Attacking logo-based phishing website detectors with adversarial perturbations

Jehyun Lee[1], Zhe Xin[2], Melanie Ng Pei See[2], Kanav Sabharwal[2], Giovanni Apruzzese[3], and Dinil Mon Divakaran[2,4]

[1] Trustwave
[2] National University of Singapore
[3] Liechtenstein Business School, University of Liechtenstein
[4] Acronis Research

Abstract. Recent times have witnessed the rise of anti-phishing schemes powered by deep learning (DL). In particular, logo-based phishing detectors rely on DL models from Computer Vision to identify logos of well-known brands on webpages, to detect malicious webpages that imitate a given brand. For instance, Siamese networks have demonstrated notable performance for these tasks, enabling the corresponding anti-phishing solutions to detect even "zero-day" phishing webpages. In this work, we take the next step of studying the robustness of logo-based phishing detectors against adversarial ML attacks. We propose a novel attack leveraging generative adversarial perturbations to craft "adversarial logos" that, with no knowledge of phishing detection models, can successfully evade the detectors. We evaluate our attacks through: (i) experiments on datasets containing real logos, to evaluate the robustness of state-of-the-art phishing detectors; and (ii) user studies to gauge whether our adversarial logos can deceive human eyes. The results show that our proposed attack is capable of crafting perturbed logos subtle enough to evade various DL models—achieving an evasion rate of up to 95%. Moreover, users are not able to spot significant differences between generated adversarial logos and original ones.

Keywords: Phishing · Adversarial Machine Learning · Deep Learning

1 Introduction

Phishing attacks are on the rise [2], and they represent a serious threat to both organizations and individuals alike. While there have been numerous research efforts to counter this long-running security problem [25,56,30,31], a universal solution against phishing has yet to be found, as new ways to lure unaware victims keep emerging [3]. We focus on the problem of detecting phishing *websites*, which has witnessed 61% increase in 2022 [6].

The first line of defense against phishing websites is represented by blocklists, which are nowadays leveraged at scale [29]. Unfortunately, such rule-based countermeasures only work against the phishing entries in the blocklist, and attackers are well-aware of this (for a recent report, see [4]). To protect users against

evolving phishing websites, current anti-phishing schemes are now equipped with data-driven methods that detect malicious webpages by leveraging some heuristics [5]. In particular, the constant progress and successes of *machine learning* (ML) algorithms in research [51,57] led to the integration of ML-based phishing detectors also in popular browsers [33].

There are various ways in which ML is used to identify phishing websites, depending on the input analyzed by the ML model [22]: URL (e.g., [53,30]), HTML contents (e.g., [56,57,32]), or visual representations (e.g., [20,7]) of a webpage. Detection methods based on visual analytics are now receiving much attention (e.g., [20,19,7,34,28,35]), likely due to the tremendous advancements in deep learning (DL). In this work, we delve into the application of DL for *logo-based* phishing website detection—a state-of-the-art approach[5] that is *(i)* considered in recent researches (e.g., [19,28,34,35]), and *(ii)* deployed in practice [11].

In logo-based detection, the first task is to extract the logo(s) from a webpage (typically from its screenshot); the subsequent task is to identify the brand of the logo. The latter task can be accomplished by means of DL today, as demonstrated by recent works, e.g., by employing `Siamese` neural networks [34,35]. Given the relevance of these solutions in anti-phishing schemes, we scrutinize the robustness of DL models for logo identification against subtle adversarial perturbations. Even though many efforts in the DL community reveal the vulnerability of image classification models to adversarial examples [50,26,43,38], to the best of our knowledge, there exists no work that studies the vulnerability of logo-based phishing detectors against such sophisticated attacks. Therefore, besides the `Siamese` models proposed by prior work, we also develop two new logo-identification solutions based on state-of-the-art transformer models from Computer Vision—namely, Vision Transformer `ViT` [23] and `Swin` [36].

Subsequently, we propose a novel attack using *generative adversarial perturbations* (GAP) [43], to craft adversarial logos that simultaneously deceive *(i)* DL models for logo identification, and *(ii)* human users, i.e., potential victims. Through a comprehensive experimental study based on datasets of real logos, we demonstrate the quality of our proposed DL models for logo identification and the efficacy of the adversarial logos generated by our GAP attack to evade all three powerful models for logo identification (`Siamese`, `ViT` and `Swin`).

Finally, we carry out two user studies to assess the impact of our attack on real humans. We summarise our three major contributions:

1. We propose *a novel attack*, based on generative adversarial perturbations (GAP), against logo-based anti-phishing schemes (Section 4). Our proposed attack treats a phishing detection (specifically, logo-identification) model as a black-box and does not require any model-specific information.

[5] **Background:** in simple terms, logo-based phishing detection seeks to identify those (malicious) webpages that attempt to imitate a well-known brand. Intuitively, if a given webpage has the logo of a well-known brand (e.g., PayPal), but the domain does not correspond to the same brand (e.g., www.p4y-p4l.com), the webpage is classified as phishing. Though these approaches require maintenance of a database of logos for brands, such a task is not impractical given that the number of brands targeted by attackers is typically small (≈ 200) [7,18,34].

2. We propose *two new logo-identification solutions* leveraging transformer-based DL models: `ViT` and `Swin` (Section 3). We empirically demonstrate that both `ViT` and `Swin` achieve performance comparable to the state-of-the-art solutions relying on `Siamese` models [34,35] (Section 5.3).
3. Through a reproducible evaluation on real data, we *evaluate the robustness of three DL models for logo-identification* (`ViT`, `Swin`, `Siamese`) against our GAP-based attack (Section 5.4). We further validate the *impact of our attack on real humans* through a user study entailing ∼250 people (Section 6).

We suggest potential countermeasures against our attack, and also discuss ways that attackers can use to circumvent such countermeasures (Section 7). Finally, we publicly release our resources to the scientific community [1].

2 Threat model

We describe the threat model by first summarizing the functionality of the target system, and then presenting the characteristic of our envisioned attacker.

2.1 Target system: Logo-based phishing website detectors

Fig. 1 presents the general workflow of logo-based phishing detection systems. From a given webpage, the detection system first extracts the logo as an image; then, it identifies the brand the logo belongs to by using a discriminator. Such a discriminator can be implemented in various ways, e.g., earlier works employed methods based on SIFT (scale-invariant feature transformation) [9,54]; however, current state-of-the-art methods use DL models [34,16,35], and we focus on these. Upon identifying the brand of a logo, the system determines if the webpage is legitimate or not by comparing the webpage's domain with the domain of the brand associated with the logo.

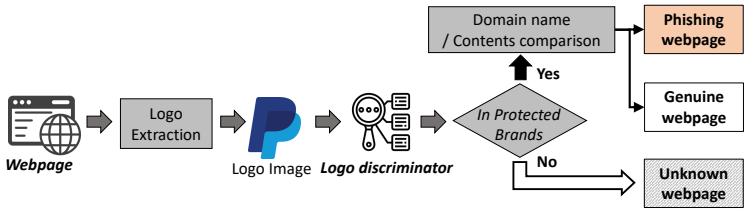


Fig. 1: Detection process of logo-based phishing detection systems

Since logo-identification is a multi-class classification problem, the DL model is trained on a static set of classes, i.e., the brands of the logos. Such a set of *protected brands* determines the size of the prediction classes; one brand may have multiple logos. Previous research has shown that 99% of the attacks target less than 200 brands [7,34,35].

