A Appendix

Due to the space limitation, the full version of the Appendix is available at our code repository. Specific references to the corresponding contents are specified below.

A.1 Model Summary

We conduct a comprehensive literature review of top conferences and highly cited papers from 2005 to 2023 to identify popular visual similarity-based models for phishing detection, as summarized in Table 10. The gray sharing in the table indicates the seven models selected for re-training and evaluation. Liu *et al.* [101] compare features like layout, colors, fonts, and image placement, while EMD [30] uses Earth Mover's Distance to measure visual similarity. Medvet *et al.* [67], CCH [18], and Goldphish [24] analyze discriminative key points, employ image hashing techniques, and leverage classifiers for phishing detection, respectively. More recent approaches like Phishpedia [59] combine text and visual content analysis, while OpenGlue [99], Bernabeu *et al.* [11], OSLD [10], Bhurtel *et al.* [12], and SeeTek [53] explore advanced deep learning techniques for image retrieval, logo recognition, and visual similarity assessment. In particular, the use of deep learning (DL) techniques since 2019 shows a growing trend for visual similarity-based phishing detection.

A.2 Selected Models

Based on the candidate papers, we carefully selected seven models in Table 11 with diverse architectures, input types, and detection methods to compare visual similarity-based phishing detection approaches comprehensively. Particularly, four of them take screenshots, URLs, and HTML as input, while three of them take screenshots as input.

A.3 FLOPs and Parameters Performance For Components

We compare the FLOPs and parameters for key model components in Table 12. `DynaPhish`, `PhishIntention`, and `Phishpedia` have similar model structures, resulting in close parameters and FLOPs for detecting logos and siamese modules. We exclude the CRP locator and web interaction parts of `PhishIntention` and `DynaPhish`, as our URLs may not be alive now. Furthermore, the online search function of `DynaPhish` is not included in the calculation. `Involution` employs the same module with `Phishpedia` for logo cropping and thus shares the parameter size. `EMD` and `PhishZoo` are not taken into consideration because they do not use neural networks.

A.4 Failure Examples Categorization

To better understand the limitations of visual similarity-based phishing detection models, we analyzed the failure cases observed during our real-world evaluations in Section 5. We categorized these failure examples into four main categories: logo, popup, login form, and other related issues, as summarized in Table 13. Logo-related issues (L1-L22) encompass various manipulations and alterations to the logo, such as elimination and color replacement. These issues highlight models' challenges in accurately identifying and comparing logos under diverse visual variations. Popup-related issues (P1-P4) pose a significant threat. They involve the presence of popups, advertisements, cookies, alerts, and other overlays on the screenshot. These elements can obstruct or confuse the visual analysis of the web pages, potentially leading to misclassifications by the phishing detection models and, consequently, to successful phishing attacks. Login form-related issues (F1-F5) include changes to the login form's text, color, language, font, and other website login forms as a phishing tactic. These variations in the login form's appearance and design can make it difficult for models to identify phishing attempts based on visual similarity alone accurately. Other manipulations (O1-O3) include adding extra text on the screenshot and some pages that are blocked.

A.5 Perturbated and SRNet

The perturbed and SRNet logo samples cropped from screenshots by Faster-RCNN are shown in Figure 4. Three white-box attacks, two black-box attacks, and SRNet methods are summarized as follows:

Fast Gradient Sign Method (FGSM). [33]: A white-box attack that perturbs the data in a single step by using an imperceptibly small vector, elements are equal to the sign of the gradient of the cost function with respect to the input.

Projected Gradient Descent (PGD). [66]: An iterative version of the FGM attack with a random start, which applies the perturbations multiple times to create a more effective adversarial example.

Carlini & Wagner (CW). [15]: A strong white-box attack that optimizes the perturbations to minimize the detection confidence of the target model. Constructing three new attacks for the three distance metrics.

[51]-**ViT** and [51]-**Swin.** Black-box attacks that utilize generative adversarial perturbations to develop adversarial logos. Taking the trained Vision Transformer (ViT) [23] and Swin Transformer [65] models as Discriminators and a Deep Residual Network with six residual blocks (ResNet-6) as the foundational architecture of the Generator.

