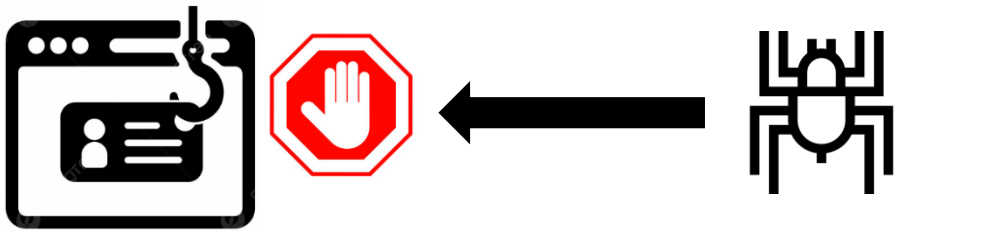
PhishDecloaker

Detecting CAPTCHA-cloaked Phishing Websites via Hybrid Vision-based Interactive Models

Xiwen Teoh, Yun Lin, Ruofan Liu, Zhiyong Huang, Jin Song Dong

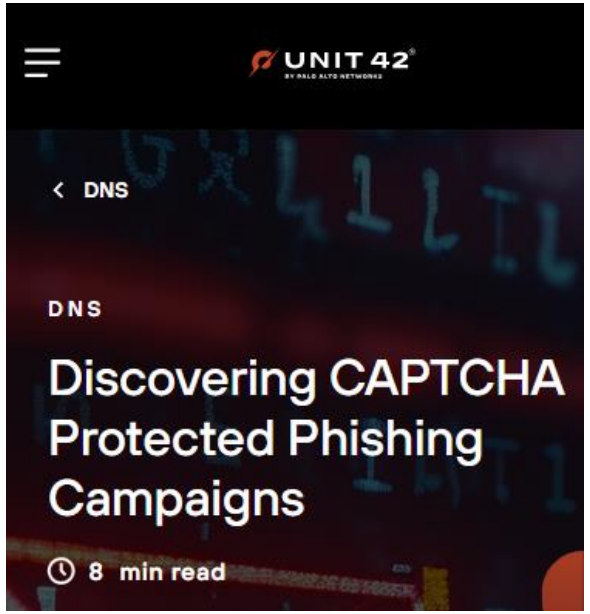
Introduction

- Phishing websites (*evaders*) and anti-phishing entities (*detectors*) are in an endless cat-and-mouse game
- Phishers use cloaking to deny access and evade detection
  - IP & User-Agent blacklist
  - One-time URLs
  - Browser fingerprinting



Introduction

- Recently, new trend: CAPTCHA-cloaked phishing
- This is reported by TrendMicro, Palo Alto Networks, AT&T, and many others

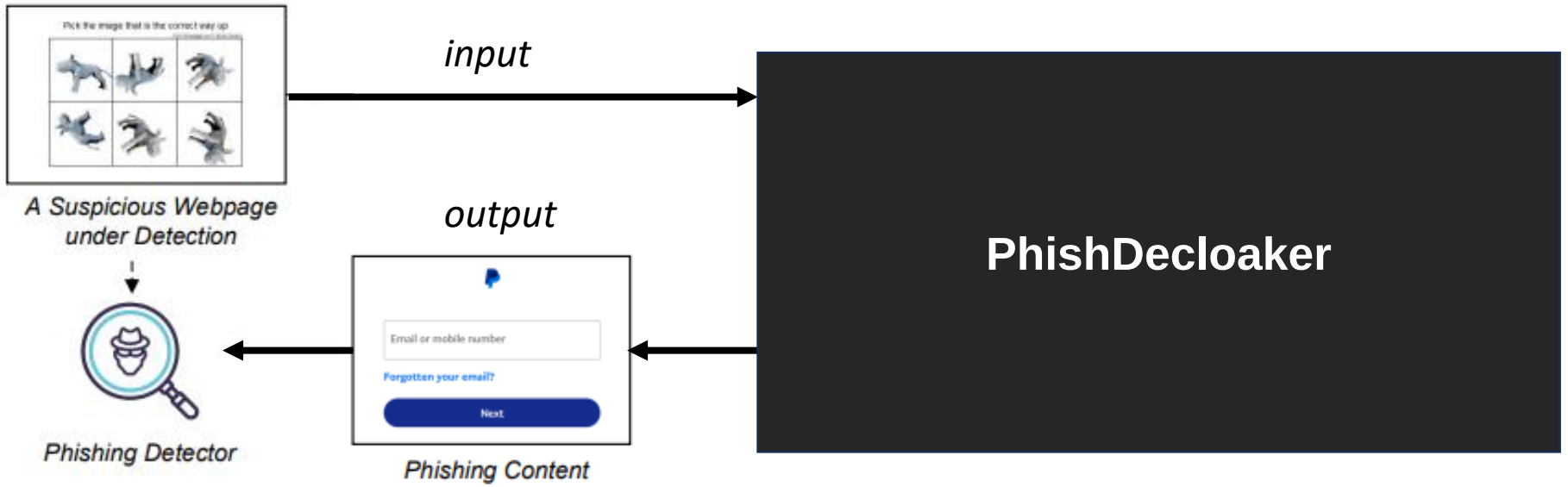


Introduction

- CAPTCHA-cloaking is problematic because:
  - It provides a false sense of legitimacy
    - Among top-1 million popular websites, 270k+ are using CAPTCHAs for common workflows (i.e., authentication)
  - It has low deployment cost
    - Many free or open-source CAPTCHA services (e.g., reCAPTCHAv2, hCaptcha) are readily available
  - It is hard to bypass
    - Our 7-day empirical study shows that *none* of our 500 CAPTCHA-cloaked phishing kits are detected by VirusTotal, Google Safe Browsing, Microsoft SmartScreen