In practice, phishing detectors must exhibit low false-positive rates (FPR), typically below 10^{-3} [31,44]. To successfully detect phishing webpages while maintaining low FPR, logo-based detectors follow two principles [34]: *(a)* the

highest predicted class is decided as the target brand *if and only if* the prediction probability is greater than a predefined *decision threshold* (say, θ); *(b)* if the identified logo does not belong to any brand in the protected set, the webpage is considered benign to avoid triggering false positives (see Fig. 1). Unfortunately, these principles can be maliciously exploited: by lowering the prediction probability, it is possible to evade logo-based phishing detectors.

2.2 Attack: Adversarial logos

The basic intuition behind our attack is to create an *adversarial logo* that is *(i)* minimally altered w.r.t. its original variant (to deceive the human eye); and that *(ii)* misleads the phishing detector. Let us describe our attacker by using the well-known notion of adversarial ML attacks [17,11].

- Goal: The attacker wants to craft an adversarial logo related to brand b which evades the phishing detector (at inference) while deceiving human eyes.
- Knowledge and Capabilities: To train a model for evasion, an attacker can collect authentic logos of any brand (e.g., of PayPal), via crawling or from public datasets (e.g., *Logo2K+* [55]). The attacker knows that their victims are protected by a logo-based phishing detector powered by ML. The attacker has a way to infer the decision result of the phishing detector (this is doable even if the detector is "invisible" [11], e.g., by inspecting visits to the hosted phishing webpage). The attacker does not i) require knowledge of the logo-identification model employed by the phishing detector, ii) manipulate the data used to train the ML model. In other words, it's neither a white-box attack nor performs data poisoning.
  Note, the attacker targets a set of brands for phishing; if the targeted brand is not within the protected set, then that is already favorable for an attacker—there is no perturbation required! Finally, the attacker naturally has control on their phishing webpages.
- Strategy: The attacker manipulates the logo(s) of brand b in their phishing webpages by introducing perturbations so that the logo-identification model predicts with lower confidence, i.e., the probability of the logo being of any brand is lower than the decision threshold (θ). This way, the phishing detector decides the logo *not* to be one of the protected brands, which makes way for successful evasion.

Scope of attack. In our threat model, the attacker exploits the vulnerability of *logo-identification* methods integrated into phishing detectors. We focus on logo-identification DL models because they are i) state-of-the-art research with phishing detecting capability in the wild ('zero-day' phishing) [34,35], and ii) used in commercial phishing detectors [11]. Threats against logo extraction from a webpage, however interesting, are not within the scope of our current work. Lastly, we do not consider attacks to make an unknown logo be identified as one of the protected logos, as that is not beneficial for the attacker.

3 Deep Learning for Logo-based Phishing Detection

Development of the transformer architecture [52] paved the way for various state-of-the-art language models, such as BERT, ChatGPT, and PaLM. Dosovitskiy et al. [23] applied transformer to Computer Vision tasks with the introduction of Vision Transformer (`ViT`), demonstrating state-of-the-art performance on benchmark datasets [23]. The attention mechanism in transformers allows them to capture local and global contextual information effectively, resulting in superior performance on large-scale image classification tasks. This capability is also beneficial for logo identification, since logos of the same brand, while being visually distinct, share the same inherent design structure. Therefore, in this work, we propose, develop and evaluate two transformer-based models, `ViT` and `Swin`, for logo identification. To the best of our knowledge, we are the first to leverage transformers for logo-based phishing detection.

We now describe our proposed `ViT` (Section 3.1) and `Swin` (Section 3.2), for which we provide an overview in Figs 2 and 3. Then, we present our own implementation of `Siamese` (Section 3.3) neural networks. Altogether, these three DL models will represent the target of our attacks (Section 5).

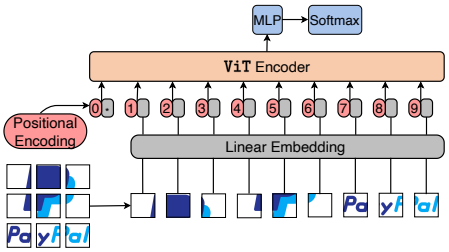


Fig. 2: `ViT`-based Model Architecture

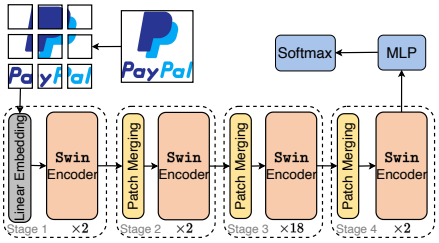


Fig. 3: `Swin`-based Model Architecture

3.1 `ViT` for logo identification

As illustrated in Fig. 2, we develop a logo-identification model by fine-tuning a pre-trained `ViT`-base model [23] on our dataset (which we discuss in Section 5.1). The model takes as input an image of size $3 \times 224 \times 224$. The image is then split into patches, each of size 16×16, for further processing. Each patch is then linearly embedded into a vector of size 1×768. An additional classification token is then added to the linear embedding to form an embedded vector of size 197×768. The embeddings are positionally encoded before being fed into the transformer encoder. Finally, a fully connected layer takes the output from the encoder and maps it to a 2-dimensional space. The resulting logits are passed through a softmax layer to produce the final prediction probabilities for each class (logo). We denote this new logo-identification model as $\mathcal{D}_{\texttt{ViT}}$.

3.2 `Swin` for logo identification

Next, we propose `Swin`-based logo-identification model that utilizes the `Swin` transformer, a hierarchical transformer architecture introduced by Liu et al. [36].

Unlike `ViT`, `Swin` uses shifted windows to efficiently compute local self-attentions and build hierarchical feature maps through patch merging techniques. As illustrated in Fig. 3, each window contains multiple non-overlapping patches, and each transformer block in the `Swin` architecture contains two attention layers: a window-based multi-head self-attention (W-MSA) layer that calculates local attention within a specific window, and a shifted window-based multi-head self-attention (SW-MSA) layer that introduces cross-window connections. This approach allows for more efficient computation while still extracting both local and global contextual information.

In our implementation, we use the `Swin-Transformer-Small` architecture proposed by Liu et al. [36]. The model takes an input image of size $3 \times 224 \times 224$, which is split into patches of size 4×4. As depicted in Figure 3, the patches are fed sequentially into four encoding stages consisting of 2, 2, 18, and 2 encoder blocks. Each encoding stage merges and downsamples the size of the feature maps by a factor of two, while doubling the number of channels.

The final feature map of size is 7×7 is transformed by a fully connected and softmax layer to obtain the output logits. We denote this model as $\mathcal{D}_{\text{Swin}}$.

