Seaspray: Transformers applied to CAPTCHA-cloaked phishing websites

Federico Gerardi, *1982783, Sapienza University of Rome,*
Murad Hüseynov, *2181584, Sapienza University of Rome,*

Abstract—Seaspray is an end-to-end system designed to counter CAPTCHA cloaking, a technique used by phishers to hide malicious content from automated detection. Building on approaches like Crawl-shing, Seaspray first identifies suspicious sites, detects CAPTCHAs using YOLO, and classifies and solves them with a Vision Transformer [2] to reveal hidden content. It then verifies brand legitimacy via logo detection with YOLO [7] and visual embeddings using a ViT encoder [2], matched against known brands. By unifying CAPTCHA de-cloaking and brand verification within a Transformer-based framework, Seaspray offers a modular and robust solution.

Index Terms—Machine Learning, Computer Vision, CAPTCHA, Phishing, Cloaking, Cybersecurity

1 Introduction

Phishing emails have permeated our digital communication, taking advantage of vulnerabilities that the information technology system poses to users. Given the potential for further cybersecurity incidents, theft of personally identifiable information, and damage to organizations' assets, cybersecurity professionals have implemented various mitigation practices to combat phishing emails [18]. To address this problem, we need to detect websites that are phishing and alert the user to imminent danger. The most common approach is the use of crawlers, automated bots that with a seed of websites can explore the Internet through links. Hackers have introduced a technique called cloaking that shows two different websites to humans and crawlers. In this way, crawlers can't detect a phishing website, since they see a different page. The most common cloaking technique is CAPTCHA cloaking which, with the use of CAPTCHA, allows hackers to block the page to crawlers and make the website more reliable for the users. Seaspray enhances the current state-of-the-art in CAPTCHA decloaking, building on the work of Teoh et al. [20], by leveraging Visual Transformers [2], [25]. Specifically, we use Visual Transformers to detect and classify CAPTCHAs directly from screenshots, and then proceed to solve them. Once the CAPTCHA is solved and we can view the webpage as a regular user would, we apply a cutting-edge phishing detection method. Building on the approach by Liu and Lin [13], we use Transformer-based deep learning models to identify which brand the website is trying to represent. This step is crucial because if a site claims to be, say, a bank or a popular online service, but the domain or other details don't match the real brand, it's a strong sign the site might be a phishing attempt. Brand identification helps us catch these kinds of impersonations more accurately.

Manuscript delivered on June 29th 2025

2 Background

One effective strategy in the fight against phishing is by using crawlers. These are automated bots that browse the internet by following links to check if websites are involved in phishing or not. Criminals and phishing experts frequently leverage cloaking mechanisms to evade detection software and web crawlers. Cloaking is a technique used by phishers to evade detection, in which phishing content is selectively displayed only to entities identified as real human users, while concealing it from automated detection systems such as crawlers. This method helps phishing websites remain hidden from security tools and appear legitimate during automated scans [11].

There are several different types of Cloaking. In particular we can distinguish:

- Server side cloaking: where attackers try to predict a possible crawler from the IP Address or User Agents.
- Client side cloaking: where attackers try to predict the crawler from cookies, the way the user interact with the website.

Recent research papers like PhishTime [15] and CrawlPhish [28] show an increasing trend of CAPTCHA-protected phishing pages. Hiding phishing content behind CAPTCHAs prevents security crawlers from detecting malicious content and adds a legitimate look to phishing login pages [17].

Our appoach to detect CAPTCHA cloaking is based on Transformers. Transformer is a network architecture based on attention mechanisms, dispensing with recurrence and convolution entirely [25]. Transformer architecture was used mainly for NLP tasks, but Visual Transformers offer a nice competitor to CNN in Computer Vision tasks [2]. The idea behind Visual Transformers is to divide the picture in different patches that are normalized and encoded. These patches are passed through a self-attention layer that figures out how each patch relates to the others and evaluate an

attention score. The attention score is used as input for a multi-layer perceptron that will execute a classification task.

3 Overview of your proposed approach

Our proposal outlines a multi-stage detection pipeline, visually represented in Figure 1. This pipeline begins with a De-cloaked version of Crawl-shing [19]. The De-cloaking approach, inspired by PhishDecloaker [20] and implemented using Transformer models [25], aims to reveal concealed content. The initial output is a list of suspicious websites, which are subsequently verified through the brand identification method developed by Liu and Lin [13]. After we recognize the brand we can check if the domain is correct or not.

3.1 Crawler

We propose adopting Crawl-shing [19] to enhance our malicious content detection and optimize resource allocation. This novel approach improves significantly upon traditional methods. Crawl-shing works by crawling websites and modeling web pages as graphs, enabling sophisticated comparison against a database of known malicious sites. This process identifies suspect, potentially phishing websites. Crucially, this allows us to focus detection efforts exclusively on high-risk targets, ensuring our time and resources are utilized most effectively.

3.2 De-cloaking

A significant limitation of Crawl-shing is its vulnerability to cloaking techniques. Specifically, the presence of a CAPTCHA can render the system inoperable by preventing the construction of a web page graph. To address this, we propose a CAPTCHA De-cloaking mechanism, inspired by the approach in [20]. Our approach achieves improved performance by integrating advanced Transformer-based models.

