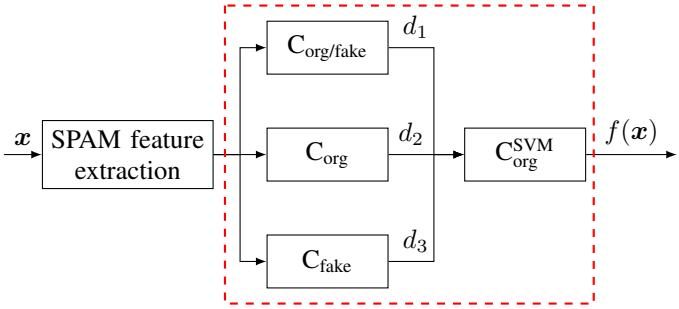

Fig. 1. The original 1.5C classifier consists of a binary classifier and two one-class classifiers trained on either class in the first level. A final one-class classifier fuses the predictions of the first-level classifiers. The schematic description is adopted from [4].

outside the training data's region of support, they underperform binary classifiers in terms of classification accuracy. To obtain the best of these two complementary concepts, Biggio *et al*. proposed the one-and-a-half-class (1.5C) classifier as a multiple-classifier architecture [1]. It offers a better trade-off between classification accuracy in the absence of attacks and robustness against evasion attacks. In the following section, we first describe the 1.5C classifier and outline three potential pitfalls. Then, we explain how to overcome the shortcomings of the current 1.5C classifier.

A. Multiple-Classifier Architecture

The 1.5C classifier architecture is depicted in Fig. 1. The 1.5C classifier consists of a binary classifier and two one-class classifiers in the first level and a single one-class classifier in the second level. The binary classifier $\mathrm{C}_{\mathrm{org/fake}}$ is trained with inputs from both classes, while the two one-class classifiers $\mathrm{C}_{\mathrm{org}}$ and $\mathrm{C}_{\mathrm{fake}}$ are each trained on samples from either class. The final prediction is made by the second-level one-class classifier $\mathrm{C}_{\mathrm{org}}^{\mathrm{SVM}}$, which is trained with original images only. $\mathrm{C}_{\mathrm{org}}^{\mathrm{SVM}}$ operates on the decision functions d_1, d_2, d_3 of the three first-level classifiers and does not observe the input features. In both [1] and [4], this architectural design is instantiated with SVMs. For simplicity, we add the superscript SVM only for the final one-class classifier, which we replace with another classifier later.

Figure 2 illustrates the role of each individual component of the 1.5C classifier on the 2-D synthetic example from [1]. Here, the 1.5C classifier is specifically designed to prevent false assignments to the blue class. The black contour marks the decision boundary. The binary classifier $\mathrm{C}_{\mathrm{org/fake}}$ in the left panel perfectly separates the training data, but assigns low-density regions to the blue class. The two one-class classifiers $\mathrm{C}_{\mathrm{org}}$ and $\mathrm{C}_{\mathrm{fake}}$ draw a tight acceptance region around the respective class, but both classifiers also misclassify a few training examples. However, the right panel of Fig. 2 shows that $\mathrm{C}_{\mathrm{org}}^{\mathrm{SVM}}$ combines the decision scores of the three classifiers and tightly encloses all the blue training examples.

B. Hyper-Parameter Optimization

All SVM classifiers use an RBF kernel, and therefore, come with a regularization and a kernel width hyper-parameter. These hyper-parameters are selected via cross-validation. For the one-class SVMs, Barni *et al*. [4] selected the set of hyper-parameters that minimizes a weighted sum of false positive rate P_{fp} and missed detection rate P_{md} as follows:

$$P_e = \lambda \cdot P_{fa} + \varphi \cdot P_{md} \ . \tag{4}$$

We run cross-validation on our dataset but use the same λ, φ as in [4]. Specifically, $\mathrm{C}_{\mathrm{org}}$ uses $\lambda = 0.8$ and $\varphi = 0.2$ such that false alarms are discouraged in favor of missed detections. Conversely, $\mathrm{C}_{\mathrm{fake}}$ uses $\lambda = 0.2$ and $\varphi = 0.8$. The $\mathrm{C}_{\mathrm{org}}^{\mathrm{SVM}}$ uses $\lambda = 0.9$ and $\varphi = 0.1$. Minimizing this weighted error rate should encourage a tightly enclosed acceptance region around the original images; however, the acceptance region can still become unnecessarily large, as we show in the next section.

C. Pitfalls of the 1.5C Classifier

The 1.5C classifier architecture claims improved robustness against evasion attacks at negligible performance degradation in the absence of attacks. Nevertheless, we identify three pitfalls that, if neglected, substantially impair the security and classification performance.

1) Missing Normalization: In both the original paper [1] and follow-up work [4], $\mathrm{C}_{\mathrm{org}}^{\mathrm{SVM}}$ directly operates on the decision scores of the first-level classifiers. The decision scores of the three first-level classifiers can take on considerably different ranges (see Fig. 7 in [4]). The final one-class SVM uses an RBF kernel with a shared length scale, which requires a distance evaluation. Therefore, if the first-level classifiers' predictions are scaled differently, the distances calculated by the RBF kernel can be misleading. As a result, optimization of the shared length scale focuses on the input dimension with the largest range, neglecting outputs of the two other classifiers.

2) Size of the Acceptance Region: As proposed in [4], the hyper-parameters of all components, including the RBF kernel length scale, are obtained by minimizing the weighted error probability from Eq. 4. The optimization goal of the final one-class classifier is therefore to include as many original images as possible but without any fake images. As a result, $\mathrm{C}_{\mathrm{org}}^{\mathrm{SVM}}$ can extend its acceptance region into rejection regions of the first-level classifiers. Therefore, an attacker does not need to fool all three first-level classifiers to deceive the final classifier. This reduces the amount of adversarial noise for a successful attack.

3) Shape of the Acceptance Region: With the ellipsoidal shape of the acceptance region, $\mathrm{C}_{\mathrm{org}}^{\mathrm{SVM}}$ yields the highest confidence in the center of the acceptance region. However, we argue that the final classifier should be confident about an original image when the three first-level classifiers are confident. Hence, the final classifier provides misleading confidence values.

This can have unintended consequences. For example, $\mathrm{C}_{\mathrm{org}}^{\mathrm{SVM}}$ rejects test points with unusually high confidence by the first-level classifiers. Rejecting samples outside the support of the training data can be desirable but can also lead to unintuitive holes in the feature space. This can be seen in the right panel of Fig. 2. Furthermore, these high-confidence regions of the

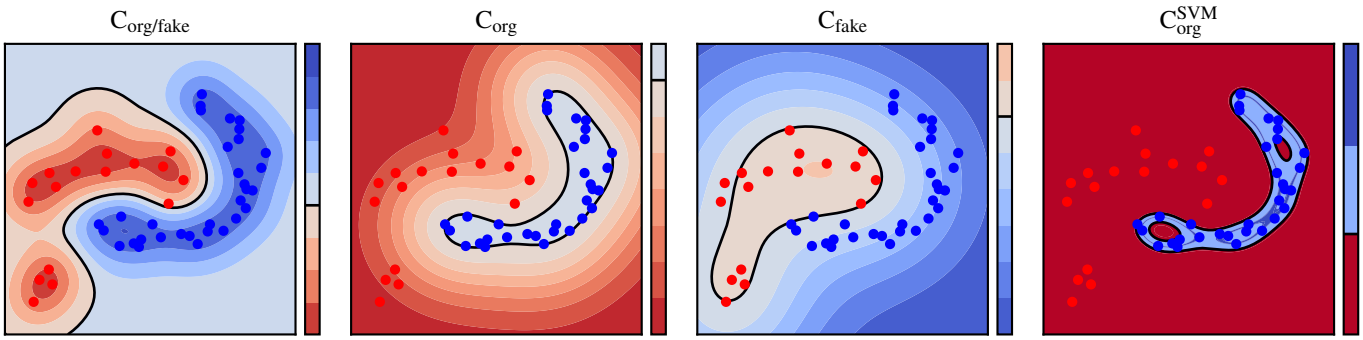

Fig. 2. Individual components of the 1.5C classifier with the synthetic 2-D toy example from [1]: The binary SVM $C_{org/fake}$ perfectly separates the classes but assigns unpopulated regions to the blue class (org). Conversely, C_{org} and C_{fake} enclose their respective classes at the cost of few misclassifications. The C_{org}^{SVM} tightly encloses all the blue samples. The unintuitive holes in the C_{org}^{SVM}'s acceptance region appear as the classifier rejects unfamiliar high-confidence decisions from the first-level classifiers.