3.3 `Siamese` and `Siamese`$^{++}$ for logo identification

The `Siamese` neural network is a state-of-the-art for image-based phishing detection, both for comparing screenshots [7] and logos [34,16,35]. In logo-based phishing detectors, `Siamese` models measure the similarity of a given logo to those in the protected set. We train a `Siamese` model as proposed in Phishpedia [34] and PhishIntention [35], utilizing a transfer learning approach. Specifically, we train a logo classification model with the `ResNetV2` network as the backbone, which effectively extracts different features from various logo variants. We then connect the trained `ResNetV2` network to a Global Average Pooling layer to output a vector for any given logo. The learned vector representation is compared to those of the logos of protected brands using cosine similarity; the target with the highest similarity is identified as the brand the logo is trying to imitate.

We refer to our implementation of the `Siamese` model as $\mathcal{D}_{\text{Siamese}}$. Additionally, Phishpedia [34] proposed an adversary-aware detector by replacing the ReLU activation function with a variant called step-ReLU (Appendix A). We also consider this robust version of `Siamese`, which we refer to as $\mathcal{D}_{\text{Siamese}^{++}}$.

4 Our Attack: Adversarial Logos

While recent logo-based phishing detection systems [34,35] have demonstrated robustness against generic gradient-based attacks such as FGSM [26] and DeepFool [39],[6] their resilience against more sophisticated adversarial attacks proposed in the literature [43,38] remains unexplored. To this end, we propose a

[6] FGSM and DeepFool assume an adversary with complete knowledge of the target classifier, which is much stronger (and less realistic [11]) than the attacker envisioned in our threat model.

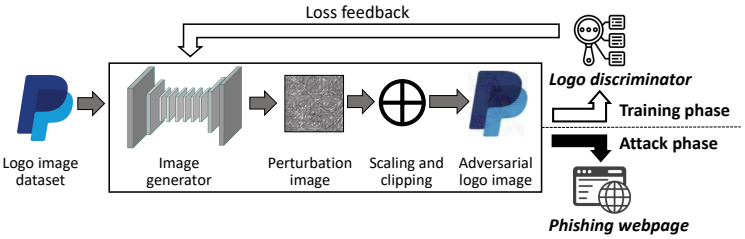


Fig. 4: Generative adversarial perturbation workflow

DL-based generative framework inspired by Generative Adversarial Perturbations (GAP) [43], that specifically trains against logo identification models. This framework generates perturbation vectors that can be added to a target logo image, allowing the perturbed logo to evade phishing detection while remaining imperceptible to the human eye. We now describe our framework at a high-level (Section 4.1), for which we provide an overview in Fig. 4; and then provide low-level details on how to practically implement our attacks (Section 4.2).

4.1 Framework: generative adversarial perturbations for logos

As illustrated in Fig. 4, our framework involves training a `Generator` that learns to generate perturbations. When added to a logo image, these perturbations can mislead a logo-identification model, which acts as the `Discriminator`, into lowering its prediction probability below the decision threshold. During the training process, the weights of the `Discriminator` are frozen, treating it as a black box to guide the training of the `Generator`.

Generator. We employ a Deep Residual Network with six residual blocks (ResNet-6) [27] as the core architecture of our `Generator`. Given a legitimate logo image as input, the `Generator` is trained to generate a *perturbation vector*. The generated perturbations undergo a *Scaling and Clipping* stage. In this stage, the perturbation vector is first scaled and normalized based on the L_∞ norm to control the magnitude of the perturbations, so that they remain imperceptible to human viewers. Subsequently, the normalized perturbations are added pixel-wise to the legitimate logo image, resulting in the adversarial logo.

Discriminator. The `Discriminator` is a pre-trained multi-class classifier designed to process a logo image and estimate the probability that the image belongs to a target brand in the protected set. In our framework, we select one of the logo-identification models described in Section 3 to serve as the `Discriminator`.

4.2 Implementation

We utilize the pre-trained `Discriminator` as a black box to assess the effectiveness of the `Generator` in crafting adversarial logo images. The `Discriminator` predicts the probability of a given logo belonging to each of the k protected brands; $\mathbf{V}_{\text{true}} : [p_1, p_2, p_3 ... p_k]$, where $\sum_{i=1}^{k} p_i = 1$. As mentioned in Section 2.1, for a webpage to be classified as phishing, the logo-identification model must

confidently identify the logo as one of the target brands i from the protected set, with a probability p_i greater than the phishing detector's decision threshold θ.

Hence, to devise our `Generator`, we introduce a target probability $p_{\text{adversarial}}$, such that $p_{\text{adversarial}} < \theta$. The `Generator` is trained to craft adversarial logos that are classified with probabilities lower than $p_{\text{adversarial}}$ for all of the protected brands, so as to evade phishing detection. Empirically, we observe that θ is very high (above 0.8) for all discriminators, and for our attacks, $p_{\text{adversarial}}$ can be much lower (in our experiments, it is 0.5; see Table 3 in Appendix B).

To guide the training process, the `Generator` is trained with a target probability vector $\mathbf{V}_{\text{target}} : [p'_1, p'_2, p'_3 .... p'_k]$, where each element p'_i is defined such that $p'_i = \min(p_i, p_{\text{adversarial}})$. This ensures that the generated adversarial logos are classified with probabilities below the θ for all protected brands.

The loss function is defined as a decreasing function of the cross entropy $\mathcal{H}(V_{\text{true}}, V_{\text{target}})$ between the target probability vector $\mathbf{V}_{\text{target}}$ and $\mathbf{V}_{\text{true}}$. The specific form of the loss function can be expressed as follows:

$$\text{loss} = \log\left(\mathcal{H}\left(\mathbf{V}_{\text{true}}, \mathbf{V}_{\text{target}}\right)\right) \tag{1}$$

Minimizing this loss, the `Generator` learns to craft adversarial logos that evade phishing detection[7]; furthermore, perturbations preserve the visual similarity with the original logo, thereby facilitating deception to the human eye.

5 Experimental evaluations

We now empirically assess the quality of our contributions. We begin by describing the datasets used for our experiments (Section 5.1), and introduce the metrics used for our performance assessment (Section 5.2). Then, we first show that our two DL models for logo-identification achieve state-of-the-art performance (Section 5.3), and then demonstrate that our attacks can evade all our considered logo-identification models (Section 5.4). Our code, dataset used, as well as generated perturbed logos are available at [1].

5.1 Dataset

To evaluate the performance of logo-based phishing detectors and their robustness against generative adversarial perturbations, we use two sets of logo images:

– **L**, **Protected brands:** The logo image set of protected brands, **L**, consists of images of 181 brands which are identical to the brands used in Phishpedia [34]. According to the empirical observation in [34], 99% of phishing

[7] **Remark:** Our attack relies on the logos generated by the `Generator`, which in turn depend on a `Discriminator`, i.e., a DL model for identifying logos. However, the `Discriminator` *does not* necessarily have to be the identical one used in the targeted phishing detection system: as our experiments show, our adversarial logos evade even DL models that have not been used to develop the `Generator` (by leveraging the well-known transferability property of adversarial examples [21]).

pages target one of these 181 brands. For these protected brands, we collected 28 263 public logo images from search engines and Pawar's logo image dataset [42]. Each brand's logo has 100–200 variants.
- $\bar{\mathbf{L}}$**, Unprotected brands:** Logo image set $\bar{\mathbf{L}}$ is the set of 2 045 images from 2 000 brands that do not belong to the brands in $\mathbf{L}$. The image samples are from the *Logo2K+* dataset, which is publicly available [55].