Introduction

- PhishDecloaker…
  - Is a hybrid deep-vision system to automatically detect, recognize, and solve diverse CAPTCHAs on phishing pages
  - Once a phishing page is “decloaked”, pass it to the phishing detectors

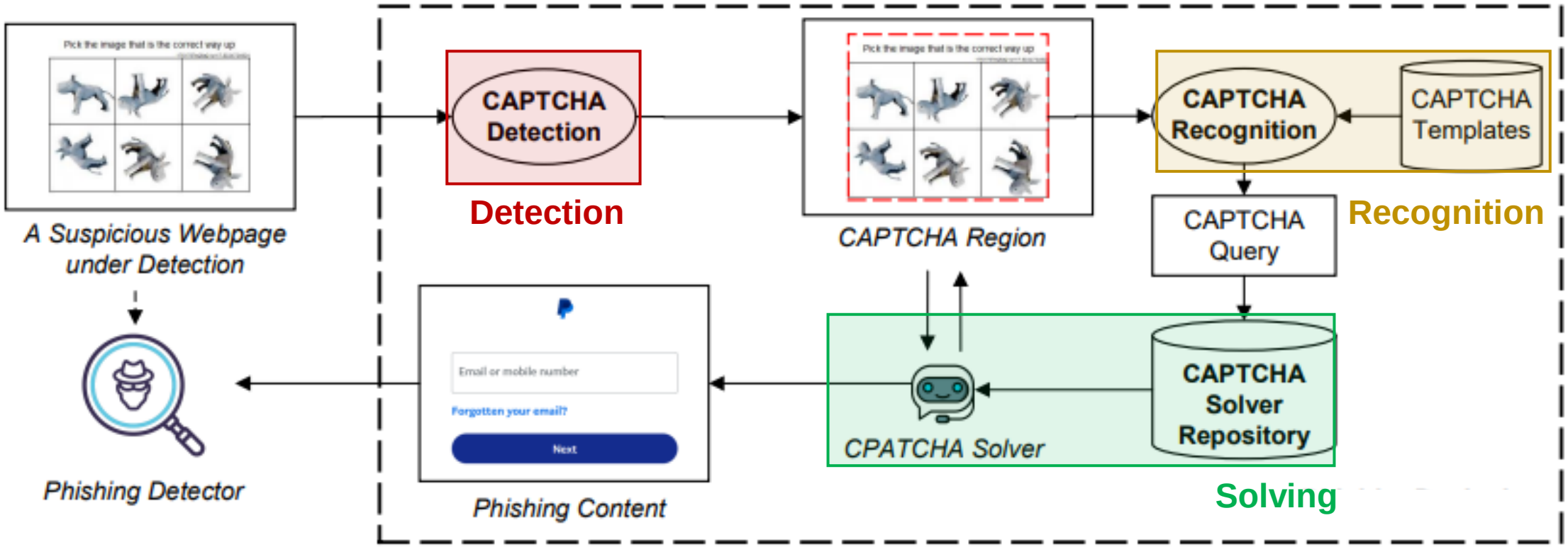


Approach

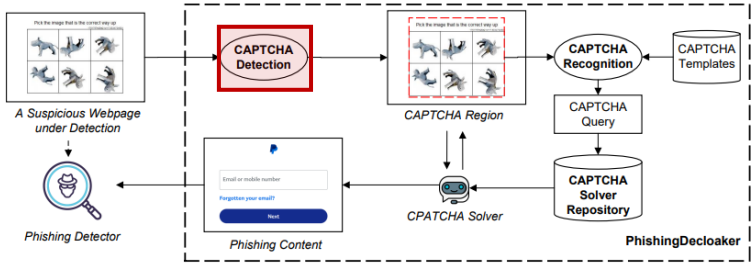
- PhishDecloaker adopts a 3-stage approach:
  - **Detection:** object detection, given a webpage screenshot, locate regions that are potentially CAPTCHAs
  - **Recognition:** classification, given a cropped region, identify the type of CAPTCHA present
  - **Solving:** browser automation, interact with the live page and complete the CAPTCHA challenge

Approach

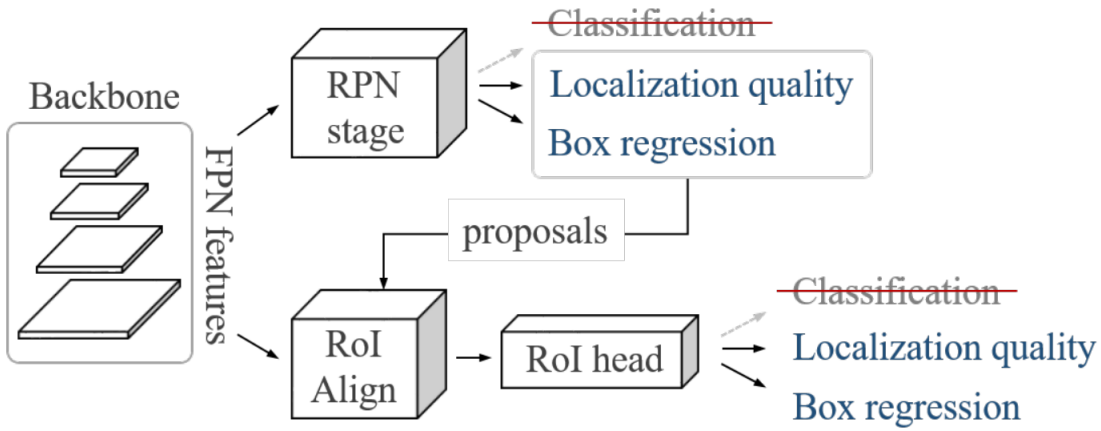
- PhishDecloaker adopts a 3-stage approach:



Approach

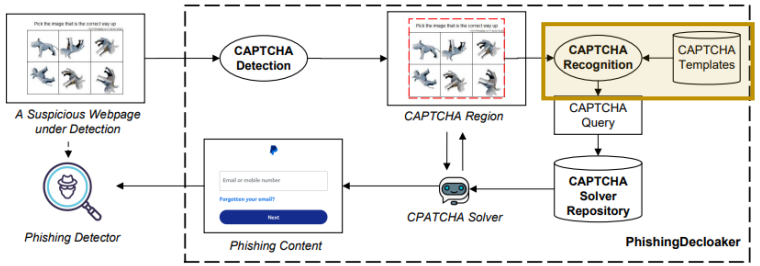


- **Detection**
  - Modified Faster-RCNN (a.k.a Object Localization Network [1])
  - Train with only localization & bbox regression loss (class agnostic)
  - Reasons:
    - Reduce overfitting to labeled objects
    - Learn stronger object cues
    - Achieve cross-category and cross-dataset generalization

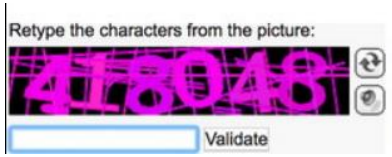


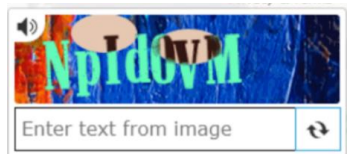
[1] Kim et al. *Learning open-world object proposals without learning to classify*. IEEE Robotics and Automation Letters, 7(2):5453–5460, 2022.

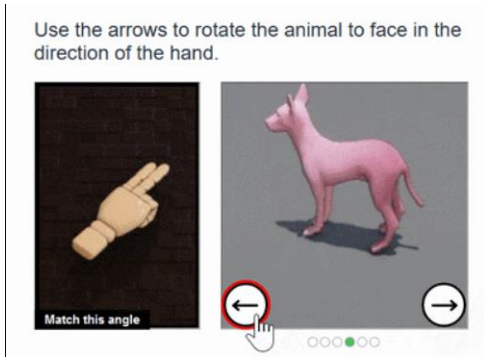
Approach



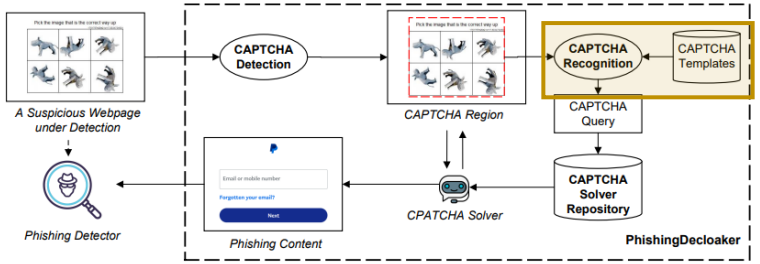
- **Recognition**
  - Design Considerations
    - Multi-modal representation learning
      - Challenge: CAPTCHA contains text and visual information
      - Solution: dual-branch architecture
    - Intra-type diversity
      - Challenge: handle same CAPTCHA type, but different challenge variants
      - Solution: metric learning with Sub-center ArcFace loss
    - Inter-type diversity
      - Challenge: handle new, unseen CAPTCHA types
      - Solution: Siamese model



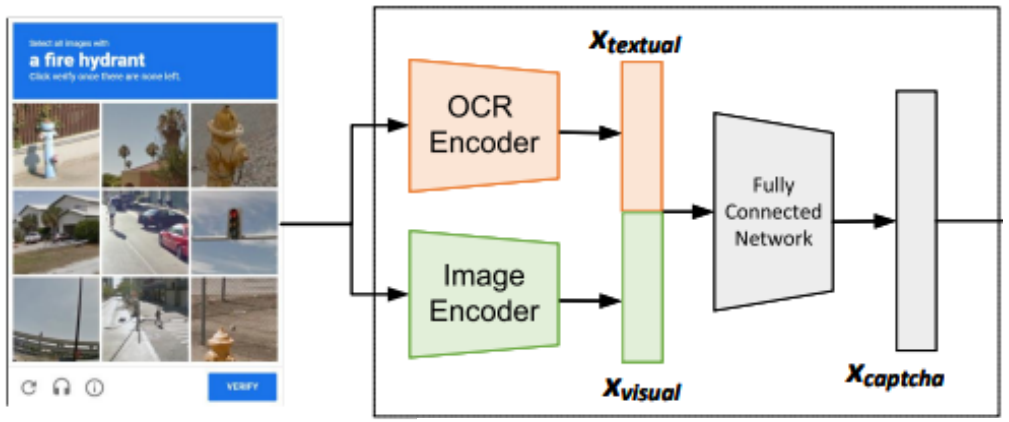




Approach



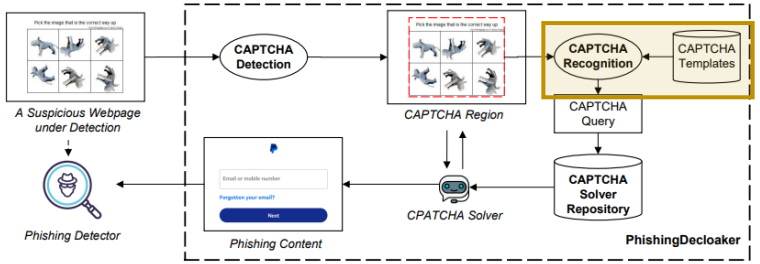
- **Recognition**
  - Deep Siamese model
  - Dual branch architecture: textual and visual features
  - Encode input images as n-dimension embeddings ($n = 512$)