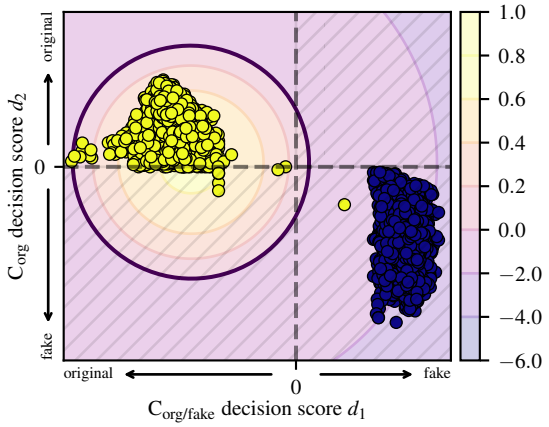

Fig. 3. Cross-section of C_{org}^{SVM}'s input space and acceptance region. Original and fake training images are shown as yellow and blue markers, respectively. The background color shows C_{org}^{SVM}'s decision score. The hashed area stretches across critical regions, where $C_{org/fake}$ or C_{org} spot fake images and should thus be rejected by a secure final classifier. Note that C_{org}^{SVM}'s acceptance region extends into the lower right quadrant, although the first-level $C_{org/fake}$ and C_{org} classify this region as fake.

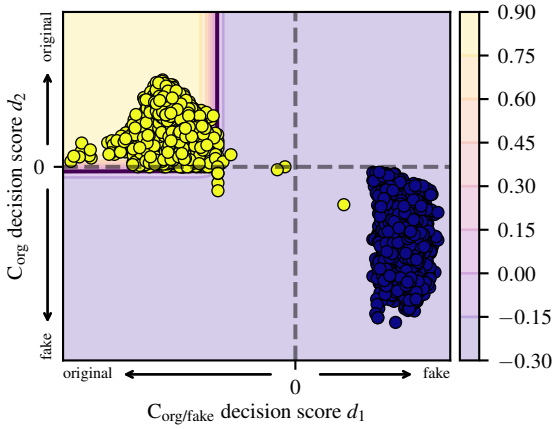

Fig. 4. Axis-aligned splits enable the analyst to transparently control the acceptance region of the proposed logical-`and` classifier. Moreover, the classifier accepts images that the first-level classifiers accept with high confidence.

first-level classifiers need not be rejected because these regions are expensive for an adversary to reach.

The latter two issues can be seen in Fig. 3. This figure displays a cross-section of C_{org}^{SVM}'s input space, which we obtained in our experiments for image rescaling detection as described later. The axes show the decision scores from $C_{org/fake}$ and C_{org}. The two dashed lines at $x = 0$ and $y = 0$ mark the decision boundaries of the two classifiers. At this cross-section, C_{fake} rejects inputs with a fixed decision score of -1 (C_{fake} classifies images as original). Yellow markers show the original training images projected onto this plane, while blue markers represent fake training images. The black contour line encloses C_{org}^{SVM}'s acceptance region. The background color indicates C_{org}^{SVM}'s decision score. C_{org}^{SVM} achieves almost perfect classification accuracy, but the shape and large size of its acceptance region leave more room than desirable for attackers to enter the acceptance region.

By including as many original training samples as possible, C_{org}^{SVM} extends its acceptance region into rejection regions of the first-level classifiers (hashed regions). The fact that the acceptance region even extends into the lower right quadrant shows that C_{org}^{SVM} accepts samples that all three first-level classifiers identify as a fake image. C_{org}^{SVM}'s confidence is highest in the center of the acceptance region (bright yellow). However, we argue that the final classifier's confidence should increase toward the top-left corner, where all first-level classifiers assign high confidence.

D. Proposed Solution: Logical-`and` Classifier

To mitigate these pitfalls, we propose replacing the final one-class SVM C_{org}^{SVM} with another one-class classifier C_{org}^{and} that uses axis-aligned splits. This classifier mimics a logical-`and` operation. An image is accepted as original only if it falls onto the correct side of all three splits; otherwise, it is rejected as a fake. Here, we describe two versions of this classifier with a "hard" and a "soft" decision function.

For the hard version, we set the split position based on the first-level classifiers' decision scores for the original training images. These decision scores are normalized to unit-variance. For each of the three dimensions, we split it such that 1% of the original training images with the lowest decision scores are classified as outliers. In cases where $C_{org/fake}$ can separate the two classes by a margin, this split ensures that an adversarial

example must not only cross the decision boundary but also the margin. Note that this split can also be fixed to 0 to reproduce the decision boundary of the corresponding classifier, or it can be adjusted to trade adversarial robustness for classification accuracy. Overall, we argue that axis-aligned splits are more transparent to the forensic analyst than the acceptance region of a one-class SVM.

As an approximation to this hard decision boundary, we implement a soft version that facilitates gradient-based attacks to evaluate the classifier's robustness. The decision function of this soft classifier is as follows:

$$\prod_{i=1}^{3} \sigma(s_i \cdot (d'_i - t_i)) \geq b \ , \tag{5}$$

where d'_i denotes the i-th first-level classifier's decision score after normalization and t_i denotes the split position. This decision function uses sigmoid functions $\sigma(\cdot)$ to approximate the non-differentiable step function at the split position. If one of the first-level classifiers rejects the input, the product of the sigmoids drops below 0.5. Therefore, setting $b = 0.5$ is a natural choice, although this offset can be adjusted based on training or validation data to achieve a given missed detection rate. The factor s_i controls the smoothness of the sigmoid activation and is shared among all three dimensions. Note that the sign of s_i depends on whether each first-level classifier's decision function increases or decreases toward the original images, *i.e.*, $s_2 > 0$ for $\mathrm{C_{org}}$, and $s_1, s_3 < 0$ for $\mathrm{C_{org/fake}}$ and $\mathrm{C_{fake}}$.

Figure 4 shows an example of the proposed classifier's acceptance region. In this example, we set $|s_i| = 5$ and calibrate b such that the classifier achieves a missed detection rate of 0.01 on the validation set.

This classifier avoids the pitfalls of a one-class SVM as the final classifier. First, it normalizes the decision scores from the first-level classifiers such that none of them is ignored. Second, it accepts images with high decision scores from the first-level classifiers, therefore avoiding holes in high-confidence regions, which are difficult for an attacker to reach anyway. Third, it transparently allows an analyst to configure the split thresholds, thereby avoiding unintuitive acceptance regions, as shown in Fig. 3.

V. Experiments and Results

This section evaluates the robustness of the 1.5C classifier against evasion attacks. The goal of the attacker is to have a fake image be classified as an original image. We assume that the attacker has perfect knowledge about the classifier including feature extraction, model architecture, and trained weights. Studying this perfect knowledge scenario allows assessing the security in a worst-case scenario. The application scenario is detecting global image manipulation detection, similar to related work [4], [30], [31]. First, we evaluate the proposed attack and defense for the application scenario of rescaling detection as in [4]. Then, we briefly present similar results on the forensic tasks of blurring and median filtering detection to demonstrate the generality of our findings.