The data was collected in the second half of January 2023.

5.2 Performance Metrics

In what follows, we denote the logo-identification models as `discriminators`; the attack `generators` also use the discriminators in their training phase.

Logo identification performance: We provide the definitions of metrics for logo-based phishing webpage detection. Note that, for a discriminator used for phishing detection, the positives are the logos in $\mathbf{L}$, the protected brand list, that need to be identified. If the highest prediction probability of a logo is below a certain decision threshold, it is classified as an unknown brand.

- *True positive (TP)*: A TP in our evaluation denotes the case of correct brand identification of the given logo (of a protected brand) by the discriminator.
- *False positive (FP)*: An FP denotes the case when the given logo image is wrongly identified as one of the protected brands when in reality, the given logo image does not belong to the protected brand set.
- *True negative (TN)*: A TN occurs when the brand of the given logo is not in the protected brand set and gets correctly classified as an unknown brand.
- *False negative (FN)*: An FN denotes when the brand of the given logo belonging to the protected brand set is classified as any other brand.

Denoting the actual brand of a given logo l as l_b, and the predicted brand by the discriminator as l_p, we define the True Positive Rate (TPR) and False Positive Rate (FPR):

$$\text{TPR} = \frac{|(l_b = l_p) \wedge (l_p \in \mathbf{L})|}{|l_b \in \mathbf{L}|}; \qquad \text{FPR} = \frac{|(l_p \in \mathbf{L}) \wedge (l_b \in \bar{\mathbf{L}})|}{|l_b \in \bar{\mathbf{L}}|} \tag{2}$$

Impact of the attacks: Recall that our attacker aims to fool the discriminator into classifying a protected brand logo as an unknown brand. Hence, we introduce the *Fooling ratio*, which is the rate of adversarial logos classified as being of an unknown brand (out of all the phishing logos). Formally:

$$\text{Fooling ratio} = \frac{|l_p \notin \mathbf{L} \wedge l_b \in \mathbf{L}|}{|l_b \in \mathbf{L}|} \tag{3}$$

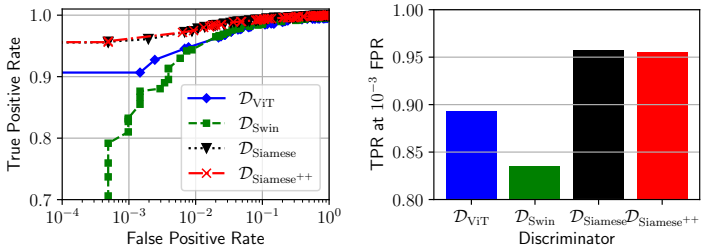
Intuitively, a higher fooling ratio denotes an attack with a higher impact.

5.3 Baseline: Analysis of logo-identification models

We assess the performance of the four DL models for logo-identification presented in Section 3. Specifically, we first measure the TPR and FPR of the state-of-the-art discriminators (i.e., `Siamese` and its robust version `Siamese`$^{++}$ [34]), and compare them with the transformer-based discriminators that we proposed in this work (i.e., `ViT` and `Swin`).

Setup. We use the datasets $\mathbf{L}$ and $\bar{\mathbf{L}}$ (see Section 5.1), with a train:test split of 85:15. For `ViT` and `Swin`, we apply the common model head fine-tuning for 50 epochs and then transfer training on the entire networks for the next 150 epochs, reducing computational time while improving performance. We provide hyper-parameters configurations of our discriminators in Table 2 (in the appendix).

Results. Fig. 5a shows the ROC curves of the four discriminators (the x-axis denoting FPR is in log-scale for visibility). Overall, `Siamese` and `Siamese`$^{++}$ show the best performance in terms of logo identification. All four models show comparable TPRs at FPR above 10^{-2}. For practical purposes, however, we have to evaluate the detection capability at low FPRs [44,22]. Observe that, the TPR values of the discriminators `ViT` and `Swin` at FPR below 10^{-2} are worse than the `Siamese` models. Fig. 5b shows the gap in TPR between the discriminators at the more practical FPR value of 10^{-3}; `Siamese` and `Siamese`$^{++}$ show around six and twelve percent-point higher TPR than the `ViT` and `Swin`, respectively.



(a) ROC curves **(b)** TPR at 10^{-3} FPR

Fig. 5: Comparing discriminators for logo identification

Although `Swin` and `ViT` are not better than `Siamese`, they still achieve an appreciable degree of performance, and hence are used to evaluate our attacks.

5.4 Attack: evasiveness of adversarial logos, and computational cost

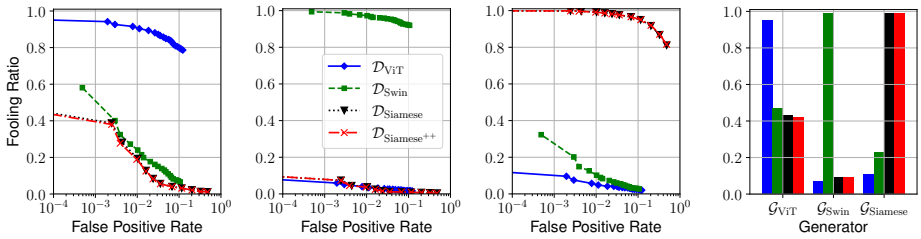
We quantitatively analyze the effects of adversarial logos generated by our attack against DL models for logo identification. We do this through a cross-evaluation that captures both 'white-box' and 'black-box' adversarial settings. At the end of this section, we also discuss the computational cost of our attacks.

Setup. Recall that our attack (Section 4) entails training a generator by using a given discriminator (i.e., DL models for identifying logos). For our experiments, we consider three discriminators: `ViT`, `Swin` and `Siamese`, thereby yielding three corresponding generators: $\mathcal{G}_{\text{ViT}}$, $\mathcal{G}_{\text{Swin}}$ and $\mathcal{G}_{\text{Siamese}}$. After training each generator, we assess the adversarial logos against *all our discriminators*. Such an

evaluation protocol allows one to analyze the effects of our attacks when the adversary does not know the DL model used for the defense.

For evaluations, we train our generators on the dataset **L**; we provide the hyperparameters of our generators in Table 3 (Appendix B). Subsequently, we test the discriminators with the adversarial logos crafted by each generator.

Results. The results are plotted in Fig. 6, where we compare the fooling ratio of discriminators against the different attacker models for varying FPRs (in log-scale). It stands out that each discriminator is much weaker against the adversarial logos created by the 'matching' generator compared to those created by generators trained on different discriminators. For instance, from Fig. 6a, we observe that the adversarial logos generated by $\mathcal{G}_{\text{ViT}}$ are more effective against `ViT` (blue line) than against `Swin` (green line). We observe from Fig. 6b and Fig. 6c that, if the attacker's generator model is not trained with `ViT`, the fooling ratio drops significantly for the defender with the `ViT` discriminator.