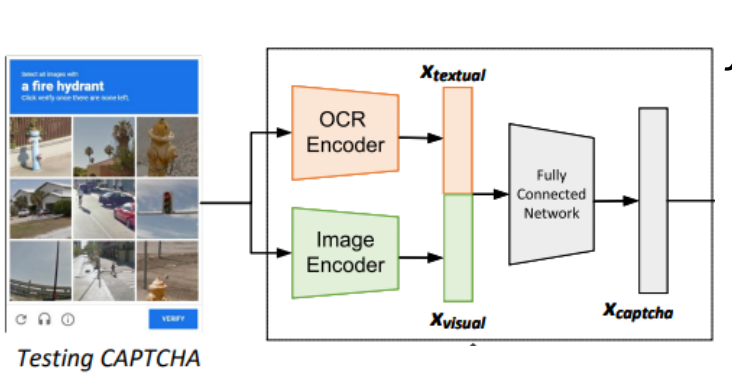
f_θ

Testing CAPTCHA

Approach

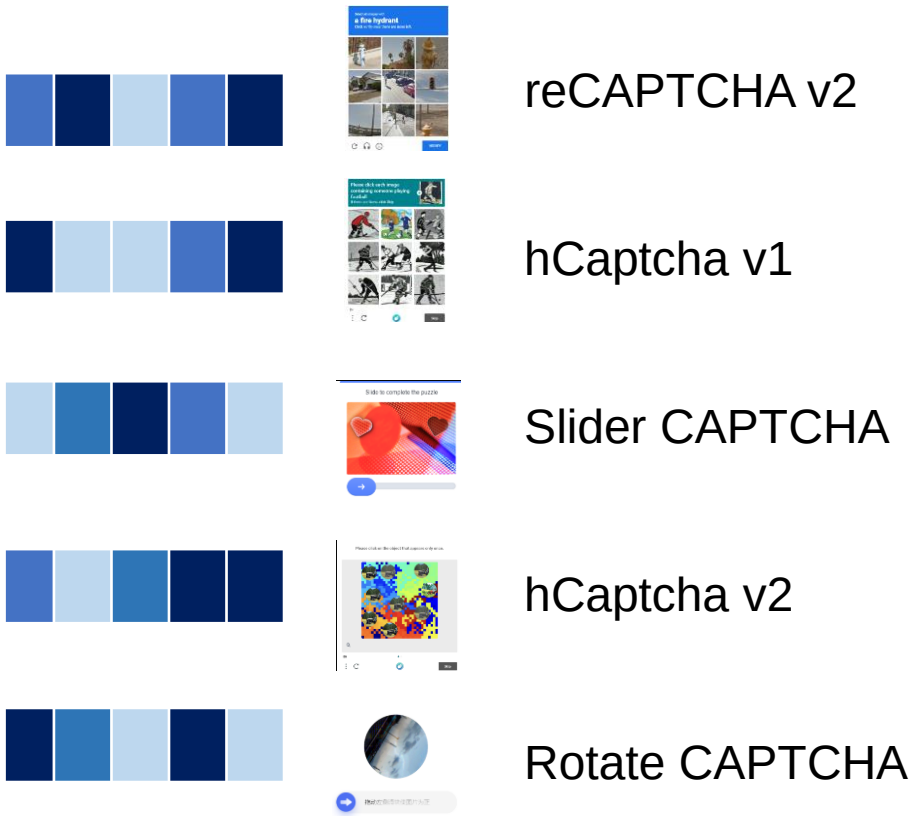


- **Recognition**
  - Classify CAPTCHA by comparing its embedding with a list of reference embeddings

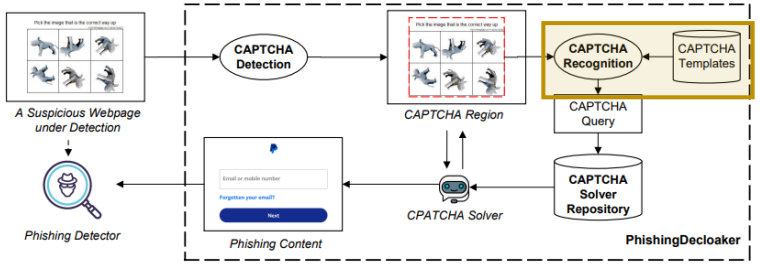


f_θ

$$c^* = \arg\max_{c_t \in C_t} cos(f_\theta(c), f_\theta(c_t))$$

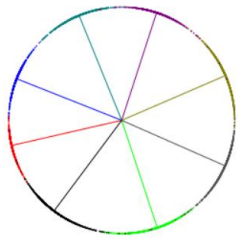


Approach

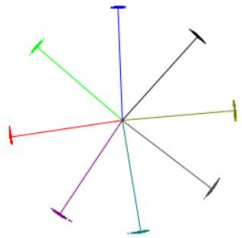


- **Recognition**
  - Train the model via metric learning
  - Pull positive pairs closer, negative pairs further in embedded space
  - Objective function: Sub-center ArcFace Loss

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{e^{s\cdot(\cos(\theta_{y_i}+m)-1)}}{e^{s\cdot(\cos(\theta_{y_i}+m)-1)}+\sum_{j=1,j\neq y_i}^{C}e^{s\cdot\cos(\theta_j)}}\right)$$