This de-cloaking component utilizes two distinct Transformer-based models:

- YOLOv11 for Object Detection: While not exclusively a Transformer, YOLOv11 incorporates several self-attention layers within its convolutional architecture, making it highly effective for identifying the CAPTCHA challenge elements.
- Vision Transformer (ViT) for Classification: A fine-tuned Visual Transformer (ViT) model is employed for the CAPTCHA classification task, specifically trained on a dataset of CAPTCHA images.

By combining YOLOv11 for CAPTCHA detection and ViT for CAPTCHA classification, we establish a powerful CAPTCHA processing system, leveraging the strengths of a leading object detection model and a cutting-edge image classifier. Once CAPTCHA is detected by YOLOv11 and its type identified by ViT, we employ advanced transformer-based solvers, such as the one introduced by Plesner et al. [16]—to effectively solve the CAPTCHA. Following the methodology outlined in [16], this involves using a fine-tuned model, adapted to handle different visual formats of CAPTCHAs. Performs image classification in grid-based challenges and image segmentation for single-image challenges to accurately identify and select the required elements.

3.3 Brand Identification

We propose an enhancement to the SIFT-based brand identification approach introduced by Liu and Lin [13]. Their method involves extracting and matching SIFT feature points between a reference logo and a webpage screenshot to detect logo presence.

Our key improvement replaces this feature matching with an embedding-based approach leveraging a Visual Transformer (ViT), preceded by a YOLO object detection step to localize potential logo regions on the webpage. The ViT is employed as an image encoder, trained to produce rich, fixed-size vector representations (embeddings) for input images. A crucial property of this learned embedding space is that visually similar or semantically related images are mapped to proximate points. For brand identification, we first construct a database of pre-computed embeddings for known logos. During the prediction phase, we apply the YOLO model to the webpage screenshot to detect candidate logo bounding boxes. Then, for each detected region, an embedding is generated using the ViT encoder. The brand is then recognized by finding the most similar embedding within the database (e.g., using nearest neighbor search) among those generated from the detected regions. This approach offers significant advantages over traditional feature-matching techniques, granting enhanced scalability, flexibility, efficiency, and robustness by first precisely locating the logo.

3.4 Adversarial Attacks

Our strategic approach to model training incorporates data augmentation as a fundamental technique across all models. This method is employed with the specific objective of building highly robust and resilient models. By systematically creating and incorporating diverse variations of the training data – simulating potential real-world variations and perturbations – we significantly expand the effective training dataset. This expanded exposure allows the models to learn more generalizable features and patterns, fundamentally improving their ability to maintain accuracy and stability when confronted with unstructured noise and enhancing their defense mechanisms against deliberately engineered adversarial inputs designed to exploit model weaknesses.

4 Evaluation

4.1 Detection Metrics & Setup

For CAPTCHA detection, we are going to measure precision, recall, and F1-score at the object-detection level, together with mean Average Precision (mAP) and mean Average Recall (mAR) across the test set [20]. High precision minimizes false positives (incorrectly flagged legitimate images), while high recall ensures most actual CAPTCHAs are detected. We choose YOLOv11 for its proven real-time accuracy in similar tasks [7]. The experiments involve screenshots or UI renderings containing CAPTCHAs, with

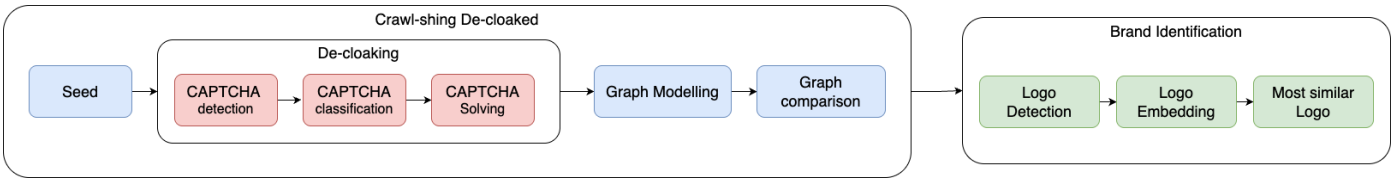


Fig. 1. Seaspray Pipeline

bounding boxes as ground truth. We are also planning to report mAP@0.5 and corresponding precision/recall to compare with prior studies. For example, PhishDecloaker, using YOLO, achieved a CAPTCHA localization mAP of approximately 0.9 [20]. Seaspray's detector will be validated on diverse pages to ensure high recall without excessive false alarms.

4.2 Classification Metrics & Setup

For CAPTCHA solving (classification), we treat it as an image recognition problem, assessing character-level accuracy, CAPTCHA solve rate, and precision/recall in multi-class scenarios. A CAPTCHA is solved only if all characters or the entire response are correct. State-of-the-art methods achieved more than 96% character-level accuracy and around 74% full CAPTCHA solve rates [26]. For example, Want et al. (2021) used a CNN-GAN solver, achieving around 96% character accuracy and 74% complete solves, with 4-8 ms GPU solve time [26]. We are going to compare Seaspray's ViT-based solver to these benchmarks, expecting competitive results due to Transformers' proven success in image classification [2]. Precision and recall will measure correctly solved CAPTCHAs (positives), and for multi-label tasks, we are going to use per-character metrics and overall sequence accuracy.