A. Experimental Setup

The RAISE 1k dataset consists of 999 decodable images. All images are converted to grayscale. From each image, we randomly select 25 patches with a side length of 512 pixels. For simplicity, we discard and re-draw patches with more than 10% of saturated pixels and patches with more than 50% of zeros in their gradient image. For each patch, we keep an original version and create a fake version. For the task of rescaling detection, fake patches are scaled by a factor of 1.3 using bicubic interpolation as in [4], center-cropped to side length 512, and eventually rounded to the unsigned byte range. For the task of blur detection, the fake images use a Gaussian kernel with a standard deviation of 0.5, as in [33]. For median filtering detection, the fake images use a kernel size of 3×3. The patches from 799 randomly selected images are used for training, the patches from 100 images are used for validation, and the patches from the last 100 images are used for testing.

B. Training and Selection of Hyper-Parameters

We replicate the training protocol from [4]. In particular, for the binary SVM with RBF kernel, we search for the regularization parameter $C \in [2^{-5}, \ldots, 2^{15}]$ and the kernel width $\kappa \in [2^{-15}, \ldots, 2^{3}]$ using 5-fold cross-validation. The grid search selects the hyper-parameters that provide the best classification accuracy.

The regularization and kernel width parameters of the one-class SVMs consider the range $\nu \in [2^{-10}, \ldots, 2^{0}], \kappa \in [2^{-10}, \ldots, 2^{10}]$. While the original paper skips cross-validation to reduce training complexity [4], we also use 5-fold cross-validation to select ν and κ. The grid search finally selects those hyper-parameters that minimize the weighted error probability as described in Sec. IV-B.

The SVMs are trained using *scikit-learn 0.23.1*. The SVM decision functions are re-implemented in *PyTorch 1.7*. We modify the DDN attack from *Foolbox 3.3.1* [45] to include quantization at the end of each iteration and run the attack on an NVIDIA GeForce 2080 Ti GPU.

For the DDN-BPDA attack, we use an initial perturbation budget of $\epsilon = 0.3$, a multiplicative factor $\gamma = 0.005$ to increase or decrease the perturbation budget, and 1 000 attack iterations. For the BPDA, we use a softmax steepness of $\alpha = 0.3$. The finite differences attack uses up to 100 iterations but usually converges earlier. In each attack iteration, we modify a fraction $K = 0.05$ of pixels. These two attack hyper-parameters are subject to an ablation study in Sec. V-E.

C. Evaluation Protocol

We compare three classifiers: A binary SVM denoted as 2C-SVM, the original 1.5C classifier with the final one-class SVM $\mathrm{C_{org}^{SVM}}$, and the 1.5C classifier with the proposed logical-`and` classifier $\mathrm{C_{org}^{and}}$. The 2C-SVM is identical to the $\mathrm{C_{org/fake}}$ as part of the 1.5C classifier, but we use these different names to disambiguate between a standalone binary classifier and the first-level component of the 1.5C classifier.

All attacks are run with the same $S = 100$ randomly chosen fake test set patches. We compare attacks and adversarial

robustness in terms of the L2 distortion required to deceive the detector. The L2 distortion is calculated as follows:

$$d(\boldsymbol{x}_{\text{fake}}, \boldsymbol{x}_{\text{adv}}) = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(x_{\text{fake},i} - x_{\text{adv},i})^2} \ , \qquad (6)$$

where M is the number of pixels, $\boldsymbol{x}_{\text{fake}}$ denotes the fake image prior to the attack, and $\boldsymbol{x}_{\text{adv}}$ denotes the image with adversarial perturbation.

For each detector, we report results for six increasingly tight acceptance regions. Therefore, we configure the attacks to push adversarial examples not only across the decision boundary but further into the acceptance region by a safety margin τ. Thus, the attacker's goal is to obtain $\boldsymbol{x}_{\text{adv}}$ from $\boldsymbol{x}_{\text{fake}}$ with the following optimization problem:

$$\min d(\boldsymbol{x}_{\text{fake}}, \boldsymbol{x}_{\text{adv}}) \quad \text{s.t.} \quad f(\boldsymbol{x}_{\text{adv}}) \geq \tau \ . \qquad (7)$$

Here, $f(\cdot)$ can denote any detector, *e.g.*, the $\text{C}_{\text{org}}^{\text{SVM}}$ as in Fig. 1, $\text{C}_{\text{org}}^{\texttt{and}}$, or the 2C-SVM. Note that the sign of the 2C-SVM is flipped such that original images are assigned positive outputs and fake images are assigned negative outputs. The attack may not always be successful. Hence, the attack success rate (ASR) reports the percentage of the $S = 100$ test images for which the attack succeeded.

Exceeding the decision boundary by a safety margin can be important in case of modifications by the distribution channel or defensive pre-processing, or in case the defender rejects low-confidence decisions. Furthermore, in a black-box setting with access to a surrogate model, exceeding the decision boundary can help to overcome approximation deficiencies of the surrogate model. In fact, several related works showed that exceeding the decision boundary by a safety margin is needed to improve attack transferability [30], [46], [47]. Hence, studying classifier robustness with different safety margins is a very relevant scenario.

To compare safety margins across different classifiers, τ is set as high as the $0, 1, 25, 50, 75$, and the 90th percentile of prediction scores assigned to original images from the validation set. Here, 0 means that adversarial examples have just crossed the decision boundary. A safety margin of 1 corresponds to an acceptance region where 1% of the original validation images are rejected, *i.e.*, the missed detection rate with this kind of tightened acceptance region would be 0.01.

D. Classification Accuracy

We report results for the task of rescaling detection. The 2C-SVM's test accuracy is 1.0. The test accuracy of the 1.5C classifier with $\text{C}_{\text{org}}^{\text{SVM}}$ is 0.9988. Individually, the 2C classifier $\text{C}_{\text{org/fake}}$, the original image one-class detector C_{org}, and the fake image one-class detector C_{fake} reach accuracies of 1.0, 0.9956, and 0.9934, respectively. In comparison, the 1.5C classifier with logical-`and` classifier $\text{C}_{\text{org}}^{\texttt{and}}$ attains an accuracy of 0.9976. These numbers demonstrate that all classifiers achieve high performance in the absence of attacks.

TABLE I
L2 DISTORTION AND ATTACK SUCCESS RATE (ASR) OF 100 ATTACKED IMAGES CRAFTED AGAINST THE 2C-SVM AND AGAINST THE 1.5C CLASSIFIER $\text{C}_{\text{ORG}}^{\text{SVM}}$. PUSHING ADVERSARIAL IMAGES FURTHER INTO THE ACCEPTANCE REGION BY A SAFETY MARGIN REQUIRES MORE DISTORTION. OUR PROPOSED DDN-BPDA ATTACK FINDS ADVERSARIAL EXAMPLES WITH LOWER DISTORTION THAN THE FINITE DIFFERENCES ATTACK. WHILE THE 1.5C CLASSIFIER IS MORE ROBUST AGAINST LOW-CONFIDENCE ADVERSARIAL IMAGES WITH A SAFETY MARGIN OF ZERO, 2C-SVM IS MORE ROBUST AGAINST HIGH-CONFIDENCE ADVERSARIAL EXAMPLES.