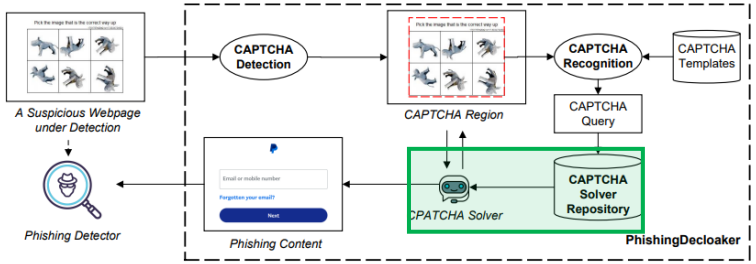
(a) Softmax



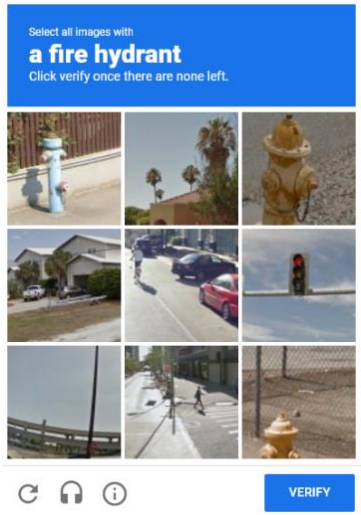
(b) ArcFace

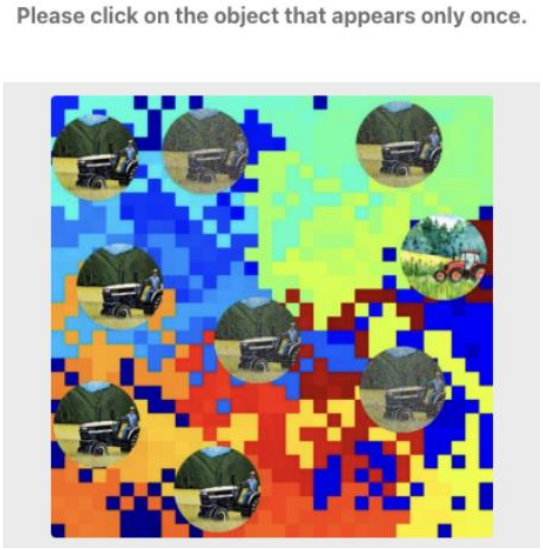
ArcFace: learned embeddings are distributed on a hypersphere with radius of s → clear decision boundary (inter-type diversity)
Sub-center: embeddings belonging to the same class can have multiple clusters (intra-type diversity)

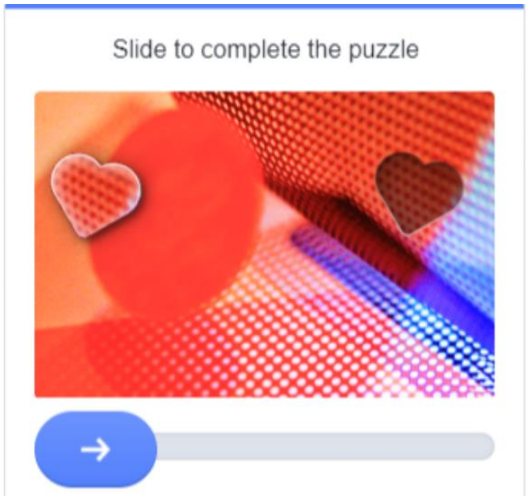
Approach

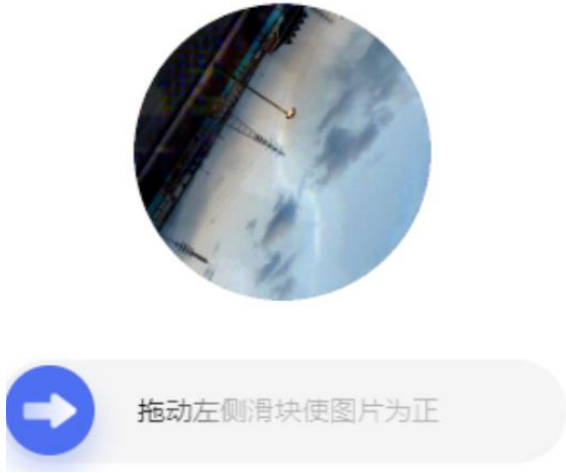


- **Solving**
  - 4 CAPTCHA types:
    - reCAPTCHAv2
    - hCaptcha
    - Slider-based
    - Rotation-based

