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

Protecting the internet from the threat of malicious bot activity is an important problem as AI tools become more powerful and commonplace over time. To that end, security measures are employed across websites in the form of CAPTCHAs, short challenges designed to identify and block fake web traffic. Yet, they become less effective over time as AI becomes more powerful, and thus more capable of solving them. This paper examines recent research on the threat to CAPTCHA security posed by current AI models and how this security can be reinforced over time, focusing primarily on Google's reCAPTCHA v3.

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

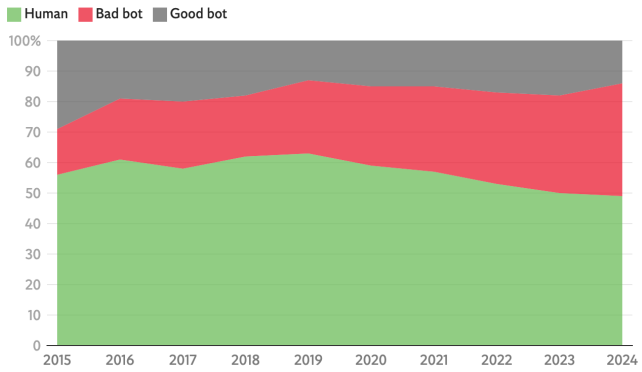

Figure 1: Area chart depicting the rise in artificial web traffic from 2015 to 2024 [4], adapted from [5]

1 Introduction

CAPTCHAs, short for "Completely Automated Public Turing tests to tell Computers and Humans Apart," are some of the most widely used security tools on the modern Internet, created to assist in preventing malicious bots on websites. Maintaining the reliability of these tools is important, since they defend millions of websites from a wide variety of threats, such as fake account creation, spam, extortion, web scraping, and credential stuffing[1].

In recent years, web traffic from malicious bots has increased substantially. Experts from cybersecurity company Imperva estimated that bots accounted for over half of all web traffic in 2024. The study distinguished "good bots," created to assist with tasks such as search indexing and site monitoring, from "bad bots," designed for harmful motives. These bad bots accounted for 37% of web traffic in 2024, while just 14% was produced by good bots, as shown in Figure 1.

As AI tools have evolved over time, so has their ability to simulate human behaviors often assessed by CAPTCHA challenges, making it easier to bypass their security. As a result, there is a clear need to assess the extent to which AI poses a threat to CAPTCHA tools. Recent research appears to show that current CAPTCHA tools may be ineffective for addressing this threat.

This paper reviews the extent to which AI models are capable of evading modern CAPTCHA security, primarily focusing on vulnerabilities in Google's reCAPTCHA v3 found by Tsingenopoulos et al. [12] as a case study. ReCAPTCHA v3 is widely used today and may provide insights into other CAPTCHA tools that behave similarly. Over one million websites as of September 2025 use reCAPTCHA v3, including X (formerly Twitter), GitHub, and StackOverflow, according to BuiltWith [3]. In the rest of this paper, Section 2 provides a general background of CAPTCHA's origins and how it has changed over time. Sections 3 and 4 respectively introduce reinforcement learning and reCAPTCHA v3, to discuss the research behind v3's security concerns in Section 5. Finally, Section 6 considers the study in the context of CAPTCHA security as a whole, followed by a conclusion.

2 CAPTCHA History

2.1 Origin of CAPTCHAs

The term CAPTCHA was first conceived by Luis von Ahn et al. [6] in the early 2000s 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 prominent qualities of human intelligence [13]. 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.

[1] An attack in which hackers gather databases of leaked user credentials and attempt to log into user accounts by setting up a bot to automatically test large batches of login information.

According to Turing, if the machine is able to fool the evaluator, then it demonstrates intelligence similar to that of a human, since it can replicate qualities associated with this.