(a) $\mathcal{G}_{\text{ViT}}$ **(b)** $\mathcal{G}_{\text{Swin}}$ **(c)** $\mathcal{G}_{\text{Siamese}}$ **(d)** at 10^{-3} FPR

Fig. 6: Comparison of different generators against different discriminators

From the adversary's perspective, `ViT` is the most effective generator against all discriminators. Fig. 6d compares the fooling ratios of the four discriminators at a fixed FPR of 10^{-3}; note, **fooling ratios against $\mathcal{G}_{\text{ViT}}$ are high, ranging from 42% to 95%**. In other words, with $\mathcal{G}_{\text{ViT}}$, at least 42% of attacker generated logos can evade phishing detectors, independent of the discriminator deployed. Against such an attacker, the defender might prefer to use `Siamese` (or `Siamese`$^{++}$) as it achieves the lowest fooling ratio (of around 42% at 10^{-3} FPR). Interestingly, the most robust model for the defender against an *arbitrary* generator model would be `ViT`, since, on average, `ViT` achieves a lower fooling ratio against all generator models.

Computational cost. Two factors contribute to the computation time to realize our adversarial logos: i) generator training and ii) perturbed logo generation. We measure the generator training time with the three models, i.e., `ViT`, `Swin`, and `Siamese`, for each training epoch and the required epochs till reaching a compelling performance, i.e., 0.9 of fooling ratio against the discriminator with the corresponding model. The experiments are performed on a system with NVIDIA RTX3090 GPU, 2.8GHz 32-core AMD CPU, 80GB RAM with Python 3.8.10, and PyTorch 1.2.0 on Ubuntu 20.04 OS. We report the results in Table 1.

From this table, we observe an apparent gap between the models in their training time. While the `ViT`-based generator, $\mathcal{G}_{\text{ViT}}$, takes only half the training

Table 1: Training time for the perturbation generators

	$\mathcal{G}_{\text{ViT}}$	$\mathcal{G}_{\text{Swin}}$	$\mathcal{G}_{\text{Siamese}}$
Avg. training time per epoch (min.)	12	23	8
No. of epochs for 0.9 fooling ratio	62	12	1
Training time for 0.9 fooling ratio (min.)	744	277	8

time per epoch in comparison to $\mathcal{G}_{\text{Swin}}$, it requires five times more training epochs to reach the same level of performance, (i.e., 0.9 fooling ratio). $\mathcal{G}_{\text{Siamese}}$ shows significantly less overhead than the other two, in both, training time per epoch and the required epoch. $\mathcal{G}_{\text{Siamese}}$ accomplishes a fooling ratio of 0.9 against $\mathcal{D}_{\text{Siamese}}$ after just one epoch of training which takes only eight minutes. Overall, training $\mathcal{G}_{\text{ViT}}$ takes 744 minutes to have 0.9 fooling ratio, which is around 2.8 and 93 times longer training time than $\mathcal{G}_{\text{Swin}}$ and $\mathcal{G}_{\text{Siamese}}$, respectively. Although there are significant differences in training times, when it comes to generating perturbed logos, all three generators take only around 0.7 seconds per image on average; this negligible cost allows an attacker to generate a large number of samples to test against a deployed phishing detector.

Takeaways. i) An attacker with knowledge of the discriminator used for defense achieves more than 95% fooling ratio with our adversarial generator. ii) In the absence of knowledge of the discriminator (i.e., independent of the discriminator), an attacker choosing $\mathcal{G}_{\text{ViT}}$ as the generator achieves a fooling ratio of at least 42% against the defender (see Fig. 6d).

6 User study: do adversarial logos trick humans?

We now provide a complementary evaluation of our proposed attack. Specifically, we seek to investigate *if our adversarial logos can be spotted by humans*. Indeed, even if a phishing detector can be evaded, this would be useless if the human, the actual target of the phishing attack, can clearly see that something is "phishy". Hence, we carry out **two user-studies**, which we describe (Section 6.1) and discuss (Section 6.2) in the remainder of this section.

6.1 Methodology

Our goal is to assess if the perturbations entailed in an adversarial logo can be recognized by humans. There are many ways to perform such an assessment through a user-study, each with its own pros and cons[8].

We build our user-studies around a central research question (RQ): *given a pair of logos (i.e., an 'original' one, and an 'adversarial' one), can the human spot any difference?* Our idea is to design a questionnaire containing multiple pairs of logos, and ask the participants to rate (through a 1–5 Linkert scale) the similarity of the logos in each pair. Intuitively, if the results reveal that users

[8] Designing bias-free user-studies in the phishing context is an open problem [48,10].

perceive the logos to be "different", then it would mean that our adversarial logos are not effective against humans.

To account for the fact that the responses we would receive are entirely subjective, we carry out (in April 2023) two quantitative user studies:

1. *Vertical Study* (VS), which entails a small population (N=30) of similar users (students of a large university, aged 20–30). The questionnaire has ten questions (each being a pair of logos to rate), wherein each participant is shown a different set of questions. The purpose of VS is to capture the responses of a specific group of humans across a large set of adversarial logos.
2. *Horizontal Study* (HS), which entails a large population (N=287) of users with diverse backgrounds (Amazon Turk Workers with 95+% hit-rate, aged 18–70). The questionnaire includes 21 questions, which are always the same for each participant. The purpose of HS is to capture the response of various humans to a small set of adversarial logos.

For both VS and HS, participants were asked to provide a response within 5s of seeing the pair of logos (because, realistically, users do not spend much time looking at the logo on a website). We also included control questions (e.g., pairs of identical logos, and pairs of clearly different logos) as a form of attention mechanism[9]. Finally, we shuffled the questions to further reduce bias. For transparency, we provide our questionnaire at [1].

For VS (resp. HS), we included 2 (resp. 3) "identical" pairs as baseline; and 5 (resp. 12) "original-adversarial" pairs to answer our RQ.

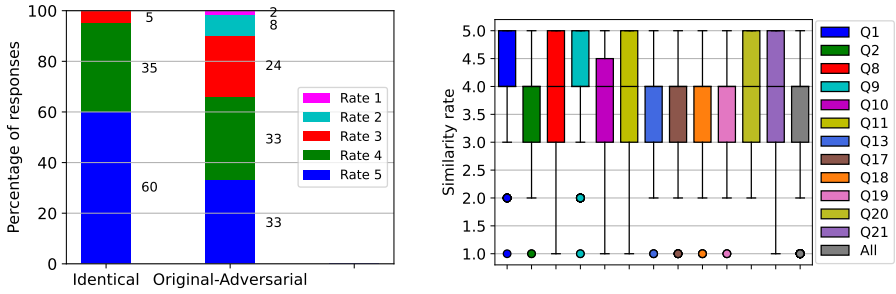
6.2 Results

We present the results of both of our user studies in Fig. 7. Specifically, Fig. 7a shows the cumulative distribution of the scores for the three 'identical' pairs, and the five 'original-adversarial' pairs in VS. Whereas the boxplots in Fig. 7b show how the participants of HS rated the 12 "original-adversarial" pairs; the rightmost boxplot aggregates all results. In our rating definition, 5 means 'similar', and 1 means 'different'.