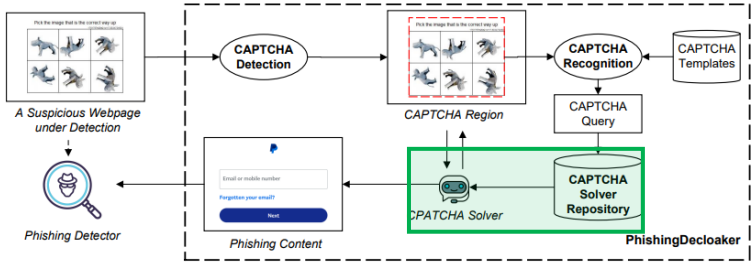






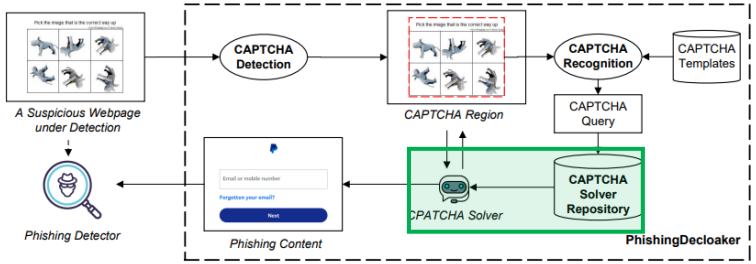


Approach

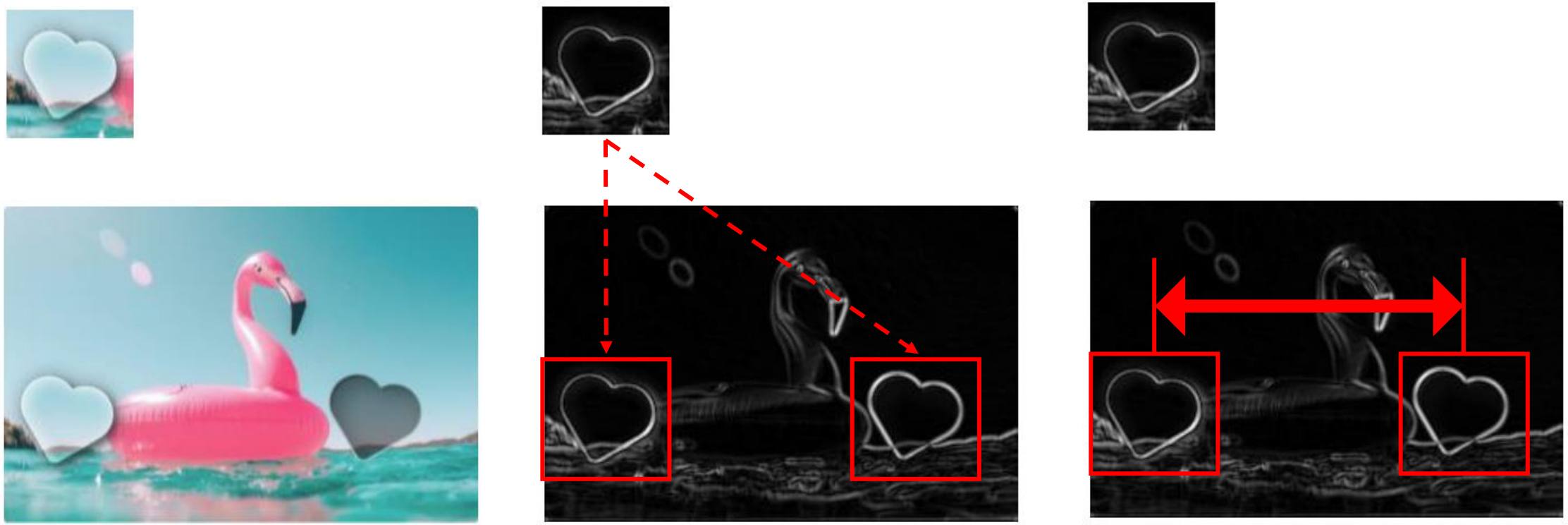


- **Solving**
  - reCAPTCHAv2 & hCaptcha solver: object detection

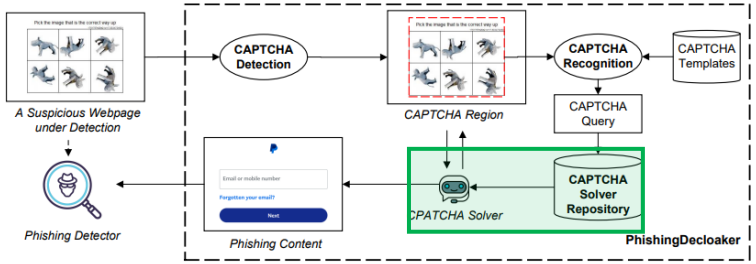
Approach



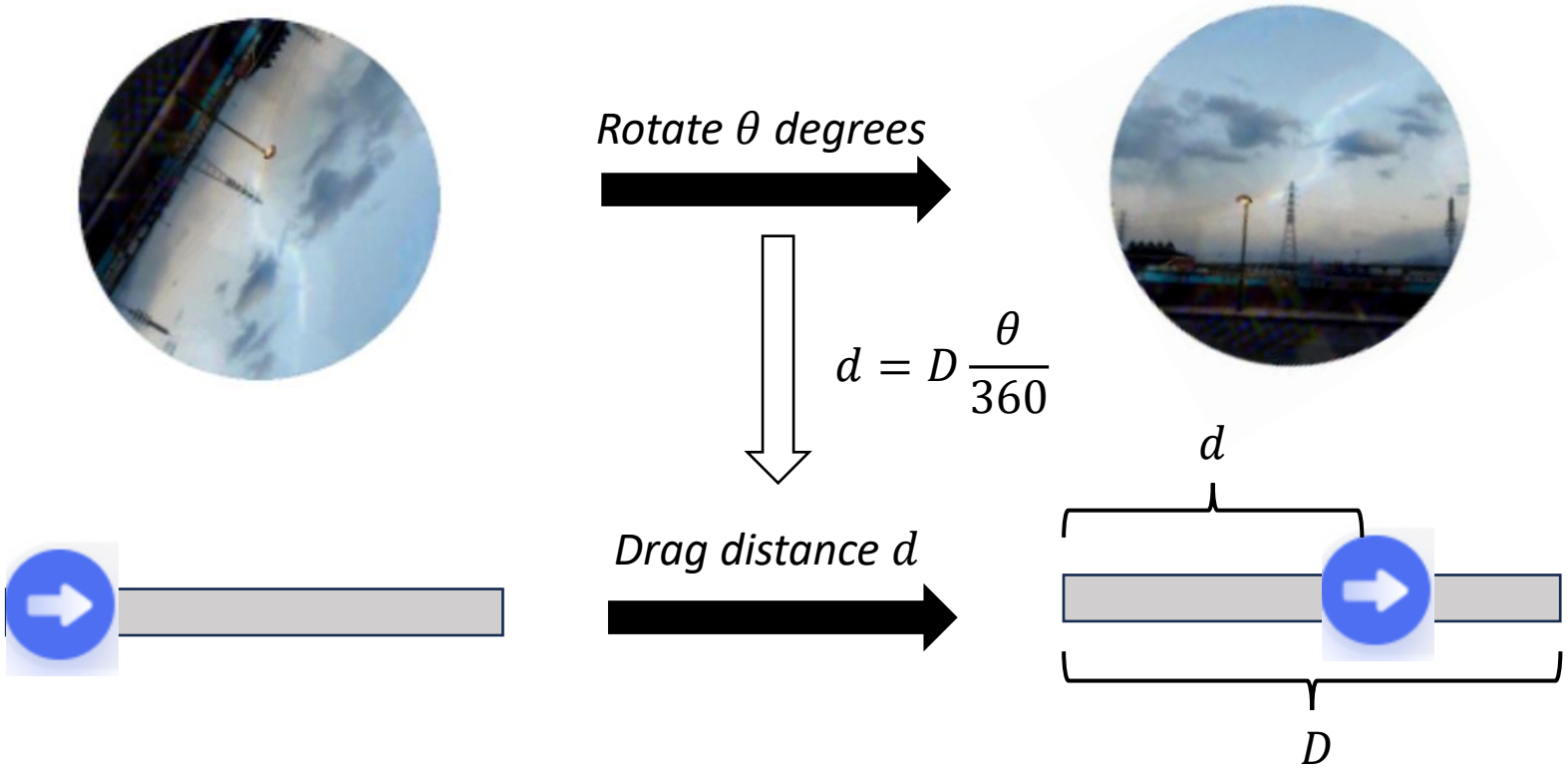
- **Solving**
  - Slider-based CAPTCHA solver: template matching