A CAPTCHA is a variation of the Turing test. Rather than strictly focusing on natural language processing, CAPTCHAs address problems that are hard for AI models as a whole. This includes, but is not limited to tasks such as spatial reasoning, image recognition, and text/audio transcription. Because CAPTCHAs are automated, they are not administered by a human evaluator. Instead, the test is provided automatically by a machine and taken by a single participant who may be a human or a bot. The underlying assumption of CAPTCHA security is that it is based on the difficulty for AI to solve certain problems, similarly to how typical crytographic protocols rely on the computational complexity of certain mathematical problems [6].

In order for a CAPTCHA program to be effective, it must be able to provide and grade tests that cannot be passed by current AI models, but are easy for humans to solve. Furthermore, CAPTCHAs are secondarily designed to be a metric for the growth of artificial intelligence. Von Ahn et al. believed that, if a CAPTCHA challenge is successfully solved by an AI model with considerable consistency, it indicates that a useful AI problem has been solved, and a new type of CAPTCHA challenge must be administered [6].

Figure 2: An example of a reCAPTCHA v1 challenge. The challenge text reads “Levelers” and “critics.”

2.2 Text-based CAPTCHAs

Many implementations of CAPTCHA have been developed and used widely on the Internet over the past two decades. The earliest form of CAPTCHA was text-based, with one of the most well-known examples being the first version of reCAPTCHA, developed by Luis von Ahn and others [7] in 2007, then acquired by Google in 2009. ReCAPTCHA v1 poses a challenge 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.

Text-based challenges are no longer considered useful as a security measure, since they are easily bypassed by models trained against them. In particular, *convolutional neural networks* (CNNs) are effective at breaking text-based CAPTCHAs, since they are a type of neural network, a machine learning model inspired by how biological neurons learn. CNNs also operate on multi-dimensional data, such as images and videos. Furthermore, publicly available chat bots such as GPT-4 are capable of solving text-based CAPTCHA challenges with ease when prompted to do so [2]. Overall, the degree of complexity now required to make text-based CAPTCHAs unreadable to machines poses unnecessary difficulty to human participants, making them not only insecure but inefficient as well.

2.3 Modern CAPTCHAs

Since the development of text-based CAPTCHAs, other types of challenges have dominated the web. Some of these include image-based, audio-based, and spatial reasoning challenges. In particular, later versions of reCAPTCHA, developed after its acquisition by Google, provide more sophisticated means of assessing participants while reducing friction for users. ReCAPTCHA v2 contains image- and audio-based CAPTCHAs as part of its suite, while both v2 and v3 offer CAPTCHAs based on behavioral metrics. They reduce the burden of solving a challenge by collecting hidden data about the user’s browser activity in the background, rather than presenting a direct challenge. If the user is scored to be a potential risk, they may be prompted to solve a different challenge, or denied access to the website entirely. Today, CAPTCHAs using behavioral metrics have largely replaced both text- and image-based CAPTCHAs, although many websites still use various other types of challenges, especially when the former provides a low-confidence score. Behavioral metrics are discussed further in Sections 4 and 5.1.

3 Reinforcement Learning

Reinforcement learning (RL) is a paradigm in machine learning for training computational models that aim to interact with an environment by making decisions to further a goal. This concept has applications in a vast variety of fields. One example is in robotics, where a robot is given joints and limbs to move of its own accord, and may be tasked with developing motor control to walk a certain distance, or learn how to climb a ladder. Another application is in social media, where a model may be trained to recommend content to users via hashtags and engagement metrics.

In all of these examples, the computer in an RL system acts as an agent seeking to learn optimal decisions influenced by rewards and punishments. Figure 3 illustrates the dynamics of this system. The agent is either rewarded or punished for taking certain actions via a numeric score that represents its achievement. The agent accumulates this score over time through its decisions and states, seeking to maximize it however possible. Typically, the agent is initially unaware of which actions will lead it to a higher score, and makes these

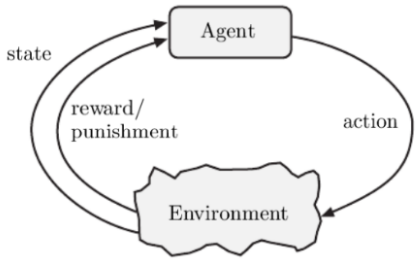

Figure 3: Block diagram modeling the basic principles of reinforcement learning [11]

choices randomly. The probability associated with taking each action, called a *policy*, is updated as the agent learns over iterations of this process.