Style Retention Network (SRNet). [106]: A generative adversarial network (GAN) that aims to transfer the style of images to another while preserving the content.

Table 10: List of Visual Similarity-based Models (Y/N: open source code or not, Y*: reproduced by non-original authors).

Year	Model	Description	DL	Code	Data Source
2005	Liu *et al.* [101]	Compares features like layout, colors, fonts, and image placement of webpages for phishing detection	N	N	N
2006	EMD [30]	Uses Earth Mover's Distance to assess visual similarity of webpages for phishing detection	N	Y* [60]	N
2008	Medvet *et al.* [67]	Relies on visual similarity, potentially comparing features like layout, colors, and overall webpage appearance, to detect phishing	N	N	PhishTank [80], Alexa
2009	CCH [18]	Employs discriminative keypoint features to distinguish phishing websites based on visual cues	N	N	N
2010	Goldphish [24]	Analyzes images for phishing detection, possibly using techniques like image recognition or text extraction from images	N	N	PhishTank [80]
2011	Zhang *et al.* [110]	Combines textual and visual content analysis with a Bayesian approach for phishing detection	Y	N	N
2011	msDT [45]	Introduces a method for logo recognition (not phishing specific) based on triangulation	N	N	Flickr [29]
2011	PhishZoo [4]	Analyzes visual appearance of webpages, likely using techniques to compare layout, colors, fonts, and potential images	N	Y* [60]	PhishTank [80], Alexa
2013	Chang *et al.* [16]	Focuses on website identity recognition, possibly using techniques like domain name analysis or website structure comparison	N	N	PhishTank [80], Alexa
2013	Romberg *et al.* [84]	Proposes bundle min-hashing for logo recognition	N	Y [69]	Flickr [29]
2015	FaceNet [87]	Deep learning architecture for face recognition (not directly related to phishing)	Y	Y [26]	LFW [43], Youtube [104]
2015	Rao *et al.* [81]	Presents a computer vision technique for phishing detection using visual similarity	N	N	PhishTank [80]
2015	LOGO-Net [40]	Leverages deep learning for logo detection (not directly related to phishing)	Y	N	N
2016	Bozkir *et al.* [14]	Uses HOG descriptors for feature extraction to potentially compare webpages for phishing detection	N	N	N
2017	DeltaPhish [21]	Compares the static features of HTML and visual appearance of the potential phishing pages against compromised websites	N	N	PhishTank [80]
2017	Haruta *et al.* [37]	Combines image and CSS analysis with a target website finder for phishing detection	N	N	Alexa
2019	Sharma *et al.* [89]	Deep learning approach for image retrieval (adaptable to phishing)	Y	Y [31]	N
2020	CSQ [108]	Deep learning method for image/video retrieval (adaptable to phishing)	Y	Y [32]	ImageNet [22]
2020	VisualPhishNet [2]	Proposes zero-day phishing website detection based on visual similarity	Y	Y [85]	N
2021	Involution [54]	Inverting the inherence of convolution for visual recognition	Y	Y [44]	Cityscapes [20]
2021	Dooremaal *et al.* [98]	Combining text and visual features to improve the identification of cloned webpages for early phishing detection	Y	N	Dataset [78]
2021	Phishpedia [59]	Employs a hybrid deep learning approach for visual phishing detection	Y	Y [61]	Phishpedia [79]
2022	PhishIntention [62]	Uses deep learning to analyze webpage appearance and dynamics for inferring phishing intention	Y	Y [60]	PhishiIntention [60]
2022	OpenGlue [99]	Open-source deep learning pipeline for image matching (not directly related to phishing)	Y	Y [1]	MegaDepth [57]
2022	Bernabeu *et al.* [11]	Leverages deep learning for multi-label logo recognition (not directly related to phishing)	Y	N	METU [68]
2022	OSLD [10]	Deep learning approach for large-scale logo detection (not directly related to phishing)	Y	N	OSLD [76]
2022	Bhurtel *et al.* [12]	Relies on machine learning with a Siamese network for logo recognition for phishing detection	Y	N	LogoSENSE [13]
2022	SeeTek [53]	Deep learning for large-scale logo recognition with text integration (not directly related to phishing)	Y	N	PL8K
2023	DynaPhish [63]	Using deep learning approaches to analyze webpages and Google search to identify brand intention	Y	Y [25]	DynaPD [63]

Where deeper shades of ▨ indicate the seven models that we select for retraining and evaluation.