		Finite diff. (FD) [30]		DDN-BPDA	
Clf.	Margin	ASR	L2 distortion	ASR	L2 distortion
2C-SVM	0	1.00	0.207 ± 0.075	1.00	0.193 ± 0.052
	1	1.00	0.333 ± 0.108	1.00	0.282 ± 0.072
	25	1.00	0.370 ± 0.119	1.00	0.304 ± 0.078
	50	1.00	0.403 ± 0.134	1.00	0.322 ± 0.083
	75	1.00	0.440 ± 0.158	1.00	0.340 ± 0.093
	90	1.00	0.496 ± 0.204	0.99	0.355 ± 0.092
$\text{C}_{\text{org}}^{\text{SVM}}$	0	1.00	0.244 ± 0.082	1.00	0.224 ± 0.057
	1	1.00	0.261 ± 0.083	1.00	0.236 ± 0.058
	25	0.99	0.306 ± 0.083	1.00	0.268 ± 0.057
	50	0.98	0.333 ± 0.082	1.00	0.285 ± 0.057
	75	0.97	0.366 ± 0.081	1.00	0.305 ± 0.055
	90	0.97	0.396 ± 0.083	1.00	0.329 ± 0.049

E. Attack Evaluation

We first compare the finite differences (FD) attack [30] and our proposed DDN-BPDA attack and then provide ablation studies for these two attacks.

1) Finite Differences vs. DDN-BPDA Attack: The columns of Table I compare the FD and DDN-BPDA attacks in terms of the L2 distortions required to reach the adversarial region with a specified safety margin. The FD attack achieved an ASR of 1.0 against the 2C-SVM and an ASR of 0.97 to 1.0 against the $\text{C}_{\text{org}}^{\text{SVM}}$ across different safety margins. The DDN-BPDA attack succeeded on all images presented to $\text{C}_{\text{org}}^{\text{SVM}}$, and it achieved an ASR between 0.99 and 1.0 against the 2C-SVM. The table only reports distortions for successfully attacked images. For both attacks, we verified that unsuccessfully attacked images can be turned into adversarial examples by tuning the attack hyper-parameters.

As expected, the L2 distortion for a successful attack increases with the desired safety margin for both attacks and classifiers. Overall, the proposed DDN-BPDA attack yields adversarial images with lower average L2 distortion than the FD attack. In the remainder of this section, we first provide an ablation study for the two attacks, and then compare the robustness of 2C-SVM and $\text{C}_{\text{org}}^{\text{SVM}}$.

2) Ablation Study for the Finite Differences Attack: Instead of pushing all of the pixels by a small step size into the gradient direction, such as conventional gradient descent methods, the FD attack updates a fraction of the pixels by a fixed step size of 1 to account for the integer nature of pixel values. In the original paper [30], the authors found that changing no more than a fraction $K = 0.2$ of the pixels per iteration yields adversarial examples with low distortion. For

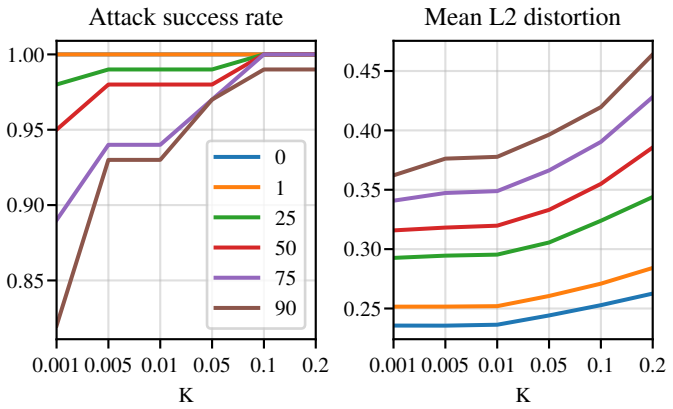

Fig. 5. The figures show the attack success rate (left) and the mean L2 distortion (right) as a function of the fraction K of pixels modified per attack iteration. We configure the attack to achieve decision scores as high as the $0, 1, 25, 50, 75$ and 90th percentiles of the original validation images. Smaller values of K lead to adversarial images with lower distortion but the attack more often becomes stuck at non-adversarial minima.

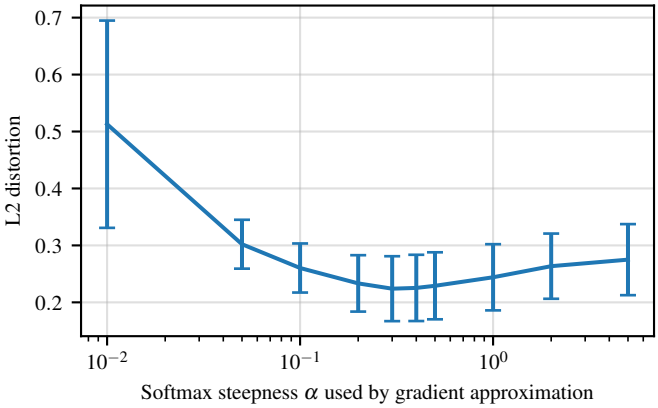

Fig. 6. The hyper-parameter α controls the softmax steepness of the backward pass differentiable approximation (BPDA). Setting $\alpha = 0.3$ allows the DDN-BPDA attack with a safety margin of 0 to produce adversarial images with low average L2 distortion against the $\mathrm{C}^{\mathrm{SVM}}_{\mathrm{org}}$.

a fair comparison to our proposed attack with tuned hyper-parameters, we evaluate the FD attack with several lower values of K. Figure 5 shows the attack success rate and the average L2 distortion of the FD attack against $\mathrm{C}^{\mathrm{SVM}}_{\mathrm{org}}$ as a function of K. The average L2 distortion decreases for lower values of K. This can be explained by the dependencies among pixels that are captured by SPAM features. Therefore, changing only a few pixels before re-computing the gradient in the next iteration can yield adversarial examples with lower distortion. At the same time, however, the attack occasionally becomes stuck at non-adversarial minima. For Tab. I, we chose $K = 0.05$ because the attack success rate is above 0.95. Additionally, changing fewer pixels per iteration increases the total number of iterations until an adversarial example is found. By decreasing K from 0.2 to 0.001, the average number of iterations to produce adversarial examples rises from 1 to 29 for safety margin 0, and even 61 for the highest safety margin. Although we optimized our implementation to re-compute only those SPAM features that are affected by a single pixel change, calculating the gradient approximation is still very expensive. Therefore, K can be seen as a hyper-parameter that trades distortion for attack speed and attack success. We note that even with a very small $K = 0.001$, the proposed attack still achieves lower distortion and a higher success rate.

3) Ablation Study for BPDA α: During the backward pass, the non-differentiable SPAM feature quantization is implemented via vector quantization, which assigns each pixel to a template vector. The hard assignment, which disrupts the gradient flow, can be approximated through a differentiable soft assignment, as explained in Sec. III-C. The hyper-parameter α controls the softmax steepness: a higher α provides a more precise approximation, and a lower α improves the gradient flow.

Figure 6 shows the average L2 distortion as a function of α after attacking $\mathrm{C}^{\mathrm{SVM}}_{\mathrm{org}}$ with a safety margin of 0. The lowest L2 distortion is achieved by $\alpha = 0.3$. For the rest of this paper, we use $\alpha = 0.3$.

4) Finite Differences Attack with the BPDA Gradient: We also evaluate the FD attack algorithm with our BPDA gradient approximation, but find that the FD attack does not benefit from this gradient approximation. Intuitively, BPDA should give a more precise approximation because it provides the analytical gradient with only one transformation approximated, while Chen *et al*. approximate the gradient using finite differences. Therefore, we adapt the algorithm by Chen *et al*. and replace the finite differences gradient with the proposed BPDA gradient.

We find that this "hybrid" attack yields an average L2 distortion larger than both the original FD and the DDN-BPDA attack. With a safety margin of 0, for example, the hybrid attack requires an average L2 distortion of 0.281, while the FD and DDN attacks require 0.244 and 0.224, respectively. We hypothesize that the FD attack does not benefit from the more precise gradient approximation because of the relatively large step size of 1. In contrast, the finite differences gradient approximation evaluates the change in the classifier's decision score by changing each pixel by a step of 1 and is therefore better suited for this kind of attack algorithm.