Mathematically, an RL system is modeled by a Markov decision process (MDP). An MDP is generally a tuple, denoted (S, A, P, R). S and A respectively represent the set of states and actions that the agent can interface with, which may be discrete or continuous. P is a matrix that describes the probability of each state s leading to a new state s' via action a, with $s, s' \in S$ and $a \in A$. Elements of P are denoted as $P_a(s, s')$. Finally, R is a matrix describing the reward (or perceived reward) associated with taking action a to transition from s to s', with elements denoted as $R_a(s, s')$.

In order to learn, the agent follows a *policy function* $\pi(s)$, a mapping between states and actions that describes which actions to take depending on the agent's current state. The policy function is altered over time with the goal of finding a series of actions and states that maximizes the cumulative reward score.

In some use cases, the agent may not be able to observe the states of its environment, and thus be unable to act according to its current state. In such cases, the agent learns through a Partially Observable MDP (POMDP), where it acts according to its own belief (probability) of being in certain states instead, which is also updated over time.

The goal of solving a reCAPTCHA v3 challenge is to produce browsing habits that are convincingly human in nature, even when it is not known what behaviors exactly may contribute to this. Thus, an RL algorithm modeled by a POMDP can be used to bypass the security of reCAPTCHA v3. This is explored further in the research by Tsingenopoulos et al. [12], discussed in Section 5.

4 reCAPTCHA v3

As mentioned in section 2.3, the modern versions of reCAPTCHA, namely, v2 and v3, primarily base their security on collecting hidden metrics about a user's browsing activity to assess the risk of that user being a bot. This section will discuss how reCAPTCHA v3's "invisible CAPTCHA" challenge is implemented by web developers, with its inner workings obscured from the end user.

4.1 Workflow

ReCAPTCHA v3 is integrated as a third-party tool on a website, providing access to Google's API of the same name. Figure 4 illustrates the reCAPTCHA v3 workflow in detail. The tool is first configured to hook onto interactive HTML and JavaScript components of the site, such as a "checkout" button on a shopping cart, in order to silently collect analytics about how users typically engage with the site over time. The API is invoked programmatically when a user interacts with a website component that reCAPTCHA v3 is bound to, sending a site key (1) that identifies the website to the service. After considering the metrics, it returns a `g-recaptcha-response` token (2) to be verified by the website on the back end (3), in order to prevent the data from being affected by the client or end user. Once the website's server verifies the token with its own secret key (4) provided by Google, it receives a discrete score value in the set $\{0.1, 0.3, 0.7, 0.9\}$ to the site's back end (5) in JSON, a common file format for web data. A lower score means the user was deemed more likely to be a bot. The server can use this score to take any necessary actions, such as denying (or approving) the user's request, or prompting them to solve another challenge.

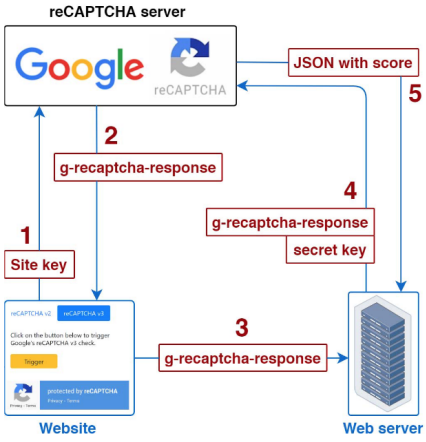

Figure 4: reCAPTCHA v3 verification workflow [12]

4.2 Considerations

Since reCAPTCHA v3 is proprietary, not much information is publicly known about exactly which metrics the tool collects in its analysis, 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, the researchers note that reCAPTCHA v2, which also integrated metrics challenges

in its suite, previously had its source code de-obfuscated and published to GitHub[2], allowing researchers to gain insights into how data was being processed and sent to Google [12]. Similar procedures are most likely employed in reCAPTCHA v3, making it easier to begin probing it for the same details.