4.3 Dataset Composition & Diversity

Robust evaluation demands a diverse CAPTCHA dataset, compiled from varied styles, distortions, and providers (including reCAPTCHA [3], hCAPTCHA [5], FunCAPTCHA, various fonts, lengths, noise, etc.). Prior research, such as PhishDecloaker, used 6,612 images from 38 CAPTCHA types and evaluated generalization on 11 unseen categories [21]. Similarly, our dataset includes both synthetic and real phishing-kit CAPTCHAs to avoid overfitting. Importantly, the dataset distribution covers diverse formats, like pure text, image-click, drag-and-drop puzzles, ensuring the test set introduces unseen formats for robustness assessment. Prior studies highlight risks of model overspecialization, so it is essential to measure performance on unfamiliar CAPTCHA types. For instance, PhishDecloaker's solver maintained 86% precision and 69% recall on entirely new CAPTCHAs, which provides a useful baseline for evaluating our Transformer-based solver's generalization [22].

4.4 ViT Model Configuration

Seaspray utilizes a Vision Transformer (ViT) backbone (which has around 86 million parameters) with a default patch size of 16x16 pixels, which balances accuracy and speed. CAPTCHA images are consistently resized to ensure that characters remain distinguishable, and, if needed, smaller patches, such as 8x8, may be used. The ViT is initialized with ImageNet-pretrained weights for faster convergence, then fine-tuned on our dataset with extensive augmentation, such as random rotations, scaling, translation, noise or occlusions, color jitter, and elastic distortions, which is vital for robustness [26]. Training will likely use the Adam optimizer [8] with moderate learning rates to retain pre-trained features and early stopping based on validation accuracy. Additionally, we will test regularization methods, such as dropout and stochastic depth, to enhance generalization. Documenting the ViT configuration, such as patch size and layers, and training approach, will ensure reproducibility. Transformers have previously excelled in distorted text recognition and have sometimes surpassed CNN performance with sufficient augmentation and data [26].

4.5 Runtime Analysis

Seaspray can process CAPTCHAs so quickly that they would hardly add to page latency. YOLOv5 needs just 1.9 ms at 640x640 on an NVIDIA T4 [23], and tiny variants stay under 15 ms on a Tesla P100 [9]. Even on a CPU, lightweight YOLO models can run near real-time, for instance, around 30 FPS on a high-end i7 CPU [9]. Considering the performances of earlier YOLO models, Seaspray, powered by a YOLOv11 detector, is expected to locate a full webpage screenshot with 1080p resolution in less than 50 ms. Once found, a ViT-base solver answers in tens of milliseconds range on mid-tier GPUs [1] and in less than 7 ms on A100 or RTX 4090 [12], which is matching the fastest CNN solvers that are between 4-8 ms [26] and far ahead of GAN methods that are around 50 ms [27]. Since both stages are lightweight, pre-processing and/or post-processing of images adds only a few additional milliseconds. Taken together, detection plus solving introduce merely 0.05-0.1s to a page that already spends 0.5-2.0s on network fetch and rendering. So, one GPU comfortably clears around 20 pages per second and scales linearly with batching or more GPUs, and real-world cost is even lower because most pages contain no CAPTCHA. In contrast, PhishDecloaker's five-model browser pipeline can stall for several seconds on complex reCAPTCHAs [20], whereas Seaspray is expected to finish the same task in less than 0.1s. A CPU-only browser-extension build still wraps up in less than 0.2s using a lightweight YOLO model [9], and a single T4 server GPU scans hundreds of cloaked pages per minute [10], while SOC pipelines that inspect millions of emails per day simply add more GPU workers to keep pace [6]. Ongoing improvements in transformer libraries and GPU silicon will

continue to shave these figures, ensuring Seaspray remains comfortably fast for real-time phishing defense.

4.6 Robustness

Robustness is central to our evaluation. We will test Seaspray's CAPTCHA solver on unseen CAPTCHA formats to assess generalization capability. For example, if trained primarily on text CAPTCHAs, we will evaluate performance on image-click CAPTCHAs to identify limitations or potential extensions. While additional modules might be needed for certain interaction types, our experiments will document how the Transformer model handles variations with text CAPTCHAs. In PhishDecloaker, high precision was maintained on unseen CAPTCHA types [22]. We will similarly quantify accuracy reductions when encountering novel styles, and if the decreases are small, this will indicate strong robustness.

We will also assess adversarial robustness by actively testing our ViT model against adversarial attacks, such as FGSM and PGD, which are constrained to subtle human-unnoticeable perturbations, and measure the impacts on accuracy [22]. PhishDecloaker previously subjected their models to multiple adversarial attacks (FGSM, JSMA, PGD, DeepFool) and they showed strong resistance, which we also aim for [22]. We will further evaluate sensitivity to input variations like brightness, contrast, and scaling to ensure the model isn't overly sensitive to minor image changes.