Table 11: Description of Used Seven Model Information.

Model Name	Training Dataset	Input	Description
EMD [30]	—	S	Calculate distance by EMD through color and coordinate feature
PhishZoo [4]	—	S, U, H	Use TF-IDF on URL and HTML for profile matching and use the SIFT feature for image matching
VisualPhishNet [2]	$\boldsymbol{R}_{ext}$	S	Use Triplet CNN to learn similarities of the same websites' screenshots and dissimilarities between different websites' screenshots.
Involution [54]	Logo2K+, $\boldsymbol{R}_{base}$ or $\boldsymbol{R}_{ext}$	S	Use Faster-RCNN to find the logo region, learn logo representations through Involution, and then compare cosine similarity
Phishpedia [59]	Logo2K+, Benign30K, $\boldsymbol{R}_{base}$ or $\boldsymbol{R}_{ext}$	S, U, H	Contains a layout classifier designed to detect and locate the logo region within images, and a Siamese neural network model that analyzes the identified logo to recognize and classify the brand it represents
PhishIntention [62]	Logo2K+, Sampled Benign30K, $\boldsymbol{R}_{base}$ or $\boldsymbol{R}_{ext}$	S, U, H	Contains a layout classier part to find the different components' regions, a CRP classifier to check if the screenshot has CRP, an HTML static classifier to check whether have CRP, a CRP locator to find additional links' CRP, and a Siamese model to recognize the logo's brand
Dynaphish [63]	—	S, U, H	Based on [62] and [59], it contains a Google search part to check targeted brands and dynamically expand reference lists

*__Testing Dataset__ = APWG Dataset, Manipulating Dataset; **__Brand Reference List__ = Baseline Ref. $\boldsymbol{D}_{base}$, Extended Ref. $\boldsymbol{D}_{ext}$;
S = Screenshot; **U** = URL; **H** = HTML;

Table 12: FLOP and Parameters Performance for Components

Model	Parameters/FLOPs				
	Detect Logo	Siamese	CRP Classifier	Others	Total
DynaPhish	41.32M/203G	24.10M/1.35G	23.50M/11.31G	—	88.92M/215.66G
PhishIntention	41.32M/203G	24.10M/1.35G	23.50M/11.31G	—	88.92M/215.66G
Phishpedia	41.30M/211G	24.10M/1.35G	—	—	65.40M/212.35G
Involution	41.30M/211G	—	—	12.01M/1.67G	53.04M/212.67G
VisualPhishNet	—	—	—	21.27M/92.49G	21.27M/92.49G

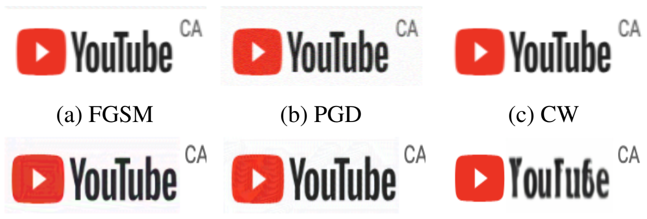
A.6 Examples of Visible Manipulation

Focusing on the logo component, we randomly select the samples that can make models fail. We also selected the samples based on the manipulations used by the adversaries. Detailed examples can refer to Figure 6.

Additionally, Figure 5 shows an example that `PhishIntention` correctly identified but `Phishpedia` failed to recognize.

Table 13: Failure Categorization in Our Dataset.