From Figure 7a, we observe that 95% of all responses (30 users $\times$ 10 questions) rated all 'identical' pairs (left bin) between 4 and 5 (only 5% answered with a 3). That is to say; they correctly guessed that all identical pairs were indeed very similar, thereby also confirming that this population was very reliable. For this reason, we find it noteworthy that **our adversarial logos are able to deceive them**: in the right bin, 66% rated the 'original-adversarial' pairs with either 4 or 5, and only 10% rated them with a 1 or 2.

Figure 7b shows the results for the 'adversarial-original' pairs (we already removed some clearly noisy answers, as stated in Section 6.1). We observe that the wide majority of HS population rated the pairs as similar (the average is always below the middle point, 3). Hence, we can conclude: HS also reveals that **our adversarial logos are barely detected by humans as perturbed**.

[9] For HS, we received 322 responses, but we removed 35 because some users took too little time to answer the entire questionnaire, or did not pass our attention checks.



(a) Vertical Study **(b)** Horizontal Study

Fig. 7: Results of our two user-studies: vertical study and horizontal study

7 Countermeasures (and counter-countermeasures)

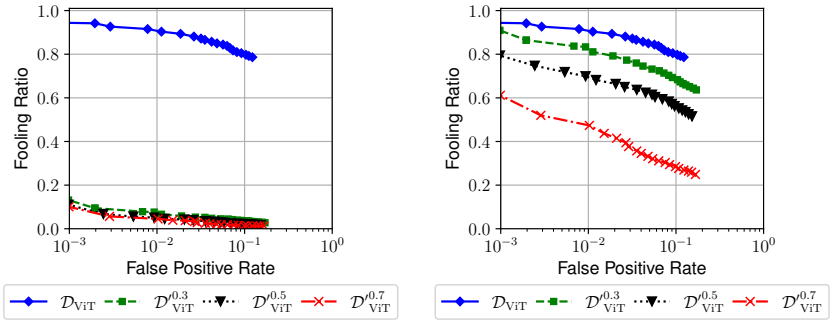
Given that our adversarial logos can simultaneously fool state-of-the-art DL models for logo-identification and human eyes, we ask ourselves: *how can adversarial logos be countered?* One potential mitigation is to leverage *adversarial learning* by injecting evasive logos in the training set [12], thereby realizing an *adversarially robust* discriminator. However, an expert attacker may anticipate this and can hence attempt to circumvent such a robust discriminator by developing a new generator, thereby crafting more evasive adversarial logos (e.g., as demonstrated in other domains [49,45]). We now investigate both of these scenarios through additional proof-of-concept experiments, which involve the strongest discriminator of our evaluation: `ViT`.

Countermeasure: building robust discriminator. Adversarial training is one of the most well-known techniques to defend against adversarial examples [46,12]. The idea is to update a given ML model by training it on adversarial examples that can mislead its predictions. We build our robust discriminators, $\mathcal{D}'^{0.3}_{\text{ViT}}$, $\mathcal{D}'^{0.5}_{\text{ViT}}$, and $\mathcal{D}'^{0.7}_{\text{ViT}}$, by replacing 30%, 50%, and 70% of the logos in the training dataset $\mathbf{L}$ with their adversarial variants, respectively. In particular, we use the adversarial logos generated with $\mathcal{G}_{\text{ViT}}$, i.e., trained with the vanilla `ViT` discriminator. Then, we compare these three robust discriminators with the vanilla `ViT` discriminator $\mathcal{D}_{\text{ViT}}$, against the same attack presented in Section 5.4. The results are shown in Fig. 8a. We observe that the robust discriminators exhibit much lower fooling ratios: while the vanilla `ViT` has a fooling ratio above 0.8, the robust discriminators have fooling ratios below 0.2 even at a low FPR of 10^{-3}.

Counter-countermeasure: evading robust discriminators. An attacker is also capable of taking a sophisticated strategy to counter a robust logo-identification discriminator built via adversarial training. To do this, the attacker must obtain such a robust discriminator—this can be done through well-known black-box strategies [41,15], or the attacker could even build one on their own. The attacker must then use the robust discriminator to train an 'adaptive' generator that can yield more evasive perturbations. For this experiment, we consider the case wherein the attacker trains the adaptive generator by using $\mathcal{D}'^{0.3}_{\text{ViT}}$, $\mathcal{D}'^{0.5}_{\text{ViT}}$, and

$\mathcal{D}'^{0.7}_{\text{ViT}}$, thereby realizing $\mathcal{G}'^{0.3}_{\text{ViT}}$, $\mathcal{G}'^{0.5}_{\text{ViT}}$, and $\mathcal{G}'^{0.7}_{\text{ViT}}$, respectively. The results are shown in Fig. 8b, which plots the fooling ratio of the *adaptive* generator against the corresponding *robust* discriminator.

Compared to the attacks from the 'vanilla' generator $\mathcal{G}_{ViT}$ in Fig. 8a (which achieves below 20% of fooling ratio at 10^{-3} FPR), the adaptive generators in Fig. 8b are much more effective. Yet, we observe that discriminators trained with more adversarial logos tend to be more robust: at 10^{-3} FPR, $\mathcal{D}'^{0.3}_{\text{ViT}}$ has a fooling ratio of 0.9, whereas $\mathcal{D}'^{0.5}_{\text{ViT}}$ and $\mathcal{D}'^{0.7}_{\text{ViT}}$ have 0.8 and 0.6, respectively.



(a) Against original generator $\mathcal{G}_{\text{ViT}}$ **(b)** Against adaptive generators

Fig. 8: Performance of discriminator and generator due to adversarial training

We find it enticing that this continuous game between attacker and defender, reflected in the generator (attacker) and discriminator (defender), eventually forms the concept of the Generative Adversarial Network (GAN). Indeed, a question rises: "what happens if this process is repeated many times?" We plan to address this intriguing research question in our future work.

8 Related works

Phishing Website Detection via ML. Many works leveraged statistical models, including ML, for phishing website detection (e.g., [8,56,57,37,51]). Typically, these models are trained on labeled datasets to learn to discriminate between phishing and benign webpages. There also exists an orthogonal family of countermeasures, referred to as reference-based phishing detectors, that identify visually similar webpages. This is based on the notion that phishing webpages are more successful when they imitate a legitimate website. This characteristic has been extensively scrutinized by prior literature [24,9,54,19,7,28,34,35]. For example, VisualPhishNet trains a `Siamese` model to detect visually similar *screenshots* between a given webpage and those in a set of well-known brands [7]. Other works (e.g., [9,54,19,34,35]) focus on identifying visually invariant *logos*.

Attacks against ML-based Phishing Website Detectors. Expert attackers are aware of the development of anti-phishing solutions and constantly refine their techniques to avoid being taken down. For instance, phishers can use cloaking to evade automated crawlers often used by security vendors [59]; alternatively,

they can exploit 'squatting' to evade detectors analyzing the URL [51]. It is also easy to change the HTML contents to evade HTML-based phishing detectors [32,13]. Researchers have also examined the impact of adversarial perturbations on image-based phishing detectors [7,34,35,20]. However, these attacks assume that the attacker possesses complete knowledge of the deployed model and can access the model gradients, enabling manipulations in the feature-space (for further details, refer to [13]). We demonstrate a successful attack conducted by an attacker lacking both knowledge of and access to the deployed model. Furthermore, none of the prior works have conducted user studies to validate the practicality of their attacks.