Throughout evaluations, we will incorporate statistical significance measures. All reported performance metrics, especially when compared to baselines, will include confidence intervals or standard deviations based on multiple runs or bootstrapped samples to reflect test variability and training nondeterminism.

Finally, we will compare Seaspray directly to known benchmarks. For CAPTCHA detection, we cite relevant results, especially from PhishDecloaker [20]. For CAPTCHA solving, comparisons will include academic benchmarks and available industry claims. In addition, runtime comparisons will also be provided. For example, Seaspray solving CAPTCHAs in around 50 ms compared to significantly slower brute-force or traditional OCR methods. This comprehensive evaluation will establish Seaspray's YOLOv11 and ViT combination as effective and justified, and potentially set new standards in automated CAPTCHA decloaking for phishing detection.

4.7 Experimental Results

For the detection, we fine-tuned YOLOv8 model on a single class over 10 epochs. As shown in Figure 2, training losses decrease steadily across all epochs, indicating consistent learning progress without overfitting during training. Validation losses briefly spike around epoch 3 but quickly stabilize and drop, which is suggesting a transient instability or noisy batch early in training. After that, the model generalizes well. Precision and recall metrics show an initial dip but then improve steadily, with precision reaching near 1.0 by epoch 10 and recall above 0.95. mAP@0.5 and mAP@0.5-0.95 both rise consistently after a dip around epoch 3 and fall just under 1.0. Detailed training and validation metrics are provided in Table 2.

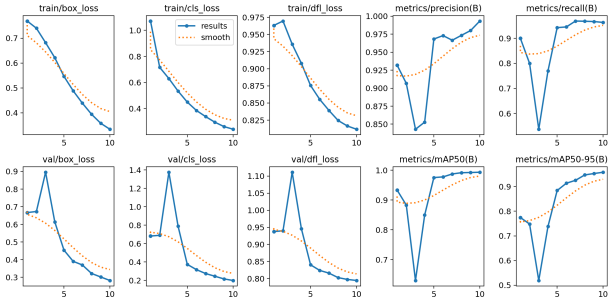


Fig. 2. Graphs indicating training/validation losses, precision, recall and mAP@0.5, mAP@0.5-0.95

For the classification, we fine-tuned Google's ViT on a diverse set of CAPTCHA classes, achieving a classification accuracy of 99% across 1244 test samples. The macro-average precision, recall, and F1 scores were consistently high at 0.99, 0.98, and 0.98 respectively. Class-wise performance, shown in the Table 1, demonstrates that most CAPTCHA types reached perfect or near-perfect scores, with slightly lower performance observed for complex visual tasks such as *"geetest_click_phrase"*, due to their increased difficulty.

Class	Precision	Recall	F1-score	Support
baidu_slide_rotate	1.00	0.95	0.97	20
dingxiang_audio	1.00	1.00	1.00	20
dingxiang_click_area	1.00	1.00	1.00	20
dingxiang_click_difference	1.00	1.00	1.00	20
dingxiang_click_font	1.00	1.00	1.00	20
dingxiang_click_icon	1.00	1.00	1.00	20
dingxiang_click_vr	1.00	1.00	1.00	20
dingxiang_click_word	0.95	1.00	0.98	20
dingxiang_drag	1.00	1.00	1.00	20
dingxiang_slide_puzzle	1.00	1.00	1.00	20
dingxiang_slide_puzzle2	1.00	1.00	1.00	20
dingxiang_slide_rotate	1.00	1.00	1.00	20
geetest_checkbox	1.00	1.00	1.00	20
geetest_click_icon	0.69	0.90	0.78	20
geetest_click_phrase	0.93	0.65	0.76	20
geetest_click_word	0.95	0.95	0.95	20
geetest_game_playing	1.00	1.00	1.00	20
geetest_game_playing2	1.00	1.00	1.00	20
geetest_select	1.00	1.00	1.00	20
geetest_slide_puzzle	1.00	1.00	1.00	20
hcaptcha	1.00	1.00	1.00	20
hcaptcha_checkbox	1.00	1.00	1.00	20
netease_click_icon	1.00	1.00	1.00	20
netease_click_phrase	0.95	1.00	0.98	20
netease_click_vr	1.00	1.00	1.00	20
netease_click_word	1.00	1.00	1.00	20
netease_drag	1.00	1.00	1.00	20
netease_slide	1.00	0.95	0.97	20
press_and_hold	1.00	1.00	1.00	20
recaptchav2	1.00	1.00	1.00	20
recaptchav2_checkbox	1.00	1.00	1.00	20
tencent_slide	1.00	1.00	1.00	20
text_1	0.98	0.97	0.98	64
text_2	0.99	1.00	0.99	200
text_3	1.00	1.00	1.00	200
text_4	1.00	0.97	0.98	100
text_5	1.00	1.00	1.00	20
text_6	1.00	1.00	1.00	20
Accuracy			0.99	1244
Macro avg	0.99	0.98	0.98	1244
Weighted avg	0.99	0.99	0.99	1244

TABLE 1
Classification report for 38 captcha types

Qualitative analysis of training batches shown in Figure 3 for one representative training batch and one valida-

tion batch in Figure 4 reveal accurate CAPTCHA localization with high confidence predictions, mostly 0.9 to 1.0.