Thus, we conclude that the proposed DDN-BPDA attack yields adversarial examples with lower distortion not because of the more precise gradient approximation but because it continuously optimizes for lower distortion even after it has found an adversarial example. Nevertheless, the BPDA gradient enables this attack because it is much faster to calculate than the finite differences gradient approximation. Furthermore, it can readily make use of GPU acceleration without requiring a specialized CUDA implementation. For the rest of the paper, we only use the proposed DDN-BPDA attack.

F. Defense Evaluation

In this section, we compare the adversarial robustness of three classifiers: a binary SVM (2C-SVM), the original 1.5C classifier $\mathrm{C}^{\mathrm{SVM}}_{\mathrm{org}}$, and the proposed variant $\mathrm{C}^{\mathtt{and}}_{\mathrm{org}}$. We begin by comparing the first two classifiers 2C-SVM and $\mathrm{C}^{\mathrm{SVM}}_{\mathrm{org}}$, again using the results from Tab. I though now with a rowwise comparison. A similar comparison is reported in [4], but we

TABLE II
ERROR RATES OF THE COMPONENTS OF THE 1.5C CLASSIFIER UNDER ATTACK. THE ADVERSARIAL IMAGES WERE CRAFTED AGAINST C_{ORG}^{SVM} WITH INCREASINGLY TIGHT SAFETY MARGINS. EVEN THOUGH C_{ORG} SPOTS MANY ADVERSARIAL IMAGES, C_{ORG}^{SVM} STILL ACCEPTS THEM.

Attacked clf.	Margin	ASR	$C_{org/fake}$	C_{org}	C_{fake}
C_{org}^{SVM}	0	1.00	1.00	0.19	0.77
	1	1.00	1.00	0.29	0.94
	25	1.00	1.00	0.36	0.94
	50	1.00	1.00	0.49	1.00
	75	1.00	1.00	0.55	1.00
	90	1.00	1.00	0.62	1.00

additionally evaluate different safety margins. This in-depth analysis reveals the pitfalls of the final one-class classifier. Afterwards, we show that the proposed C_{org}^{and} mitigates these pitfalls and achieves higher robustness.

1) 2C-SVM vs. 1.5C Classifier: Table I shows that for a safety margin of 0, more distortion is required to cross the decision boundary of C_{org}^{SVM} than the boundary of 2C-SVM. Surprisingly, this behavior changes for higher safety margins. Here, C_{org}^{SVM} is less robust than the 2C-SVM. For example, to achieve a prediction score as high as the lowest 25 percent of the original validation images, attacking 2C-SVM requires an L2 distortion of 0.304, while attacking C_{org}^{SVM} only requires 0.268.

Overall, we assume that the decreased robustness of the C_{org}^{SVM} for higher safety margins is due to the ellipsoidal acceptance region shape, as described in Sec. IV-C3. The final one-class classifier C_{org}^{SVM} assigns the highest decision scores to samples that are predicted with medium scores by the first-level classifiers. Hence, an attacker can produce high-confidence adversarial examples against C_{org}^{SVM} although these do not need to fool the first-level classifiers, which costs less distortion.

2) Which of the 1.5C Classifier's Components Is Fooled?: We now investigate the failure of the specific components of the 1.5C classifier, namely $C_{org/fake}$, C_{org}, and C_{fake} (cf. Fig. 1). Table II shows which of these components are fooled by the 100 adversarial examples crafted against C_{org}^{SVM}.

While all adversarial examples fool the $C_{org/fake}$ component, only a fraction thereof is accepted by C_{org}. For a safety margin of 0, even C_{fake} recognizes 23 images as fake, but C_{org}^{SVM} still accepts these images. This may be due to the hyper-parameter optimization, which enlarges C_{org}^{SVM}'s acceptance region into rejection regions of the first-layer classifiers. In any case, the fact that C_{org}^{SVM} accepts attacked images despite one or more first-level classifiers identifying them indicates room for improvement for the robustness of the 1.5C classifier.

3) Logical-and Classifier: We now evaluate the robustness of the proposed logical-and classifier C_{org}^{and}, which replaces the final one-class SVM in the 1.5C architecture. The same first-level classifiers are used as for the C_{org}^{SVM}. The bias b is set such that C_{org}^{and} achieves the same missed detection rate as the C_{org}^{SVM} on the validation set. Because the decision function of the logical-and classifier is very steep, the attack minimizes the logarithm of C_{org}^{and}'s decision function from Eq. 5.

TABLE III
L2 DISTORTION OF THE ADVERSARIAL EXAMPLES CREATED USING DDN-BPDA AGAINST THE LOGICAL-AND CLASSIFIER C_{ORG}^{AND}.

Margin	ASR	L2 distortion
0	1.00	0.264 ± 0.070
1	1.00	0.280 ± 0.073
25	1.00	0.306 ± 0.072
50	1.00	0.343 ± 0.071
75	1.00	0.407 ± 0.099
90	0.99	0.476 ± 0.164

TABLE IV
ERROR RATES OF THE COMPONENTS OF THE 1.5C CLASSIFIER WITH ADVERSARIAL IMAGES CRAFTED AGAINST C_{ORG}^{AND}. WITH A SAFETY MARGIN OF 1 OR HIGHER, DECEIVING THE LOGICAL-AND CLASSIFIER REQUIRES DECEIVING ALL THREE FIRST-LEVEL CLASSIFIERS, WHICH REQUIRES HIGHER DISTORTION.

Attacked clf.	Margin	ASR	$C_{org/fake}$	C_{org}	C_{fake}
C_{org}^{and}	0	1.00	1.00	0.83	0.85
	1	1.00	1.00	0.99	0.98
	25	1.00	1.00	1.00	1.00
	50	1.00	1.00	1.00	1.00
	75	1.00	1.00	1.00	1.00
	90	0.99	0.99	0.99	0.99

Table III reports the L2 distortion against the proposed classifier with six increasingly tight security margins. Compared to the C_{org}^{SVM} (cf. Tab. I, bottom right), adversarial images against the logical-and classifier C_{org}^{and} require more distortion across all safety margins. For a safety margin of 0, adversarial images against the C_{org}^{and} require an average L2 distortion of 0.264, while adversarial images against the C_{org}^{SVM} only need 0.224. For a safety margin of 1, C_{org}^{and} is slightly less robust than

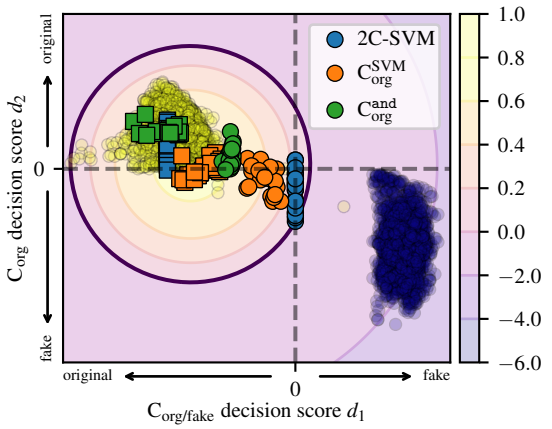

Fig. 7. 25 randomly selected adversarial examples crafted against the 2C-SVM (blue), the 1.5C classifier C_{org}^{SVM} (orange), and the proposed C_{org}^{and} (green). The background color indicates the decision scores of the C_{org}^{SVM}. The circle markers show adversarial examples that have just crossed the individual classifier's decision boundary. The square markers show adversarial examples that were pushed into the classifier's acceptance region as far as 75% of the original validation images.