Approach



- **Solving**
  - Rotation-based CAPTCHA solver: regression



Field Study

- Can PhishDecloaker help discover more 0-day phishing websites in the wild? We prepared 6 study groups:

Group	Detector	JavaScript (JS)	Anti-Cloaking	Description
G1	PhishIntention	No	No	Control group
G2		Yes	No	JS rendering
G3		Yes	Anti-interaction-cloaking	Automatically closes popups, randomly moves and clicks mouse
G4		Yes	Anti-fingerprint-cloaking	Randomizes user agent and cookies, spoofs referrer, uses stealth headless browser
G5		Yes	Anti-behavior-cloaking	Follows redirects, waits and retries page loading up to 3 times
G6		Yes	Anti-CAPTCHA-cloaking	Uses PhishDecloaker to detect and solve CAPTCHAs

Field Study

- Experiment setup
  - Crawl new domains from Certstream (domains w/ new SSL certs)
  - Deploy the 6 study groups on the crawled domains

- Validation and monitoring
  - If a domain is reported as phishing by any group, we manually inspect the domain and track some metrics
    - **0-day:** a phishing website is 0-day if it is not reported by VirusTotal at the time of inspection
    - **Time-to-takedown:** time taken (hours) for site to go offline
    - **Time-to-blacklist:** time taken (hours) to be blacklisted by any of VirusTotal, Safe Browsing, or SmartScreen

Field Study

- Findings #1: PhishDecloaker's (G6) performance
  - Discovers **7.6%** more phishing websites not reported by any other study group
  - Captures the most 0-day phishing websites

Group	Setup	Unique Ratio	# 0-Days	# Phishing
G1	PI	0.0%	101 (–0.0%)	361 (–0.0%)
G2	PI + JS	0.0%	176 (↑74.3%)	582 (↑61.2%)
G3	PI + JS + AI	14.1%	197 (↑95.0%)	710 (↑96.7%)
G4	PI + JS + AF	0.0%	165 (↑63.4%)	543 (↑50.4%)
G5	PI + JS + AB	7.4%	198 (↑96.0%)	692 (↑91.7%)
G6	PI + JS + AC	10.2%	203 (↑101.0%)	648 (↑79.5%)

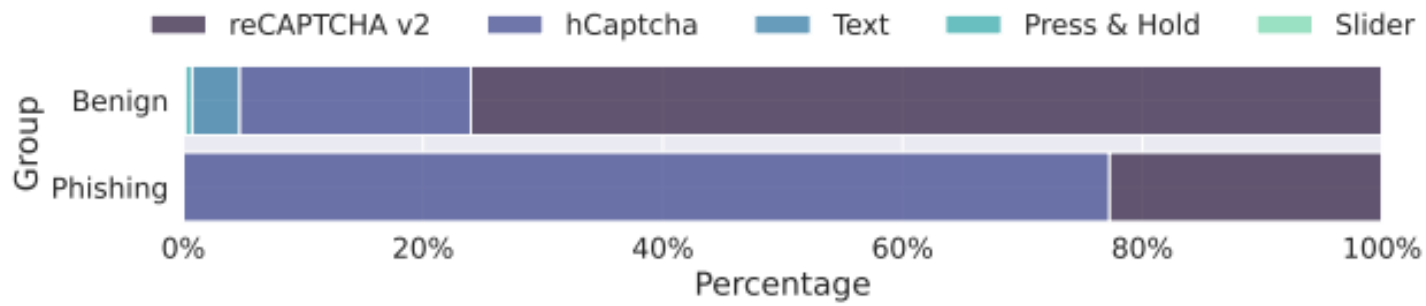
Field Study

- Findings #2: targeted sectors
  - Sectors targeted by CAPTCHA-cloaked phishing differs from ordinary phishing websites

Ordinary	%	CAPTCHA-Cloaked	%
Telecommunications	23.8	Cryptocurrency	43.9
Social Networking	22.8	Social Networking	19.3
Gambling	12.5	Logistics / Courier	15.8
Online Services / Software	12.3	Government Services	8.8
Financial / Insurance	10.1	Financial / Insurance	5.3

Field Study

- Findings #3: CAPTCHA types
  - Phishers tend to use *free* and *convenient* CAPTCHA services
  - Predominantly reCAPTCHAv2 (**22.7%**) and hCaptcha (**77.3%**)
  - Distribution differs from CAPTCHAs used by benign websites



Field Study

- Findings #4: CAPTCHA service API keys
  - These keys are extracted from CAPTCHA iframe in DOM
  - The distribution of API key usage is “roughly Pareto” — **fewer than 20%** of the API keys account for **more than 55%** of CAPTCHA-cloaking
  - For example, one hCaptcha API key was found to be reused across 19 different phishing websites.
  - Suggestion: as phishers reuse keys, they can be used as an Indicator of Compromise (IoC)

Field Study

- Findings #5: Phishing lifespan and time-to-blacklist
  - Surprisingly, CAPTCHA-cloaked phishing have a shorter lifespan compared to ordinary phishing (**9.7 vs 13.2 hours**)
  - However, it takes blacklist-based detectors **45.5% longer time** (11 hours) to register CAPTCHA-cloaked phishing as opposed to ordinary phishing.