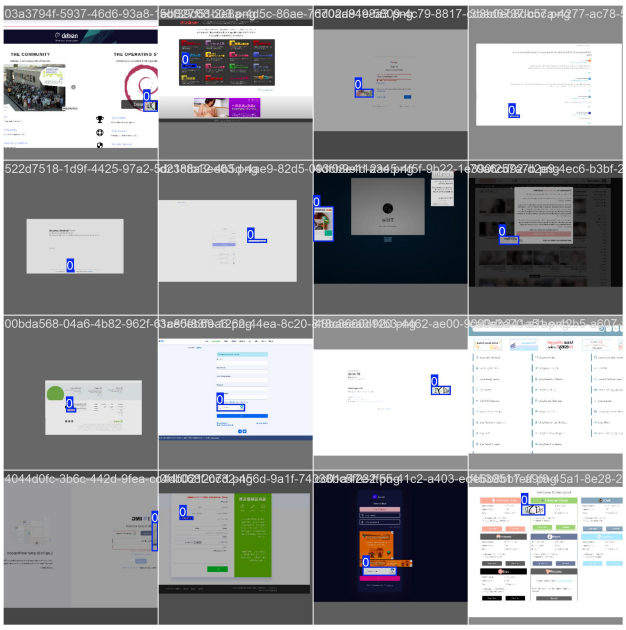
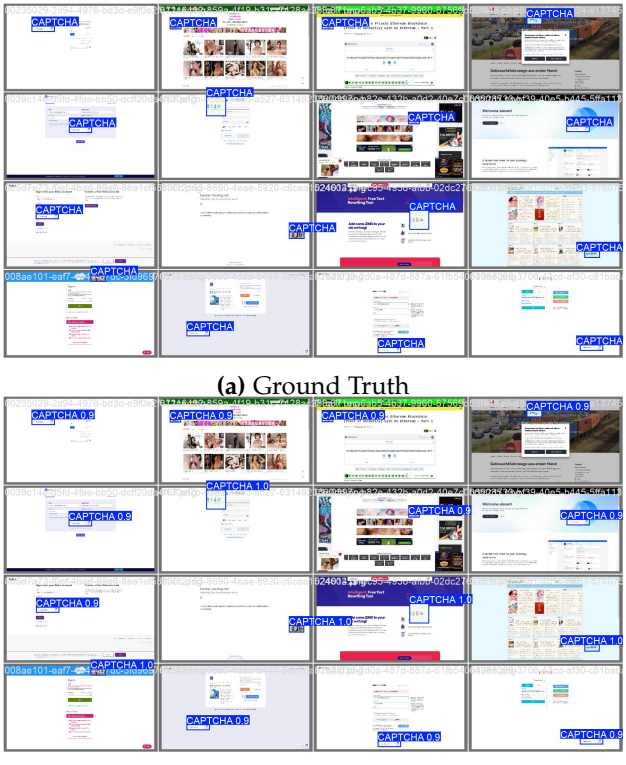


Fig. 3. One training batch result



(a) Ground Truth

(b) Predictions

Fig. 4. One validation batch: (a) Ground truth labels and (b) model predictions.

5 Related Work

5.1 CAPTCHA Cloaking in Phishing

CAPTCHA cloaking has become a popular technique in phishing where attackers show automated scanners only a blank or CAPTCHA page while real users who solve the puzzle reach the malicious content [15] [28]. *Oest et al.* reported that both server-side and client-side cloaking has become a standard in phishing kits around 2018-2019 [28] [15]. CrawlPhish (Zhang et al. 2021) conducted a large-scale analysis of client-side evasion in phishing websites, and the results show that between 23-33% of phishing sites used some form of cloaking and many employed fake CAPTCHA pages that mimic Google reCAPTCHA [28]. These gatekeeper pages block security crawlers and also look credible to users [28]. Another study, PhishTime (Van Acker et al., USENIX Security 2020), evaluated how different evasion techniques affect phishing detection latency. They showed that CAPTCHA-based cloaking can fully bypass automated URL blacklists. In the experiments, when a phishing URL used a pre-content CAPTCHA, detection rates from popular tools dropped to 0% [20] [15]. In one case, Microsoft's SmartScreen detected one such phish only because it flagged the obfuscated CAPTCHA script as malicious rather than the hidden content [15]. This showcases how traditional detectors are essentially blind for the true content behind CAPTCHAs. As defenders improve visual scanning, phishers add more sophisticated cloaking to avoid modern defenses. A confirming report was published by *Unit 42* indicating 7,572 unique CAPTCHA-protected phishing URLs within a single month in 2020 [17].

