Assessing Flaws in CAPTCHA Security through Progress in AI

Jaydon A. Stanislowski
stani152@morris.umn.edu
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

Abstract

Turing tests are widely employed on the Internet in the form of CAPTCHAs, short challenges designed to identify and prevent artificial web traffic. This technology is important for web security as a whole, and yet, as computational models become more sophisticated, its effectiveness only wanes. In this paper, we examine recent research on the growing threat to CAPTCHA security, in particular, the widely-used reCAPTCHA v2 and v3, and explore proposals to reinforce it against future attacks.

Keywords: CAPTCHA, reCAPTCHA, Turing test, artificial intelligence, neural networks, reinforcement learning, internet, security

1 Introduction

CAPTCHA, short for "Completely Automated Public Turing test to tell Computers and Humans Apart," is one of the most widely used security tools on the modern Internet, created to help mitigate bot traffic on websites. Maintaining the reliability of such tools is essential, as they defend millions of websites from spam, web scraping, credential stuffing (a type of attack against user login info that involves automatically testing several common or recycled passwords), and fake users pretending to be human actors.

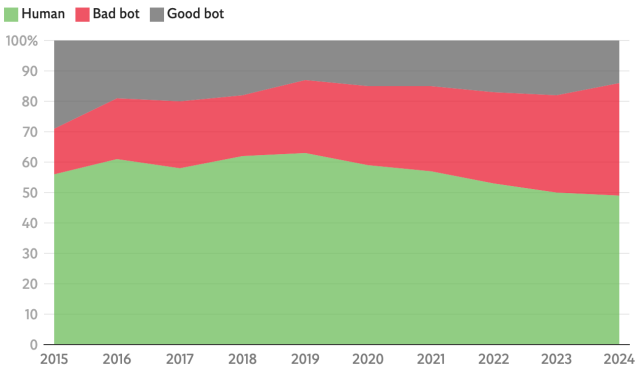

Figure 1: A graph of global internet traffic over recent years [1]

I don't really know if there's a better way to cite this in-line. This graph comes from an article by The Independent, but the data itself is from the 2025 Imperva Bad Bot Report.

In recent years, malicious artificial web traffic has increased substantially, with experts believing that bots accounted for over half of all web traffic in 2024, as shown in Figure 1. As AI technology has evolved over the years, so has the capacity for machines to simulate human intelligence, thereby bypassing security measures designed to impede them. AI models are now capable of performing a broad range of tasks previously thought to only be possible for humans to complete, further redefining what it takes to differentiate a computer from a real person. The increasingly sophisticated nature of these models poses a threat to the safety of the web as a whole, and yet, CAPTCHA development is seemingly unable to keep up.

This paper reviews the extent to which AI models are capable of bypassing modern CAPTCHA security, primarily focusing on the development of Google's reCAPTCHA as a case study. We will explain the computational models used in recent research to complete the tasks required by the most widely used CAPTCHAs today, and it will evaluate how successful the models are in this goal. The paper will also discuss predictions for how the threat to the Internet may grow with time, and explore recent proposals to improve existing tools to reinforce CAPTCHA security.

2 Background

2.1 Origin of CAPTCHAs

The term CAPTCHA was first conceived by Luis von Ahn et al. [2] in the early 2000's to describe a form of Turing test for the purpose of preventing spam and bot attacks on the Internet. The Turing test was originally designed by Alan Turing to assess machine intelligence by evaluating its ability to mimic human intelligence [3]. In the Turing test, a human evaluator holds a conversation with a machine and a human simultaneously, with the goal of identifying which is the human and which is the machine. According to Turing, if the machine is able to fool the evaluator, then it demonstrates an ability to exhibit intelligence similar to that of a human.

A CAPTCHA is slightly different from a Turing test in that, rather than a human evaluator and two participants, the test is administered automatically by a machine and taken by a single participant who may be a human or a bot. According to Luis von Ahn et al., "A CAPTCHA is a cryptographic protocol whose underlying hardness assumption is based on an AI problem" [2]. In order for a CAPTCHA program to be effective, it must be able to grade and provide tests that cannot be passed by current AI models, but are easy for humans to solve. The developers of CAPTCHA believed that, if a CAPTCHA challenge is successfully broken by AI, it "implies a win-win situation" because "a useful AI problem is solved," further advancing the field.