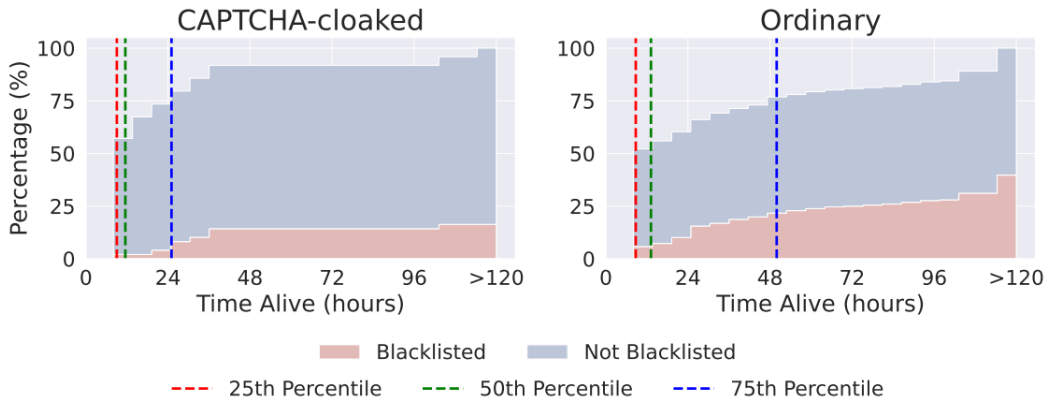


Figure 8: Cumulative distribution of life span for CAPTCHA-cloaked and ordinary phishing sites.

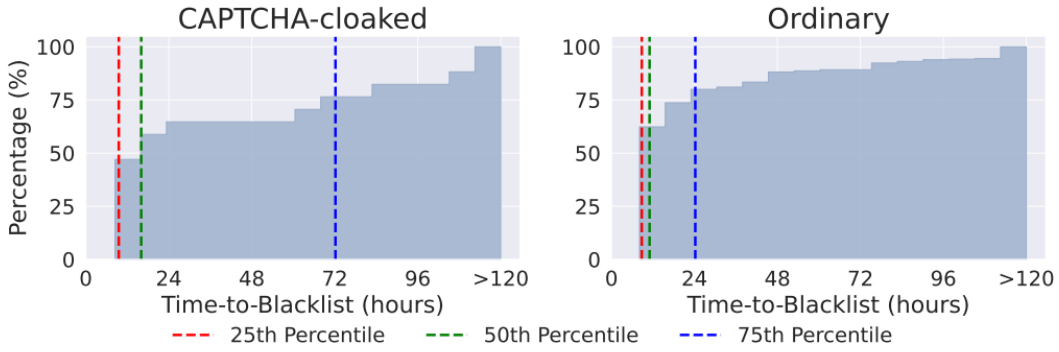
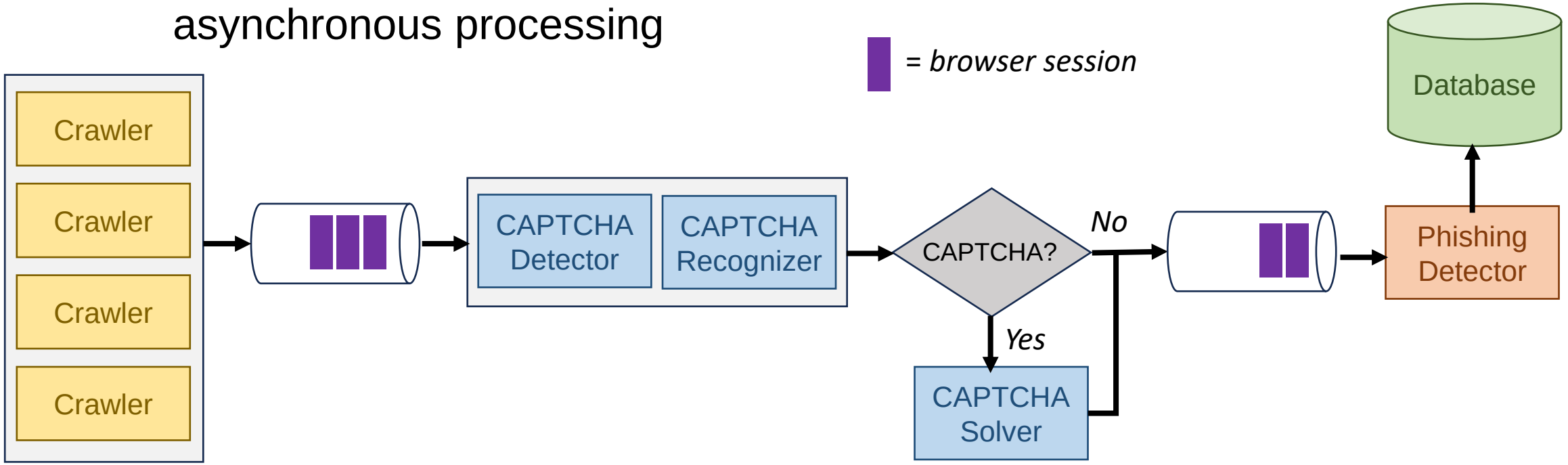


Figure 9: Cumulative distribution of time taken to be blacklisted by SmartScreen or GSB for CAPTCHA-cloaked and ordinary phishing sites.

Field Study

- Findings #6: Overhead
  - The median time of PhishDecloaker for detection, recognition and solving are 0.4s, 0.3s, 15.3s respectively
  - Long solving time can be mitigated by priority queues and asynchronous processing



Thank You!

- Questions:
  - e1374478@u.nus.edu (Xiwen Teoh)
  - lin_yun@sjtu.edu.cn (Prof. Yun Lin)

- Resources:
  - https://github.com/code-philia/PhishDecloaker (Codebase)
  - https://zenodo.org/records/11228974 (Datasets)
  - https://huggingface.co/code-philia/PhishDecloaker (Models)