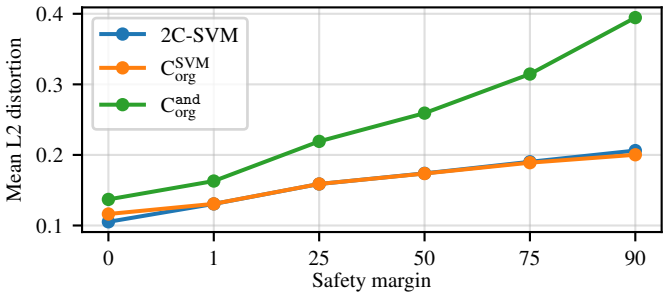

Fig. 8. Amount of distortion required to deceive the three classifiers for blur detection. The 2C-SVM and C_{org}^{SVM} show similar robustness because of the missing normalization pitfall. The proposed C_{org}^{and} overcomes this pitfall, thereby achieving greater robustness.

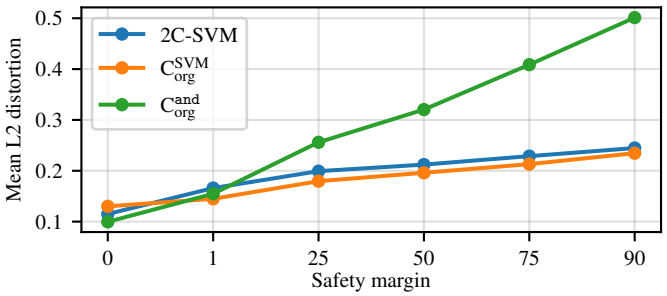

Fig. 9. Amount of distortion required to deceive the three classifiers for median filtering detection. At safety margins of 0 and 1, C_{org}^{and} is less robust than the 2C-SVM because it requires an excessive acceptance region to match the targeted missed detection rate. At higher safety margins, C_{org}^{and} outperforms the other classifiers.

the 2C-SVM. This is because C_{org}^{and} rejects images with very low d_2 and must therefore accept images with slightly higher d_1 than 2C-SVM to match the targeted missed detection rate. However, C_{org}^{and} outperforms the other classifiers in all other safety margins.

Table IV illustrates which of the 1.5C classifier's individual components are fooled by adversarial images crafted against C_{org}^{and}. For a safety margin of 0, C_{org} still detects 17 of the adversarial examples. The defense can be improved by tightening the acceptance region. For example, by setting b such that 1 percent of the original images is rejected, 99 of 100 attacked images deceive C_{org}. For higher safety margins, all successful attacks deceive all the individual classifiers. This result indicates that deceiving C_{org}^{and} requires fooling all three first-level classifiers, thus requiring higher distortion.

4) Comparison of the Three Classifiers: Figure 7 presents 25 randomly selected adversarial examples crafted against the 2C-SVM, against the C_{org}^{SVM}, and against C_{org}^{and}. The background shows a cross-section of the input space to the final one-class classifier, colored by the C_{org}^{SVM}, as in Fig. 3. The original and fake training images are shown as semi-transparent yellow and blue markers in the background. The circle markers show adversarial images that just crossed the three classifiers' decision boundaries. The square markers show adversarial images where the attacker's goal was to push adversarial examples as far into the classifier's acceptance region as 75 percent of the original validation images. The adversarial examples against the 2C-SVM (blue) align with the y-axis. As expected, the blue circles coincide with $d_1 = 0$. Adversarial images crafted against the C_{org}^{SVM} are shown in orange. With higher safety margins, the attacked examples move closer to the acceptance region's midpoint. Note that the orange circles only appear inside the acceptance region due to the 2-D representation, but they actually lie on the 3-D decision boundary. The adversarial examples against C_{org}^{and}, shown in green, are close to the original training points. The square markers show that a tighter acceptance region forces the attacker to push adversarial examples toward the top-left corner, which requires more distortion.

G. Case Study: Gaussian Blur Detection

We demonstrate that the pitfalls of C_{org}^{SVM} also occur in other forensic applications, such as the detection of Gaussian

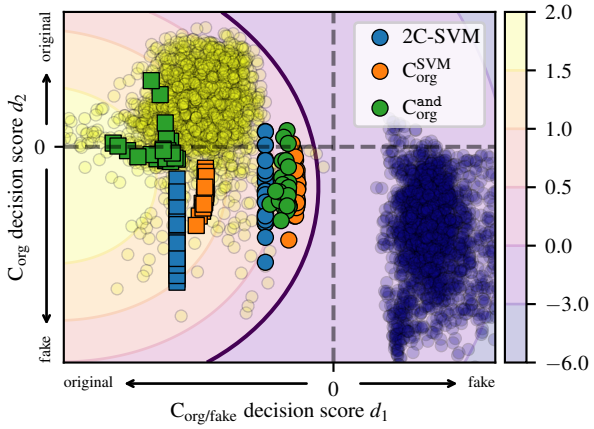

Fig. 10. A cross-section of the 1.5C classifier's acceptance region for median filtering detection is shown. The circles show 25 randomly selected adversarial examples crafted with safety margin 1 against the three classifiers. The squares show adversarial examples with safety margin 75. Because C_{org} achieves an accuracy of only 0.9506, C_{org}^{and} was configured to accept low-confidence rejections of C_{org} to match the targeted missed detection rate.

blur. The 1.5C classifier is trained analogously to the previous experiment. The test accuracies of $C_{org/fake}$, C_{org}, C_{fake}, and C_{org}^{SVM} are 1, 0.9686, 0.9296, and 0.9992, respectively. Figure 8 compares the attack robustness of the C_{org}^{SVM} to the 2C-SVM and C_{org}^{and}. The attack succeeded for all images. In this case, deceiving the C_{org}^{SVM} and the 2C-SVM requires a similar amount of distortion. This is because the decision scores d_2, d_3 of C_{org} and C_{fake} are smaller by almost two orders of magnitude than the decision score d_1 from $C_{org/fake}$. As a result, the C_{org}^{SVM} ignores the output of the two first-level one-class classifiers. The proposed C_{org}^{and} normalizes the decision scores of the first-level classifiers and therefore achieves greater robustness.

H. Case Study: Median Filtering Detection

As a third application, we evaluate median filtering detection as in [4]. Compared to the previous experiments, the grid search for all one-class SVMs used a finer step size for the kernel width, *i.e.*, $\kappa \in [2^{-10}, 2^{-9.5}, \ldots, 2^{9.5}, 2^{10}]$, because this fine-grained search range considerably increased the classification accuracy. The test accuracies of $C_{org/fake}$, C_{org}, C_{fake}, and the C_{org}^{SVM} are 0.9998, 0.9506, 0.9788, and 0.9992, respectively. Figure 9 shows the average L2 distortion required

to fool the 2C-SVM, $\mathrm{C}_{\mathrm{org}}^{\mathrm{SVM}}$, and $\mathrm{C}_{\mathrm{org}}^{\mathtt{and}}$. The attack succeeded for all images. Analogous to the results for rescaling detection, the $\mathrm{C}_{\mathrm{org}}^{\mathrm{SVM}}$ is more robust than the 2C-SVM with no safety margin, but the 2C-SVM outperforms the $\mathrm{C}_{\mathrm{org}}^{\mathrm{SVM}}$ at higher safety margins. As before, the proposed $\mathrm{C}_{\mathrm{org}}^{\mathtt{and}}$ is configured to match the missed detection rate of the $\mathrm{C}_{\mathrm{org}}^{\mathrm{SVM}}$ on the validation set. If configured to achieve the same low missed detection rate as the $\mathrm{C}_{\mathrm{org}}^{\mathrm{SVM}}$ with no safety margin, $\mathrm{C}_{\mathrm{org}}^{\mathtt{and}}$ requires an extensive acceptance region, thereby sacrificing its robustness. With safety margin 1, $\mathrm{C}_{\mathrm{org}}^{\mathtt{and}}$ already shows greater robustness than the $\mathrm{C}_{\mathrm{org}}^{\mathrm{SVM}}$. Nevertheless, $\mathrm{C}_{\mathrm{org}}^{\mathtt{and}}$ is slightly less robust than the 2C-SVM because it rejects images with very low d_2, and therefore, accepts slightly larger values of d_1 than the 2C-SVM to match the targeted missed detection rate. However, this cut-off value for d_1 and d_2 could be tuned further. For higher safety margins, $\mathrm{C}_{\mathrm{org}}^{\mathtt{and}}$ outperforms both the 2C-SVM and $\mathrm{C}_{\mathrm{org}}^{\mathrm{SVM}}$.