Figure 2: An example of a reCAPTCHA v1 challenge

Many such tests have been developed and used widely on the Internet over the past two decades. A well-known example is the first iteration of reCAPTCHA, known as reCAPTCHA v1. This type of CAPTCHA challenge is text-based, requiring the participant to transcribe a short string of text deemed difficult or impossible for machines to read, as shown in Figure 2. The text is often manipulated to further obscure from more sophisticated models, via warping, rotating, scaling, or the addition of random noise [4]. ReCAPTCHA v1 was deprecated in 2018, as the degree of complexity required to make it effective became too great, placing an unnecessary burden on human participants that defeated the purpose of CAPTCHA.

Since the development of these text-based CAPTCHAs, other types of challenges have dominated the web. In particular, the later versions of reCAPTCHA provide more sophisticated means of assessing participants while reducing friction for users. reCAPTCHA v2 and v3 simply "test" the participant by collecting hidden metrics about browser activity, discussed further in section 3.2. If the participant is scored to be a potential risk, they may be asked to solve another challenge, such as an image labeling problem, described in section 3.1.

Today, CAPTCHA challenges that analyze browser data have largely replaced both text- and image-based CAPTCHAs, although many websites still use image labeling challenges as a complete alternative or a second check should a user fail the former challenge. Despite the seeming effectiveness of combining these challenges, they are independently quite insecure, as discussed in section ??.

2.2 Artificial Intelligence

In order to understand the role of artificial intelligence in CAPTCHA security, it is helpful to first examine the details of the tools involved in cracking it.

2.2.1 Convolutional Neural Networks.

These AI models are typically employ to solve computer vision problems, or problems that task computers with identifying and analyzing objects or text in images. Due to their increasing effectiveness in recent years thanks to hardware developments, they are, to some extent, able to be trained and run on normal computers, posing a risk to web security applications that identify bots via image labeling problems.

Regular neural networks are a type of computational model based on human neurons, designed to simulate how humans learn sophisticated concepts and form mental associations. They consist of interconnected layers of nodes, where numerical data is transmitted between each node and manipulated via some activation function, whose parameters are determined by weights that get adjusted through the learning process. CNNS are unique in that they typically take an image as input, and compute an output that classifies the image via hidden (intermediate) layers that perform processes known as convolution and pooling.

Todo: Explain convolution and pooling in some detail, maybe just enough to span the rest of this page. This is still something that I'm very unfamiliar with—maybe having an extra meeting just to discuss and explain this would help.

2.2.2 Reinforcement Learning.

I haven't done sufficient research on reinforcement learning to write much here, but it is definitely a priority.

3 Modern reCAPTCHA Challenges

Somewhere here, I need to include something about how reCAPTCHA was acquired by Google for clarity, though I'm not sure where to slot that in.

As mentioned in section 2.1, modern versions of reCAPTCHA, namely, v2 and v3, primarily depend on collecting user metrics to assess the risk of bot activity, while also providing an image labeling challenge in alternative cases. This section will first discuss in detail how the image labeling challenges are created and served to the user, then touch on what is known about the metric-based challenges, according to existing research.

3.1 Image Labeling

Though less commonly used now, image labeling CAPTCHAs are featured in reCAPTCHA v2, designed for accessibility on mobile browsers as an alternative to the previous text challenges. In these challenges, the user is typically presented with a single image split into multiple segments, and must identify the segments of the image that constitute a particular object, such as a stop light or a car. Alternatively, the challenge may contain several individual images of objects, presenting the user with the same prompt.