Adversarial Perturbations. Moving away from gradient-based perturbations, Moosavi et al. introduced Universal Adversarial Perturbations [38], a framework for learning perturbations that are image-agnostic and generalized across various image classification models. This work sparked further proposals [47,40,58] aiming to enhance universal perturbations. Subsequently, Poursaeed et al. proposed Generative Adversarial Perturbations [43]. The generative model achieved state-of-the-art performance, unifying the framework for image-agnostic and image-dependent perturbations and considering both targeted and non-targeted attacks. We draw inspiration from their framework to develop a generative network specifically for crafting adversarial logos.

Summary. While prior works have investigated gradient-based attacks [34,35] against image classifiers, to the best of our knowledge, we are the first to show the feasibility of attacks using a generative neural network model trained to craft adversarial logos, and comprehensively evaluate the impact of such attacks on state-of-the-art methods for logo-identification.

9 Conclusions

Logo-based phishing detectors have shown significant capabilities with the employment of DL models. In this work, we developed and presented a novel attack against logo-based phishing detection systems. Our experiments demonstrate the capability of an attacker equipped with a generative adversarial model in defeating the detection systems as well as human users. We hope this will trigger further research and development of phishing detection solutions that are robust to adversarial ML attacks.

Ethical Statement. Our institutions do not require any formal IRB approval to carry out the research discussed herein. We always followed the guidelines of the Menlo report [14]. For our user-studies, we never asked for sensitive data or PII. Finally, although we publicly release our code for the sake of science, as mentioned on the GitHub page [1], such code should not be used for any unethical or illegal purposes.

Acknowledgment. We thank the Hilti Corporation, Trustwave, NUS (National University of Singapore) and Acronis, for supporting this research.

References

1. Adversarial logos against phishing detection systems: Code repository, https://github.com/JehLeeKR/Adversarial-phishing-logos
2. APWG: Phishing activity trends report, 4th quarter 2022. https://docs.apwg.org//reports/apwg_trends_report_q4_2022.pdf
3. Browser In The Browser (BITB) Attack. https://mrd0x.com/browser-in-the-browser-phishing-attack/ (2022)
4. COFENSE: Phishing URLs 4x more likely than attachments to reach users. https://cofense.com/blog/urls-4x-more-likely-than-phishing-attachments-to-reach-users/ (2023)
5. Google Safe Browsing. https://developers.google.com/safe-browsing/ (2023)
6. Phishing attacks jump 61% in 2022. https://venturebeat.com/security/report-phishing-attacks-jump-61-in-2022-with-255m-attacks-detected/ (2023)
7. Abdelnabi, S., Krombholz, K., Fritz, M.: Visualphishnet: Zero-day phishing website detection by visual similarity. In: Proc. ACM CCS. pp. 1681–1698 (2020)
8. Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S.: A Comparison of Machine Learning Techniques for Phishing Detection. In: Proc. of the Anti-Phishing Working Groups, 2nd Annual eCrime Researchers Summit. eCrime '07 (2007)
9. Afroz, S., Greenstadt, R.: Phishzoo: Detecting phishing websites by looking at them. In: IEEE International Conf. on Semantic Computing (2011)
10. Alsharnouby, M., Alaca, F., Chiasson, S.: Why phishing still works: User strategies for combating phishing attacks. International Journal of Human-Computer Studies (2015)
11. Apruzzese, G., Anderson, H., Dambra, S., Freeman, D., Pierazzi, F., Roundy, K.: Position:"Real Attackers Don't Compute Gradients": Bridging the Gap Between Adversarial ML Research and Practice. In: IEEE Conference on Secure and Trustworthy Machine Learning (2023)
12. Apruzzese, G., Andreolini, M., Marchetti, M., Venturi, A., Colajanni, M.: Deep reinforcement adversarial learning against botnet evasion attacks. IEEE Transactions on Network and Service Management (2020)
13. Apruzzese, G., Conti, M., Yuan, Y.: Spacephish: The evasion-space of adversarial attacks against phishing website detectors using machine learning. In: Proc. ACSAC (2022)
14. Bailey, M., Dittrich, D., Kenneally, E., Maughan, D.: The Menlo Report. IEEE S&P (2012)
15. Bhagoji, A.N., He, W., Li, B., Song, D.: Practical black-box attacks on deep neural networks using efficient query mechanisms. In: Proc. ECCV. pp. 154–169 (2018)
16. Bhurtel, M., Siwakoti, Y.R., Rawat, D.B.: Phishing Attack Detection with ML-Based Siamese Empowered ORB Logo Recognition and IP Mapper. In: Proc. IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (2022)
17. Biggio, B., Roli, F.: Wild patterns: Ten years after the rise of adversarial machine learning. Pattern Recogn. (2018)
18. Bitaab, M., Cho, H., Oest, A., Zhang, P., Sun, Z., Pourmohamad, R., Kim, D., Bao, T., Wang, R., Shoshitaishvili, Y., et al.: Scam Pandemic: How Attackers Exploit Public Fear through Phishing. In: Proc. APWG Symposium on Electronic Crime Research (eCrime). pp. 1–10. IEEE (2020)

19. Bozkir, A.S., Aydos, M.: LogoSENSE: A Companion HOG based Logo Detection Scheme for Phishing Web Page and E-mail Brand Recognition. Computers & Security (2020)
20. Corona, I., Biggio, B., Contini, M., Piras, L., Corda, R., Mereu, M., Mureddu, G., Ariu, D., Roli, F.: Deltaphish: Detecting phishing webpages in compromised websites. In: Proc. ESORICS (2017)
21. Demontis, A., Melis, M., Pintor, M., Jagielski, M., Biggio, B., Oprea, A., Nita-Rotaru, C., Roli, F.: Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In: USENIX Security Symp. (2019)
22. Divakaran, D.M., Oest, A.: Phishing Detection Leveraging Machine Learning and Deep Learning: A Review. IEEE Security & Privacy **20**(5), 86–95 (2022)
23. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
24. Fu, A.Y., Wenyin, L., Deng, X.: Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover's Distance (EMD). IEEE Transactions on Dependable and Secure Computing **3**(4), 301–311 (2006)
25. Garera, S., Provos, N., Chew, M., Rubin, A.D.: A framework for detection and measurement of phishing attacks. In: Proc. ACM workshop on Recurring malcode (2007)
26. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples. In: International Conf. on Learning Representations (Poster) (2015)
27. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. IEEE CVPR. pp. 770–778 (2016)
28. Hout, T.v.d., Wabeke, T., Moura, G.C.M., Hesselman, C.: LogoMotive: detecting logos on websites to identify online scams - a TLD case study. In: Proc. PAM (2022)
29. Kondracki, B., Azad, B.A., Starov, O., Nikiforakis, N.: Catching transparent phish: analyzing and detecting mitm phishing toolkits. In: Proc. ACM CCS (2021)
30. Le, H., Pham, Q., Sahoo, D., Hoi, S.C.: URLNet: Learning a URL representation with deep learning for malicious URL detection. arXiv preprint arXiv:1802.03162 (2018), https://arxiv.org/abs/1802.03162
31. Lee, J., Tang, F., Ye, P., Abbasi, F., Hay, P., Divakaran, D.M.: D-Fence: A Flexible, Efficient, and Comprehensive Phishing Email Detection System. In: Proc. IEEE EuroS&P (2021)
32. Lee, J., Ye, P., Liu, R., Divakaran, D.M., Choon, C.M.: Building robust phishing detection system: an empirical analysis. In: Proc. NDSS MADWeb (2020)
33. Liang, B., Su, M., You, W., Shi, W., Yang, G.: Cracking classifiers for evasion: a case study on the google's phishing pages filter. In: Proc. WWW (2016)
34. Lin, Y., Liu, R., Divakaran, D.M., Ng, J.Y., Chan, Q.Z., Lu, Y., Si, Y., Zhang, F., Dong, J.S.: Phishpedia: A Hybrid Deep Learning Based Approach to Visually Identify Phishing Webpages. In: Proc. USENIX Security Symposium (2021)
35. Liu, R., Lin, Y., Yang, X., Ng, S.H., Divakaran, D.M., Dong, J.S.: Inferring Phishing Intention via Webpage Appearance and Dynamics: A Deep Vision Based Approach. In: Proc. USENIX Security Symposium (2022)
36. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proc. the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
37. Ma, J., Saul, L.K., Savage, S., Voelker, G.M.: Identifying Suspicious URLs: An Application of Large-Scale Online Learning. In: Proc. ICML (2009)

38. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: Proc. IEEE CVPR. pp. 1765–1773 (2017)
39. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proc. IEEE CVPR. pp. 2574–2582 (2016)
40. Mopuri, K.R., Ganeshan, A., Babu, R.V.: Generalizable data-free objective for crafting universal adversarial perturbations. IEEE transactions on pattern analysis and machine intelligence **41**(10), 2452–2465 (2018)
41. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proc. ACM ASIACCS (2017)
42. Pawar, R.: Logo images dataset. https://github.com/revanks/logo-images-dataset (2021), gitHub repository
43. Poursaeed, O., Katsman, I., Gao, B., Belongie, S.: Generative adversarial perturbations. In: Proc. IEEE CVPR. pp. 4422–4431 (2018)
44. Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., Rieck, K.: Do's and don'ts of machine learning in computer security. In: Proc. USENIX Security Symposium (2022)
45. Rahman, M.S., Imani, M., Mathews, N., Wright, M.: Mockingbird: Defending against deep-learning-based website fingerprinting attacks with adversarial traces. IEEE Transactions on Information Forensics and Security **16**, 1594–1609 (2020)
46. Shafahi, A., Najibi, M., Ghiasi, M.A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G., Goldstein, T.: Adversarial training for free! Advances in Neural Information Processing Systems (2019)
47. Shafahi, A., Najibi, M., Xu, Z., Dickerson, J., Davis, L.S., Goldstein, T.: Universal adversarial training. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 5636–5643 (2020)
48. Sharma, K., Zhan, X., Nah, F.F.H., Siau, K., Cheng, M.X.: Impact of digital nudging on information security behavior: an experimental study on framing and priming in cybersecurity. Organizational Cybersecurity Journal: Practice, Process and People **1**(1), 69–91 (2021)
49. Shenoi, A., Vairam, P.K., Sabharwal, K., Li, J., Divakaran, D.M.: iPET: Privacy Enhancing Traffic Perturbations for Secure IoT Communications. Proceedings on Privacy Enhancing Technologies **2**, 206–220 (2023)
50. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing Properties of Neural Networks. CoRR (2014)
51. Tian, K., Jan, S.T., Hu, H., Yao, D., Wang, G.: Needle in a haystack: Tracking down elite phishing domains in the wild. In: Internet Measurement Conference (2018)
52. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
53. Verma, R., Dyer, K.: On the character of phishing URLs: Accurate and robust statistical learning classifiers. In: Proc. ACM Conf. Data Appl. Secur. Privacy (2015)
54. Wang, G., Liu, H., Becerra, S., Wang, K., Belongie, S.J., Shacham, H., Savage, S.: Verilogo: Proactive phishing detection via logo recognition. Department of Computer Science and Engineering, University of California, San Diego (2011)
55. Wang, J., Min, W., Hou, S., Ma, S., Zheng, Y., Wang, H., Jiang, S.: Logo-2K+: A large-scale logo dataset for scalable logo classification. In: Proc. AAAI. pp. 6194–6201 (2020)
56. Whittaker, C., Ryner, B., Nazif, M.: Large-Scale Automatic Classification of Phishing Pages. In: Proc. NDSS (2010)

57. Xiang, G., Hong, J., Rose, C.P., Cranor, L.: CANTINA+: A Feature-Rich Machine Learning Framework for Detecting Phishing Web Sites. ACM Trans. on Information and System Security **14**(2) (2011)
58. Zhang, C., Benz, P., Imtiaz, T., Kweon, I.S.: CD-UAP: Class discriminative universal adversarial perturbation. In: Proc. AAAI Conference on Artificial Intelligence. vol. 34, pp. 6754–6761 (2020)
59. Zhang, P., Oest, A., Cho, H., Sun, Z., Johnson, R., Wardman, B., Sarker, S., Kapravelos, A., Bao, T., Wang, R., et al.: CrawlPhish: Large-scale Analysis of Client-side Cloaking Techniques in Phishing. In: Proc. IEEE S&P (2021)

Appendix

A Step-ReLu activation Function

The step-ReLU function utilised in training the robust `Siamese` model $\mathcal{D}_{\text{Siamese}^{++}}$ (Section 3.3) is expressed as:

$$f(x) = \max(0, \alpha \cdot \lceil \frac{x}{\alpha} \rceil) \tag{4}$$

B Discriminator and generator configurations

Table 2: Hyperparameter configurations for discriminators

Parameters	$\mathcal{D}_{\text{ViT}}$	$\mathcal{D}_{\text{Swin}}$	$\mathcal{D}_{\text{Siamese}}$
Backbone	ViT	Swin	ResNetV2
Pre-trained Model	ViT-b/16	Swin-S	BiT-M-R50x1
No. of params	85.9M	49.0M	23.9M
Batch size	32	32	32
Optimizer	SGD	SGD	SGD
Momentum	0.9	0.9	0.9
Weight decay	0.0005	0.0005	-
Epochs (Steps)	200	200	10000 (Steps)
Learning rate	0.01	0.01	0.003 (Staircase decay)
λ (value clipping)	2.5	2.5	-

Table 3: Hyperparameter configurations for generators

Parameters	$\mathcal{G}_{\text{ViT}}$	$\mathcal{G}_{\text{Swin}}$	$\mathcal{G}_{\text{Siamese}}$
Batch size	32	16	32
Optimizer	Adam	Adam	Adam
β_1 & β_2 for Adam	0.5 & 0.999	0.5 & 0.999	0.5 & 0.999
Magnitude of perturbations	10	10	10
Epochs	200	200	100
Learning rate	0.0002	0.0002	0.0002
Target probability, $p_{\text{adversarial}}$	0.5	0.5	0.5