Doing so is a more difficult task than it appears at first glance. Since the score returned by reCAPTCHA v3 is the only descriptive information available to the researchers, it is the only metric for judging progress towards breaking the scheme. The score itself has been indicated to contain noise through random variations in the tool's calculations, reflected in reCAPTCHA v2's source code, making it intentionally misleading and reducing its effectiveness for achieving this goal. Additionally, because the numeric output in the score is limited to one of only four numbers, as explained in Section 4.1, training AI using the score is a challenge. This is because a model may be misled by the limited information given by the score. However, the researchers address this using a policy-based approach, focusing on a POMDP model, described further in Section 3.

A final concern worth noting here is the application of Kerckhoff's principle, which argues that secrecy is a fundamental point of weakness in cryptographic encryption schemes, and that they must be secure even when the workings of their components are public knowledge, except the private key used for encryption [8]. It is possible to argue that this notion to security systems as a whole, including reCAPTCHA v3 — the "security by obscurity" approach of its design suggests that an attacker may be able to create a bot to perfectly evade detection should they have access to all or some of the exact methods used in the tool's calculations. However, it is important to clarify that Kerckhoff's principle in its literal sense does not apply to reCAPTCHA v3, because it is not a formal cryptographic protocol.

5 Threat Model

ReCAPTCHA v3 has potentially exploitable security flaws in its design, as discussed in Section 4.2, due to its "security by obscurity" approach to CAPTCHA challenges and its practice of providing a score to website owners that may hint at how confident the tool is of a user being a bot. This section will cover the research of Tsingenopoulos et al. [12] to expose these insecurities in reCAPTCHA v3's design by training an RL agent to mimic organic browsing habits using the score provided by reCAPTCHA v3.

5.1 Static and Dynamic Features

Section 4.2 noted that, prior to the research by Tsingenopoulos et al. [12], reCAPTCHA v2's source code was publicized, providing information about the metrics being collected by Google. Such metrics can be divided into two categories: *static* and *dynamic* features.

[2]https://github.com/neuroradiology/InsidereCAPTCHA

Static features encompass aspects of browsing activity that the user does not have immediate control over. This includes information such as the presence of cookies in the browser, as well as the User-Agent request header. This is part of the request sent to reCAPTCHA v3 containing information that identifies the user, such as their browser, operating system, IP address, and device model. These features are less accessible to the CAPTCHA service if the user modifies or spoofs their data, such as by using a VPN or a private browser.

Dynamic features, on the other hand, refer to aspects of browsing activity that the user directly controls over the duration of their browsing session. This primarily refers to activity such as mouse movements, keyboard inputs, request submissions, and input delay. While navigating websites, naive bots may be more likely to exhibit clearly inorganic activity through these dynamic features. For example, a bot might type in a box at a words-per-minute (WPM) rate that is impossible for humans to achieve, or move the mouse in a perfectly straight line towards a button, then click on it with little to no delay.

According to research done previously on reCAPTCHA v2, Google's metrics-based CAPTCHA challenges appear to be strongly biased in favor of users that exhibit strong static features, rewarding them with a higher score. In such cases where these CAPTCHAs are initially confident that the user is a human, dynamic features have a less noticeable impact on the score, even when the user exhibits dynamic features that are more bot-like in nature. In contrast, users that browse privately, with settings such as cookies disabled, they are initially more likely to be deemed a bot, and must prove their humanity through dynamic features. This is further supported by the results in Section 5.3. With this information, the researchers in [12] sought primarily to assess how effective RL was for evading metric-based CAPTCHAs using dynamic features alone, having a bot start from a low score and convince reCAPTCHA v3 that it was human through its browsing habits.