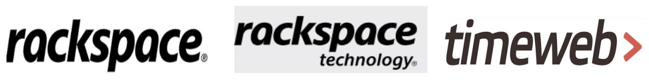
	ID	Name	Description
Logo	L1	Similar	Similar to the reference list
	L2	Elimination	Screenshots delete logos
	L3	BrokenImage	Logo images are damaged
	L4	ColorReplace	Different colors of logos
	L5	LogoBackground	Different backgrounds of logos
	L6	Integration	Logos are combined with other logos
	L7	Re-position	Logos appear in different locations on the screenshot
	L8	Outdated	Logos are not in the reference list
	L9	CaseConversion	Changing the case of textual logos
	L10	TextAsLogo	Type text as the logo
	L11	Scaling	Enlarge or shrink logos
	L12	Resizing	Logos' height-to-width are changed
	L13	FontReplace	Changing the textual font of logos
	L14	Omission	Only partial logos are used
	L15	Shape	Logo with different shapes, like square, rectangular
	L16	LogoAddText	Add text close to the logos
	L17	Replacement	Screenshots replace logos with other logos
	L18	Rotation	Logos are rotated in some angles
	L19	Flipping	Flipping logo by vertical or horizontal
	L20	Blurring	The logo or screenshot is blurred
	L21	CraftLogo	Craft logos based on different information
	L22	Language	Change the textual logos language
Popup	P1	LoginPopup	Login forms pop-up on the screenshot
	P2	AdPopup	Advertisements pop-up on the screenshot
	P3	CookiePopup	The cookie pop up on the blurred screenshot
	P4	OtherPopup	Alert, remind, location, etc.
Login	F1	LoginForm	Change login form text (text, color, language, fonts)
	F2	Button	Change button color, shape, location, text, etc.
	F3	NewForm	Design a new form
	F4	ThirdParty	Use other websites as login methods
	F5	QR	Login by scanning the QR code
Others	O1	ImageAddText	Add text on the screenshot not close to logo areas
	O2	Blocked	The image is blocked, only left text
	O3	ImageBackground	Different backgrounds of screenshots



(a) FGSM (b) PGD (c) CW

(d) ViT (e) Swin (f) SRNet

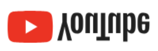
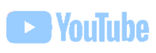
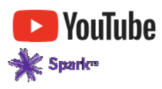
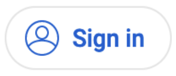
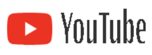
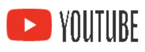
Figure 4: Perturbated Logos Cropped by Faster-RCNN.

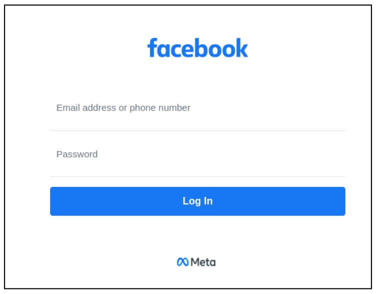


(a) Rackspace brand target list example (b) Recent searched Rackspace logo (c) Misidentified target logo

Figure 5: Text Logo Case (Original logo and benign URL).

Table 14: Example and Description of Visible Manipulation Methods.

Example	Method Description	Example	Method Description
	Original: This is the original logo cropped from the "YouTube" original website.		**Flipping:** We flip the logo vertically or horizontally. It differs from "Rotation," where we control the rotation to a small degree.
	**Color Replacement:** We identify the logo in the screenshot and then replace the color. In this example, we change the original red to blue, but the attacker could use any other predefined color.		**Resizing:** We randomly modify the height-to-width ratio of the logo. Note that logo resizing does not necessarily maintain the proportion.
	Rotation: We rotate the logo in small increments clockwise or counterclockwise, and fill the empty area created by the rotation with the color of the surrounding background. In this example, it is rotated clockwise by one degree.		**Integration:** We randomly select a second logo from a set of 110 different target brands and place it either above, below, or to the left of the original logo in the screenshot. For example, the "YouTube" is combined with "Spark NZ."
	Replacement: We replace the original logo with a logo randomly selected from 110 brands. For example, the login form is still "YouTube," but the logo is replaced with "Raiffeisen Bank."		**Scaling:** We scale up the logo, increasing both the length and width to 1.1 times the original size. Then, we place the resized logo in the screenshot of "Elimination."
	**Elimination:** We remove the logo from the screenshot and fill the area with the surrounding background color. The region detector may identify other components ("sign in") as the logo.		**Blurring:** We add Gaussian blurring with kernel size 9 to the entire screenshot image, including the logo and the background, by the "OpenCV" Python package.
	Re-position: We move the position of the logo horizontally within the screenshot and fill it with the surrounding background color. The example is cropped from the screenshot when the logo is moved from the top left to the bottom left.		**Omission:** We use only one of the elements of the logo (either icon or text) and fill the rest with the surrounding color. For example, we keep the icon and remove the text "YouTube."
	**Font Replacement:** We use a font identification tool [102] to find similar fonts. Then, we generate text in those fonts and replace the original logo. We also use the SRNet [106] to generate text logos while keeping the background context, font style, and color.		**Case Conversion:** We find a font that looks similar to the text logo and then change the capitalization of the text to make all letters capitalized, all letters lowercase, or just the first letter capitalized. For example, "YouTube" is transformed into "YOUTUBE."