Image labeling CAPTCHA challenges are easily defeated by CNNs, (...), **(there will be more information here about how reCAPTCHA v2 still largely relies on metrics collected from user input rather than strictly from successfully identifying image contents, as this is important to discussing flaws of the research and may form a case for CAPTCHA still being secure by combining methods)**

I think I will need to find better sources on reCAPTCHA v2 in particular, or just drop it from the paper entirely. I'm struggling with narrowing down my sources on this part to ones that specifically discuss the Google Street View type of image CAPTCHA and instead either focus on text-based ones or different computer vision problems, which isn't really helping to follow this flow.

3.2 Behavioral Metrics

Google's reCAPTCHA v2, first implemented in 2013, was the first tool to introduce an "invisible" CAPTCHA, a type of verification challenge that did not explicitly ask the user to solve a challenge to measure intelligence, but purely operated in the background and scored users based on browsing habits. This has been expanded on with reCAPTCHA v3, released in 2017. The tool regularly collects data as the user browses the web, returning a score ranging from 0 (highly likely to be a bot) to 1 (likely a human). While this type of challenge has been praised for reducing tedium resulting from CAPTCHA challenges for real users, many have criticized it for intruding on user privacy [5].

Not sure how relevant this last sentence is, or where to include it.

ReCAPTCHA v3 differs from v2's "invisible" challenge in that the user is never asked to check a box, but rather, it collects user data over time, and verification occurs either when the user clicks a button on a website that is bound to v3, or automatically invoked through some other action taken by the user. This verification is done through generating an encrypted token on the website's back end that contains the necessary metrics. This token is then sent to Google's servers, which return a response containing the user's score,

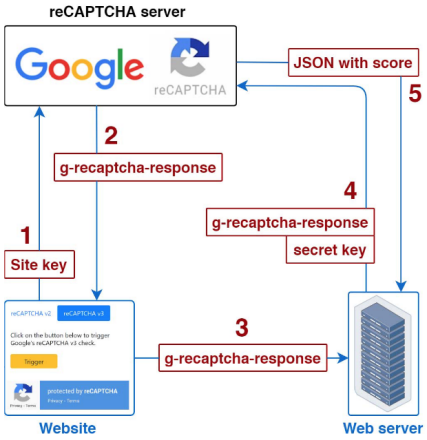

Figure 3: reCAPTCHA v3 verification workflow, adapted from [6]

whereupon the website can programmatically decide which actions to take. Figure 3 illustrates this process in detail [6].

Since reCAPTCHA v3 is proprietary, not much is known about exactly which metrics the tool collects, or how they contribute to the overall score. Additionally, keeping this information confidential is important to protecting the security of the software, as knowledge of it could theoretically be used to engineer a perfectly undetectable bot. However, Tsingenopoulos et al. [6] identified, through analysis of the obfuscated source code, that a few of the main factors contributing to the score may include mouse movements, keyboard inputs and timings, and the presence of cookies on the browser. The researchers further note that "security by obscurity is a fundamentally flawed approach," a common idiom in cryptography.

Again, I would like to insert this last sentence somewhere but doesn't really feel like a good spot exists.

4 Breaking Current CAPTCHA Schemes

The reCAPTCHA challenges discussed in section 3 are highly imperfect, and prone to attacks by well-constructed artificial intelligence models. This section will focus specifically on the work by **(??? for section 4.1)** and Tsingenopoulos et al. [6] to expose security vulnerabilities in the most widely used CAPTCHA schemes today.

4.1 reCAPTCHA v2 and CNNs

I hope to show here that convolutional neural networks are good enough at computer vision problems to complete image CAPTCHA challenges, particularly those associated with reCAPTCHA v2. It may be fine if I have to pivot here, but would make the story a bit less cohesive.

4.2 reCAPTCHA v3 and RL

Heavily focused on the methods of Tsingenopoulos et al. but need to read more on this and finish the section on reinforcement learning. May also be worth going over in a meeting.

5 Proposals to Improve CAPTCHAs

6 Conclusion

Acknowledgments

I sincerely thank Dr. Elena Machkasova for her invaluable work in not only instructing the fall 2025 senior seminar course, but also guiding me through the writing process and helping me stay motivated during this challenging semester. I also thank the University of Minnesota Morris for providing me the opportunity to conduct this literature review and present my work to my peers.