5.2 Existing Approaches to CAPTHA Cloaking Detection

PhishDecloaker (Teoh et al., USENIX Security 2024) is the first dedicated solution for CAPTCHA-cloaked phishing sites. It combines vision-based recognition with automatic interaction [20]. In particular, it uses five deep learning models to recognize many CAPTCHA formats and imitates human actions, like moving the cursor and waiting [22] [20]. By solving the puzzle, it exposes the underlying page, so that the existing detectors can work. When PhishDecloaker was paired with systems such as Phishpedia, overall detection rate on cloaked pages jumped from 0% to 74.25% [22]. In tests with previously unseen CAPTCHAs, it achieved 86% precision and 69% recall [22]. It remained effective against adversarial perturbations produced with FGSM, JSMA, PGD, and DeepFool [22]. Moreover, a 30-day field study with PhishDecloaker showed it discovered 7.6% more phishing websites that were using CAPTCHA cloaking compared to other existing detectors [22].

5.3 Comparison of Techniques

Early phishing detectors relied on two main ideas, visual similarity and DOM or content analysis, with visual methods dominating. PhishZoo (Afroz et al., 2011) and VeriLogo (2012) are examples of early methods that computed Scale-Invariant Feature Transform (SIFT) keypoints on screenshot and compared them with a logo database [14]. This worked at the time, yet suffered from high false positives and could not scale to many brands. Phishpedia (Lin et al., USENIX Security 2021) improved matters by using a Faster R-CNN to spot the most visible logo on the page, then applied SIFT matching to confirm which brand it belonged to, and cross-checked that logo against

Epoch	Time	Train Box	Train Cls	Train DFL	Prec(B)	Rec(B)	mAP50(B)	mAP50-95(B)	Val Box	Val Cls	Val DFL	lr/pg0	lr/pg1	lr/pg2
1	158.755	0.76869	1.07143	0.96309	0.93202	0.89991	0.93342	0.77453	0.66686	0.68237	0.93768	0.000665	0.000665	0.000665
1	278.552	0.76869	1.07143	0.96309	0.93202	0.89991	0.93342	0.77453	0.66686	0.68237	0.93768	0.000665	0.000665	0.000665
1	381.178	0.76869	1.07143	0.96309	0.93202	0.89991	0.93342	0.77453	0.66686	0.68237	0.93768	0.000665	0.000665	0.000665
1	155.922	0.76869	1.07143	0.96309	0.93202	0.89991	0.93342	0.77453	0.66686	0.68237	0.93768	0.000665	0.000665	0.000665
2	294.168	0.73972	0.71613	0.96952	0.9067	0.79997	0.88302	0.74803	0.67292	0.69000	0.93971	0.001300	0.001300	0.001300
3	428.711	0.68171	0.63035	0.93568	0.84289	0.53698	0.63034	0.51924	0.89544	1.37515	1.11133	0.001810	0.001810	0.001810
4	562.846	0.62147	0.53413	0.90791	0.85282	0.76970	0.84949	0.73801	0.61409	0.78886	0.94615	0.001592	0.001592	0.001592
5	696.475	0.54648	0.44997	0.87551	0.96829	0.94266	0.97525	0.88400	0.45375	0.37442	0.84008	0.001316	0.001316	0.001316
6	830.941	0.48859	0.38546	0.85528	0.97305	0.94559	0.97748	0.91345	0.39048	0.31710	0.82367	0.001010	0.001010	0.001010
7	965.607	0.43906	0.33857	0.83922	0.96659	0.96913	0.98729	0.92525	0.36976	0.27466	0.81575	0.000704	0.000704	0.000704
8	1100.39	0.39442	0.29584	0.82455	0.97342	0.96819	0.99155	0.94701	0.32169	0.24548	0.80254	0.000428	0.000428	0.000428
9	1234.55	0.35816	0.26210	0.81657	0.98009	0.96708	0.99225	0.95277	0.30167	0.21620	0.79733	0.000209	0.000209	0.000209
10	1368.97	0.33423	0.24165	0.81178	0.99330	0.96352	0.99308	0.95799	0.28187	0.19940	0.79421	0.0000685	0.0000685	0.0000685

TABLE 2
Training and validation metrics across epochs

the domain [13] [24]. With this approach, the accuracy was good when the page was fully visible. However, that assumption broke when CAPTCHAs hid the content. In PhishDecloaker's tests, both Phishpedia and PhishIntention fell to zero detection once a CAPTCHA blocked the page [20]. PhishIntention (Liu et al., NDSS 2021) takes a visual approach by segmenting a webpage into regions, such as forms and images, and feeding each of them through a CNN to infer the intended brand. It reportedly achieves high precision in identifying phishing pages by the layout and logos. However, PhishIntention and similar CNN-based schemes are prevented by content-blocking CAPTCHAs, as they cannot analyze what they can't load. SIFT approaches are also fragile when logos are slightly modified or when CAPTCHA images introduce noise, whereas deep networks that learn feature embeddings cope better with such variations. Seaspray's adoption of Transformer-based models (ViT) specifically addresses key weaknesses of CNNs and traditional SIFT models. Unlike CNNs, which rely heavily on localized features, Transformers inherently capture broader global context through self-attention mechanisms, which enable handling of distorted or noisy CAPTCHA images. Once the page is revealed, a logo detector generates an embedding vector with a pretrained network like a Siamese network or contrastive-trained model on logos to generate an embedding vector and compare it with vectors for known brands. This method enables Transformers to identify logos despite slight modifications, color shifts, or stylized changes. The embedding approach also inherently generalizes better, as adding new brands requires merely placing their logos in the reference database without retraining, which allows seamless adaptation to novel scenarios. To our knowledge, Seaspray is the first pipeline that couples transformer CAPTCHA de-cloaking with logo verification through vision embeddings. While PhishDecloaker only focuses on solving the CAPTCHA, and then handing the page to external detector, Seaspray integrates all steps that lets the CAPTCHA solver and the logo verifier share information and improves overall robustness.