This can also be seen in Fig. 10, which shows 25 randomly selected adversarial examples against the three classifiers for safety margins of 1 (circles) and 75 (squares). Because $\mathrm{C}_{\mathrm{org}}$ misses many original validation images (yellow circles with $d_2 < 0$), $\mathrm{C}_{\mathrm{org}}^{\mathtt{and}}$ was configured to accept low-confidence rejections of $\mathrm{C}_{\mathrm{org}}$ to achieve the targeted missed detection rate.

This case study shows that achieving a good trade-off between classification accuracy and robustness requires the first-level classifiers to have a low missed detection rate; otherwise the logical-and classifier needs to accept low-confidence rejections by the first-level classifiers in order to achieve high classifier accuracy. Thus, one direction for future work is to minimize the missed detection rate of the first-level detectors at the cost of more false alarms. These false alarms may be identified by another first-level detector and would therefore be rejected by the logical-and classifier.

I. Comparison to a CNN Baseline

For comparison, we trained individual CNN detectors for the three manipulation detection tasks. We used EfficientNet-B0, which is currently one of the most popular backbones in related tasks [48], [49] and the same training, validation, and test images as described in Sec. V-A. The learning rate was set to 0.001, and we used a batch size of 16. Training was stopped when the validation loss did not decrease for five consecutive epochs. The CNN achieved an accuracy of 1.0 in all three manipulation detection tasks. Given the large capacity of the network, the good performance is not surprising, but another important quality of a forensic detector is its adversarial robustness. To evaluate the adversarial robustness, we used the same evaluation protocol and the DDN attack with naive rounding at the end of each iteration as described above. Attacking the CNN for rescaling detection required a mean L2 distortion of 0.041 for a safety margin of 0 and 0.0721 for a safety margin of 90. In comparison, attacking $\mathrm{C}_{\mathrm{org}}^{\mathtt{and}}$ for this task required an L2 distortion between 0.264 and 0.476 (cf. Tab. III). Similarly, attacking the CNN for median filtering detection and Gaussian blur detection required an L2 distortion of only 0.024 and 0.026 for a safety margin of 0, and 0.076 and 0.058 for a safety margin of 90. In comparison, attacking $\mathrm{C}_{\mathrm{org}}^{\mathtt{and}}$ for these two tasks requires substantially higher distortion (cf. Fig. 9 and Fig. 8).

Overall, the comparison demonstrates that evading the $\mathrm{C}_{\mathrm{org}}^{\mathtt{and}}$ with SPAM features requires considerably more distortion than evading a single CNN. Although it is beyond the scope of this paper, the idea of combining an ensemble of classifiers can easily be transferred to deep learning. In particular, diversity among the ensemble members can be achieved through different strategies, including random initialization, training data subsampling, or varying network architectures. The last strategy appears particularly promising, as recent work indicates that adversarial examples in multimedia forensics fail to transfer between network architectures [47]. Therefore, combining multiple CNN architectures with the logical-`and` classifier can be an interesting direction for increasing the adversarial robustness of CNN-based forensic detectors.

VI. Conclusion

Combining the decisions of multiple classifiers can increase the robustness against evasion attacks. However, subtle pitfalls in fusing classifier outputs compromise the performance and security of these architectures. In this paper, we outline three pitfalls of the one-and-a-half-class classifier. These pitfalls can be mitigated by replacing the final one-class SVM with a simple logical-`and` classifier using axis-aligned splits. Our experimental evaluation demonstrates the increased robustness of this classifier compared to the original one-and-a-half-class classifier. To evaluate detector robustness with SPAM features, we additionally develop a white-box attack that achieves lower distortion than the previous attack against SPAM feature-based detectors. In future work, we plan to examine the robustness and reliability of other intrinsically secure detectors against unseen processing and adversarial attacks.

References

[1] B. Biggio, I. Corona, Z.-M. 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 *Multiple Classifier Systems*. Cham: Springer International Publishing, 2015, pp. 168–180.

[2] E. Quiring, D. Arp, and K. Rieck, "Forgotten siblings: Unifying attacks on machine learning and digital watermarking," in *European Symposium on Security and Privacy*. IEEE, Apr. 2018, pp. 488–502.

[3] D. Maiorca, P. Russu, I. Corona, B. Biggio, and G. Giacinto, "Detection of malicious scripting code through discriminant and adversary-aware api analysis," in *1st Italian Conference on CyberSecurity*, vol. 1816, 2017, pp. 96–105.

[4] M. Barni, E. Nowroozi, and B. Tondi, "Improving the security of image manipulation detection through one-and-a-half-class multiple classification," *Multimedia Tools and Applications*, vol. 79, no. 3-4, pp. 2383–2408, Nov. 2019.

[5] R. Böhme and M. Kirchner, "Counter-forensics: Attacking image forensics," in *Digital Image Forensics: There is More to a Picture than Meets the Eye*, H. T. Sencar and N. Memon, Eds. Springer, 2013.

[6] M. Barni, K. Kallas, and B. Tondi, "A new backdoor attack in CNNs by training set corruption without label poisoning," in *IEEE International Conference on Image Processing*, Aug. 2019, pp. 101–105.

[7] M. Barni, M. C. Stamm, and B. Tondi, "Adversarial multimedia forensics: Overview and challenges ahead," in *European Signal Processing Conference*, Sep. 2018, pp. 962–966.

[8] H. Li, W. Luo, X. Qiu, and J. Huang, "Identification of various image operations using residual-based features," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 1, pp. 31–45, Aug. 2018.

[9] H. C. Nguyen and S. Katzenbeisser, "Performance and robustness analysis for some re-sampling detection techniques in digital images," in *Digital Forensics and Watermarking*. Springer Berlin Heidelberg, 2012, pp. 387–397.

[10] W. Fan, S. Agarwal, and H. Farid, "Rebroadcast attacks: Defenses, reattacks, and redefenses," in *European Signal Processing Conference*, Dec. 2018, pp. 942–946.

[11] M. Barni, M. Fontani, and B. Tondi, "A universal technique to hide traces of histogram-based image manipulations," in *ACM Workshop on Multimedia and Security*, New York, NY, USA, Sep. 2012, pp. 97–104.

[12] ——, "Universal counterforensics of multiple compressed JPEG images," in *Digital-Forensics and Watermarking*. Springer International Publishing, Oct. 2014, pp. 31–46.

[13] C. Pasquini, P. Comesaña-Alfaro, F. Pérez-González, and G. Boato, "Transportation-theoretic image counterforensics to first significant digit histogram forensics," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2014, pp. 2699–2703.

[14] P. Comesaña-Alfaro and F. Pérez-González, "The optimal attack to histogram-based forensic detectors is simple(x)," in *IEEE International Workshop on Information Forensics and Security*, Dec. 2014, pp. 137–142.

[15] M. Kirchner and R. Böhme, "Hiding traces of resampling in digital images," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 4, pp. 582–592, Nov. 2008.

[16] G. Cao, Y. Zhao, R. Ni, and H. Tian, "Anti-forensics of contrast enhancement in digital images," in *ACM Workshop on Multimedia and Security*, 2010, pp. 25–34.

[17] M. C. Stamm and K. R. Liu, "Anti-forensics of digital image compression," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1050–1065, 2011.

[18] M. Fontani and M. Barni, "Hiding traces of median filtering in digital images," in *European Signal Processing Conference*, Oct. 2012, pp. 1239–1243.