facebook

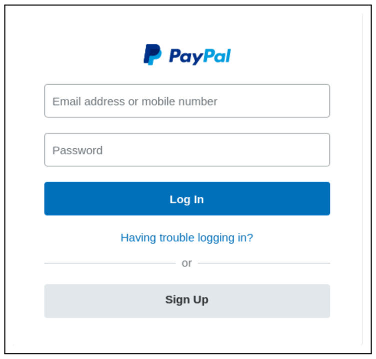
Email address or phone number

Password

Log In

Meta

(a) EMD Failed Example


PayPal

Email address or mobile number

Password

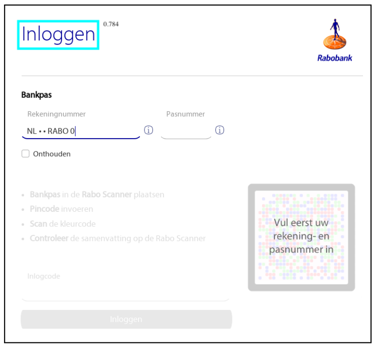
Log In

Having trouble logging in?

or

Sign Up

(b) VisualPhishNet Failed Example


Inloggen 0.784

Rabobank

Bankpas

Rekeningnummer

NL • • RABO 0

Pasnummer

Onthouden

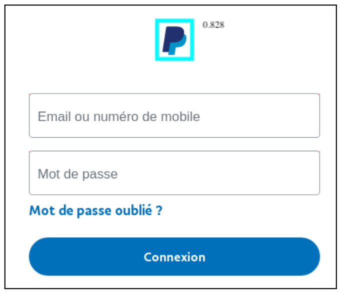
- Bankpas in de Rabo Scanner plaatsen
- Pincode invoeren
- Scan de kleurcode
- Controleer de samenvatting op de Rabo Scanner

Vul eerst uw rekening- en pasnummer in

Inlogcode

Inloggen

(c) Wrong Logo Area and QR Code


0.828

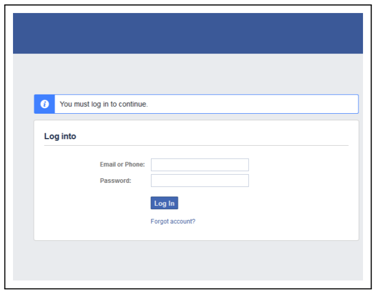
Email ou numéro de mobile

Mot de passe

Mot de passe oublié ?

Connexion

(d) Similar But Lower Than the Threshold


You must log in to continue.

Log into

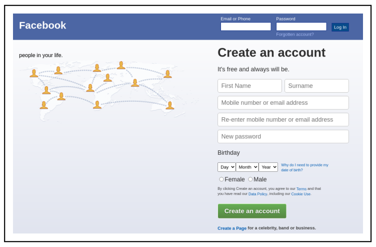
Email or Phone:

Password:

Log In

Forgot account?

(e) Elimination of Logo


Facebook

Email or Phone

Password

Log In

Forgotten account?

people in your life.

Create an account

It's free and always will be.

First Name

Surname

Mobile number or email address

Re-enter mobile number or email address

New password

Birthday

Day Month Year

Why do I need to provide my date of birth?

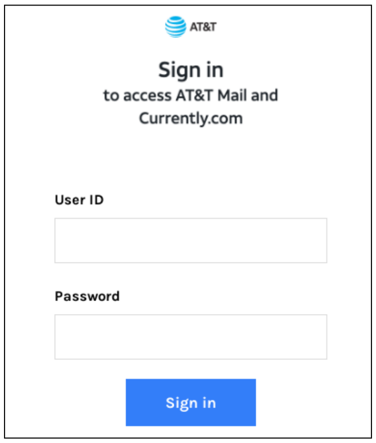
Female Male

By clicking Create an account, you agree to our Terms and that you have read our Data Policy, including our Cookie Use.