5.4 Performance Comparisons

CNN-based solutions, such as Phishpedia and PhishIntention report high accuracy on standard phishing pages. For example, PhishIntention had near 90% precision in some studies, but those numbers dropped sharply to 0% for cloaked pages until the CAPTCHA is removed [14]. PhishDecloaker showed that by adding an automated CAPTCHA solver, one could boost detectors' success on cloaked pages to around 74% [22]. Seaspray aims to push that even further by using more powerful models. Also by using Transformers and extensive data augmentation, we expect Seaspray to handle a wide range of CAPTCHAs, including ones not seen before and be resistant to adversarial manipulations. The arms race will continue as attackers will try new tactics by designing CAPTCHAs that are easy for humans to solve and tricky for machines. The recently explored academic proposals show that this can be potentially achieved using adversarial examples or diffusion models that confuse machine vision [26]. However, Seaspray's modular pipeline eases adaptation because its solving, page analysis, and brand verification components can be upgraded separately. We also note that Seaspray could be combined with complementary defenses. For example, *Unit 42* study found that checking for reuse of the same reCAPTCHA site key across many pages is a strong indicator of phishing campaigns [17]. This is a "back-end" signal that does not require solving the CAPTCHA at all. A comprehensive anti-phishing system could use Seaspray to analyze visual content while also correlating such network or key reuse signal to catch even more attacks.

5.5 Novelty and Transferability

Seaspray's use of learned visual embeddings for brand logos offers value well beyond phishing pages. A single logo detector and embedding matcher can be reused with almost no adjustment to spot brand impersonation in fake product photos and app icons, or scam social-media profiles. Earlier systems such as Phishpedia, which relied on SIFT matching, and PhishIntention, which trained a CNN only for webpage screenshots, were limited to one scenario. In contrast, Seaspray's module could easily scan email attachments or PDF files for trademarked images and flag suspicious messages that carry official branding. Because an embedding space captures how logos look across many styles and variations, features learned for phishing transfer smoothly to other security tasks [14] [4]. Seaspray therefore tackles CAPTCHA-protected phishing while also delivering a flexible vision component that improves protection across several domains which is something prior work did not address.

6 Conclusions

Seaspray introduces an end-to-end pipeline to defeat CAPTCHA-cloaked phishing. It improves upon existing

graph-based crawlers like Crawl-shing [19], which are vulnerable to CAPTCHA cloaking. Seaspray addresses this by first using YOLO and a Vision Transformer (ViT) to detect and solve CAPTCHAs in HTML/screenshots. Once the page is accessible, it applies embedding-based brand logo detection to verify spoofed brands. Apart from being modular and adaptable, its components can be retrained for new CAPTCHA styles or brands. Deployable in Security Operations Center (SOC) workflows, browser extensions, or email filters, it operates in real time on modest hardware. By unifying de-cloaking and verification, Seaspray shifts the balance toward defenders, forcing attackers to adopt costlier evasion tactics.

REFERENCES

[1] R. S. Bhowmick, R. Indra, I. Ganguli, J. Paul, and J. Sil, "Breaking CAPTCHA system with minimal exertion through deep learning: Real-time risk assessment on indian government websites," vol. 4, no. 2, pp. 1–24. [Online]. Available: https://dl.acm.org/doi/10.1145/3584974

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," Jun. 2021, arXiv:2010.11929 [cs]. [Online]. Available: http://arxiv.org/abs/2010.11929

[3] Google Developers. Google reCAPTCHA documentation. [Online]. Available: https://developers.google.com/recaptcha

[4] Q. Hao, N. Diwan, Y. Yuan, G. Apruzzese, M. Conti, and G. Wang, "It doesn't look like anything to me: Using diffusion models to subvert visual phishing detectors." [Online]. Available: https://gangw.cs.illinois.edu/logomorph.pdf

[5] hCaptcha Team. hcaptcha documentation. [Online]. Available: https://docs.hcaptcha.com/

[6] N. Huq, P. Lin, R. Reyes, and C. Perine. Navigating the Threat Landscape for Cloud-Based GPUs. Trend Micro. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/news/threat-landscape/navigating-the-threat-landscape-for-cloud-based-gpus

[7] R. Khanam and M. Hussain, "YOLOv11: An overview of the key architectural enhancements." [Online]. Available: http://arxiv.org/abs/2410.17725

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[9] LearnOpenCV Team. Performance comparison of YOLO object detection models – an intensive study. LearnOpenCV. [Online]. Available: https://learnopencv.com/performance-comparison-of-yolo-models/