[19] I. Amerini, M. Barni, R. Caldelli, and A. Costanzo, "SIFT keypoint removal and injection for countering matching-based image forensics," in *ACM Workshop on Information Hiding and Multimedia Security*, 2013, pp. 123–130.

[20] D. Güera, Y. Wang, L. Bondi, P. Bestagini, S. Tubaro, and E. J. Delp, "A counter-forensic method for CNN-based camera model identification," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 1840–1847.

[21] D. Gragnaniello, F. Marra, G. Poggi, and L. Verdoliva, "Analysis of adversarial attacks against CNN-based image forgery detectors," in *European Signal Processing Conference*, Sep. 2018, pp. 967–971.

[22] D. Kim, H.-U. Jang, S.-M. Mun, S. Choi, and H.-K. Lee, "Median filtered image restoration and anti-forensics using adversarial networks," *IEEE Signal Processing Letters*, vol. 25, no. 2, pp. 278–282, 2018.

[23] Y. Luo, H. Zi, Q. Zhang, and X. Kang, "Anti-forensics of JPEG compression using generative adversarial networks," in *European Signal Processing Conference*, Dec. 2018, pp. 952–956.

[24] C. Chen, X. Zhao, and M. C. Stamm, "Generative adversarial attacks against deep-learning-based camera model identification," *IEEE Transactions on Information Forensics and Security*, 2019.

[25] D. Cozzolino, J. Thies, A. Rössler, M. Nießner, and L. Verdoliva, "SpoC: Spoofing camera fingerprints," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2021, pp. 990–1000.

[26] J. Rony, L. G. Hafemann, L. S. Oliveira, I. B. Ayed, R. Sabourin, and E. Granger, "Decoupling direction and norm for efficient gradient-based L2 adversarial attacks and defenses," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2019.

[27] H. Zhang, Y. Avrithis, T. Furon, and L. Amsaleg, "Walking on the edge: Fast, low-distortion adversarial examples," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 701–713, 2021.

[28] B. Bonnet, T. Furon, and P. Bas, "What if adversarial samples were digital images?" in *ACM Workshop on Information Hiding and Multimedia Security*, 2020, pp. 55–66.

[29] F. Marra, G. Poggi, F. Roli, C. Sansone, and L. Verdoliva, "Counter-forensics in machine learning based forgery detection," in *Media Watermarking, Security, and Forensics*, A. M. Alattar, N. D. Memon, and C. D. Heitzenrater, Eds., vol. 9409, International Society for Optics and Photonics. SPIE, Mar. 2015, pp. 181–191.

[30] Z. Chen, B. Tondi, X. Li, R. Ni, Y. Zhao, and M. Barni, "A gradient-based pixel-domain attack against SVM detection of global image manipulations," in *IEEE Workshop on Information Forensics and Security*, Dec. 2017.

[31] B. Tondi, "Pixel-domain adversarial examples against CNN-based manipulation detectors," *Electronics Letters*, vol. 54, pp. 1220–1222, Oct. 2018.

[32] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80, Jul. 2018, pp. 274–283.

[33] D. Cozzolino, G. Poggi, and L. Verdoliva, "Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection," in *ACM Workshop on Information Hiding and Multimedia Security*, New York, NY, USA, 2017, pp. 159–164.

[34] M. Kirchner and S. Chakraborty, "A second look at first significant digit histogram restoration," in *IEEE International Workshop on Information Forensics and Security*, Dec. 2015.

[35] G. Valenzise, M. Tagliasacchi, and S. Tubaro, "Revealing the traces of JPEG compression anti-forensics," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 2, pp. 335–349, 2013.

[36] H. Zeng, T. Qin, X. Kang, and L. Liu, "Countering anti-forensics of median filtering," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2014, pp. 2704–2708.

[37] O. Mayer and M. C. Stamm, "Countering anti-forensics of lateral chromatic aberration," in *ACM Workshop on Information Hiding and Multimedia Security*, Jun. 2017, pp. 15–20.

[38] A. De Rosa, M. Fontani, M. Massai, A. Piva, and M. Barni, "Second-order statistics analysis to cope with contrast enhancement counter-forensics," *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1132–1136, 2015.

[39] M. Barni, Z. Chen, and B. Tondi, "Adversary-aware, data-driven detection of double JPEG compression: How to make counter-forensics harder," in *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec. 2016.

[40] Z. Chen, B. Tondi, X. Li, R. Ni, Y. Zhao, and M. Barni, "Secure detection of image manipulation by means of random feature selection," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 9, pp. 2454–2469, 2019.

[41] M. Barni, E. Nowroozi, B. Tondi, and B. Zhang, "Effectiveness of random deep feature selection for securing image manipulation detectors against adversarial examples," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2020, pp. 2977–2981.

[42] M. Fontani, A. Bonchi, A. Piva, and M. Barni, "Countering anti-forensics by means of data fusion," in *Media Watermarking, Security, and Forensics 2014*, vol. 9028, International Society for Optics and Photonics. SPIE, Feb. 2014, pp. 346–360.

[43] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[44] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "Evaluation of residual-based local features for camera model identification," in *New Trends in Image Analysis and Processing – ICIAP 2015 Workshops*, Sep. 2015, pp. 11–18.

[45] J. Rauber, R. Zimmermann, M. Bethge, and W. Brendel, "Foolbox native: Fast adversarial attacks to benchmark the robustness of machine learning models in PyTorch, TensorFlow, and JAX," *Journal of Open Source Software*, vol. 5, no. 53, p. 2607, 2020.

[46] W. Li, B. Tondi, R. Ni, and M. Barni, "Increased-confidence adversarial examples for deep learning counter-forensics," in *ICPR Workshops*. Springer International Publishing, 2021, pp. 411–424.

[47] M. Barni, W. Li, B. Tondi, and B. Zhang, *Adversarial Examples in Image Forensics*. Springer Singapore, 2022, pp. 435–466.

[48] Y. Yousfi, J. Butora, J. Fridrich, and C. Fuji Tsang, "Improving EfficientNet for JPEG steganalysis," in *ACM Workshop on Information Hiding and Multimedia Security*, Jun. 2021, pp. 149–157.

[49] S. Mandelli, N. Bonettini, and P. Bestagini, *Source Camera Model Identification*. Springer Singapore, 2022, pp. 133–173.

Benedikt Lorch received the M.Sc. degree in computer science from the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany, in 2018. In September 2018, he joined the IT Security Infrastructures Lab as a Ph.D. student. His research interests include security-related aspects of image processing, with particular focus on image forensics, machine learning, and computer vision.

Franziska Schirrmacher received the M.Sc. degree in medical engineering from the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany, in 2017. From 2017 to 2019, she was a researcher at the Pattern Recognition Lab at FAU. In 2019, she joined the IT Infrastructures Lab at FAU and is a member of the Multimedia Security Group. Her research interests include image processing, machine learning, and image forensics.

Anatol Maier received the M.Sc. degree in computer science from the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany, in 2019. Since November 2019, he is Ph.D. student at the IT Security Infrastructures Lab at FAU and a member of the Multimedia Security Group. His research interests include reliable machine learning, deep probabilistic models, and computer vision with particular application in image and video forensics.

Christian Riess received the Ph.D. degree in computer science from the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany, in 2012. From 2013 to 2015, he was a Postdoc at the Radiological Sciences Laboratory, Stanford University, Stanford, CA, USA. Since 2015, he is the head of the Phase-Contrast X-ray Group at the Pattern Recognition Laboratory at FAU. Since 2016, he is senior researcher and head of the Multimedia Security Group at the IT Infrastructures Lab at FAU. He is currently a member of the IEEE Information Forensics and Security Technical Committee. His research interests include all aspects of image processing and imaging, particularly with applications in image and video forensics, X-ray phase contrast, color image processing, and computer vision.