Create an account

Create a Page for a celebrity, band or business.

(f) Upper Case Conversion for First Letter


AT&T

Sign in

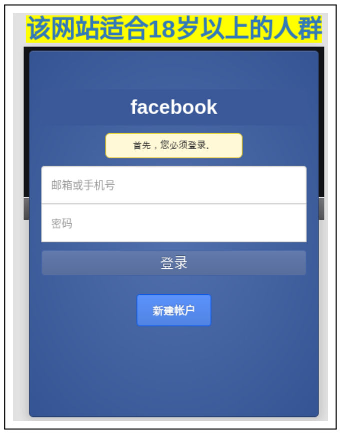
to access AT&T Mail and Currently.com

User ID

Password

Sign in

(g) PhishZoo Failed Example


该网站适合18岁以上的人群

facebook

首先，您必须登录.

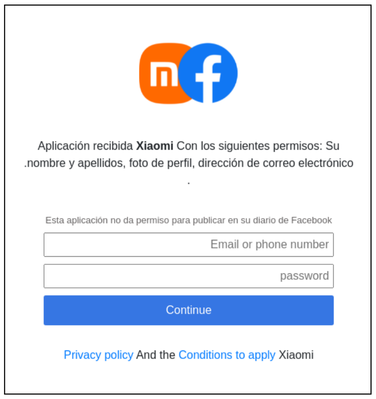
邮箱或手机号

密码

登录

新建帐户

(h) Adding Text


Aplicación recibida **Xiaomi** Con los siguientes permisos: Su .nombre y apellidos, foto de perfil, dirección de correo electrónico

.

Esta aplicación no da permiso para publicar en su diario de Facebook

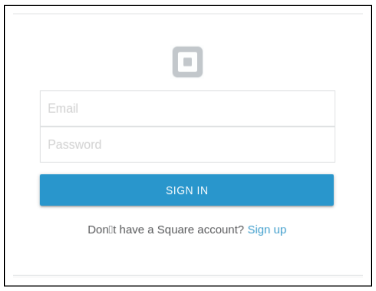
Email or phone number

password

Continue

Privacy policy And the Conditions to apply Xiaomi

(i) Integration of Logos

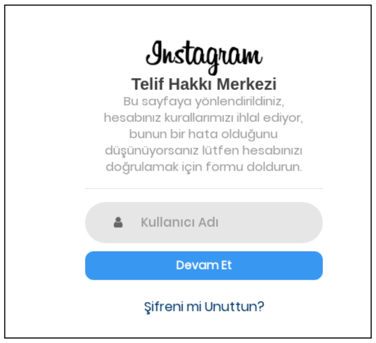

Email

Password

SIGN IN

Don't have a Square account? Sign up

(j) Omission and Color Replacement


Instagram

Telif Hakkı Merkezi

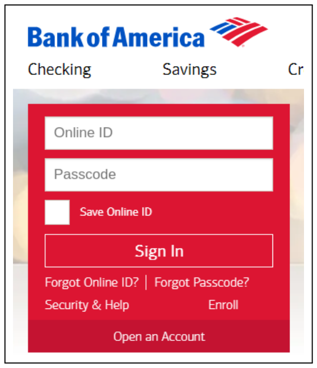
Bu sayfaya yönlendirildiniz, hesabınız kurallarımızı ihlal ediyor, bunun bir hata olduğunu düşünüyorsanız lütfen hesabınızı doğrulamak için formu doldurun.

Kullanıcı Adı

Devam Et

Şifreni mi Unuttun?

(k) Font Replacement


Bank of America

Checking Savings Cr

Online ID

Passcode

Save Online ID

Sign In

Forgot Online ID? | Forgot Passcode?

Security & Help Enroll

Open an Account

(l) Case Conversion and Outdated

Figure 6: Examples of Manipulated Samples Found in Our Real-world Phishing Dataset.