[10] C. Li, L. Li, Y. Geng, H. Jiang, M. Cheng, B. Zhang, Z. Ke, X. Xu, and X. Chu, "YOLOv6 v3.0: A full-scale reloading," Jan 2023. [Online]. Available: https://arxiv.org/pdf/2301.05586

[11] W. Li, S. Manickam, S. U. A. Laghari, and Y.-W. Chong, "Uncovering the cloak: A systematic review of techniques used to conceal phishing websites," vol. 11, pp. 71 925–71 939. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10175532

[12] Y. Li, G. Yuan, Y. Wen, J. Hu, G. Evangelidis, S. Tulyakov, Y. Wang, and J. Ren, "EfficientFormer: Vision transformers at mobilenet speed." [Online]. Available: https://arxiv.org/pdf/2206.01191

[13] Y. Lin, R. Liu, D. M. Divakaran, J. Y. Ng, Q. Z. Chan, Y. Lu, Y. Si, F. Zhang, and J. S. Dong, "Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 3793–3810. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/lin

[14] R. Liu, Y. Lin, X. Teoh, G. Liu, Z. Huang, and J. S. Dong, "Less defined knowledge and more true alarms: Reference-based phishing detection without a pre-defined reference list." [Online]. Available: https://www.usenix.org/system/files/usenixsecurity24-liu-ruofan.pdf

[15] A. Oest, Y. Safaei, P. Zhang, B. Wardman, K. Tyers, Y. Shoshitaishvili, and A. Doupé, "{PhishTime}: Continuous longitudinal measurement of the effectiveness of anti-phishing blacklists," pp. 379–396. [Online]. Available: https://yancomm.net/papers/2020%20-%20USENIX%20Security%20-%20PhishTime.pdf

[16] A. Plesner, T. Vontobel, and R. Wattenhofer, "Breaking reCAPTCHAv2," in *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*, Jul. 2024, pp. 1047–1056, arXiv:2409.08831 [cs]. [Online]. Available: http://arxiv.org/abs/2409.08831

[17] O. S. B. S. Starov, Billy Melicher. Discovering CAPTCHA protected phishing campaigns. [Online]. Available: https://unit42.paloaltonetworks.com/captcha-protected-phishing/

[18] Y. E. Suzuki and S. A. S. Monroy, "Prevention and mitigation measures against phishing emails: a sequential schema model," vol. 35, no. 4, pp. 1162–1182. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8478002/

[19] F. Tchakounte, J. C. T. Ngnintedem, I. Damakoa, F. Ahmadou, and F. A. K. Fotso, "Crawl-shing: A focused crawler for fetching phishing contents based on graph isomorphism," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, Part A, pp. 8888–8898, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157821003037

[20] X. Teoh, Y. Lin, and J. S. Dong, "PhishDecloaker: Detecting CAPTCHA-cloaked phishing websites via hybrid vision-based interactive models."

[21] X. Teoh, Y. Lin, R. Liu, Z. Huang, and J. S. Dong, "CAPTCHA recognition dataset." [Online]. Available: https://zenodo.org/records/11228974

[22] ——, "Phishdecloaker: Detecting CAPTCHA-cloaked phishing websites via hybrid vision-based interactive models." USENIX Association, pp. 505–522. [Online]. Available: https://www.classcentral.com/course/youtube-usenix-security-24-phishdecloaker-detecting-captcha-cloaked-phishing-websites-via-hybrid-378861

[23] Ultralytics Team. YOLOv5 vs. YOLOv8: A detailed comparison. Ultralytics. [Online]. Available: https://docs.ultralytics.com/compare/yolov5-vs-yolov8/

[24] T. van den Hout, T. Wabeke, G. C. M. Moura, and C. Hesselman, "Logomotive: Detecting logos on websites to identify online scams – a TLD case study," in *Passive and Active Measurement, 23rd International Conference, PAM 2022, Proceedings*, ser. Lecture Notes in Computer Science, vol. 13210. Springer, pp. 3–29. [Online]. Available: https://logomotive.sidnlabs.nl/downloads/LogoMotive_paper.pdf

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need." [Online]. Available: http://arxiv.org/abs/1706.03762

[26] Y. Wang, Y. Wei, M. Zhang, Y. Liu, and B. Wang, "Make complex CAPTCHAs simple: A fast text captcha solver based on a small number of samples," *Information Sciences*, vol. 578, pp. 181–194. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025521007301

[27] G. Ye, Z. Tang, D. Fang, Z. Zhu, Y. Feng, P. Xu, X. Chen, and Z. Wang, "Yet another text captcha solver: A generative adversarial network based approach," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*. ACM, 2018, pp. 2315–2332. [Online]. Available: https://doi.org/10.1145/3243734.3243754

[28] P. Zhang, A. Oest, H. Cho, Z. Sun, R. Johnson, B. Wardman, S. Sarker, A. Kapravelos, R. Wang, Y. Shoshitaishvili, A. Doupe, and G.-J. Ahn, "CrawlPhish: Large-scale analysis of client-side cloaking techniques in phishing." [Online]. Available: https://www.kapravelos.com/publications/crawlphish-sp21.pdf