5.2 Methodology

The researchers used reinforcement learning agents in an environment modeled by a POMDP (described in Section 3). Since reCAPTCHA v3 operates using information that is not visible to the agents and can only be estimated, one can say that the agents are unable to observe the states of their environment and must learn by observations and assumptions instead. The score returned by reCAPTCHA v3 acts as an "oracle" in the model — the agents observe the decisions of reCAPTCHA v3 and make assumptions about how their actions might have produced those results, using them to learn even in a black-box environment.

Because dynamic features encompass many variables that are difficult to accurately measure and account for, the researchers developed multiple different modes of operation for their RL model, with varying degrees of control over their

inputs. The first agent, labeled *Piecewise*, was a rudimentary approach, only being able to control mouse movements towards a target component in a linear, piecewise fashion. The second agent, *Bezier*, was slightly more sophisticated, being able to control input delay and move the mouse in a curved fashion. The most complex agent, *Abstract*, was given the capabilities of the previous two, in addition to typing and scrolling.

The researchers conducted their experiments in a cascading manner, training and testing these agents on three unnamed websites. Website A was hosted by the researches, with the sole intention of hosting reCAPTCHA v3 and training their rudimentary models. *Piecewise* was both trained and tested on this website, while *Bezier* was only trained.

Website B was one that saw real web traffic, receiving hundreds of daily requests from users, and configured reCAPTCHA v3 with the consent of the researchers, giving them access to the score provided on the back end. This was used to test *Bezier* and train *Abstract*.

Finally, Website C was a much larger website that received thousands of daily requests, had already fully integrated reCAPTCHA v3 without assisting the researchers, and only allowing them to observe whether the agent's requests were blocked. This website was used only to test *Abstract*, with the intention of placing it in a typical user scenario, a black-box environment with no information about reCAPTCHA v3's evaluations.

5.3 Results

Data from the research supports the hypothesis that RL can be employed to bypass the tool's security with reasonable consistency by learning from the score alone, even when initially flagged as a potential bot.

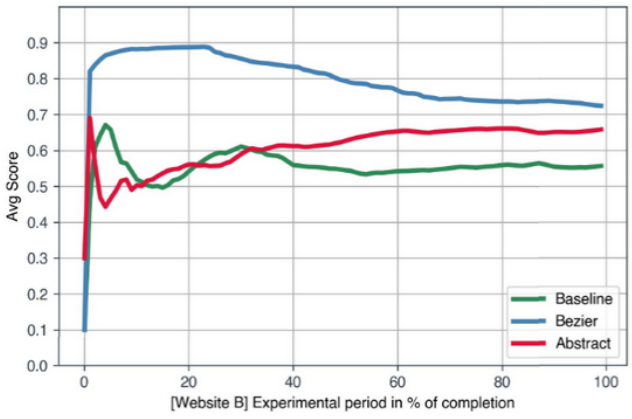

Figure 5: Average cumulative score over the period of evaluation for Website B [12]

Figure 5 depicts the progress of both the *Bezier* and *Abstract* agents over time on Website B. Since *Bezier* was in testing mode rather than learning, its parameters were no longer being updated, and therefore likely saw considerable score depletion over time due to repetitive inputs identified by reCAPTCHA v3, although the agent still performed well out of the box. Meanwhile, *Abstract* demonstrated considerable progress in its learning over time, improving from its initial average score of around 0.45 to around 0.65, making it promising for testing. The researchers also deployed a baseline (naive) agent, distinct from the *Piecewise* agent, that did not learn and only acted as a typical bot. While this naive approach performed worse than the RL-based approaches, it still consistently achieved an average score above 0.5 on Website B, which is the default threshold set by reCAPTCHA v3 to distinguish bots from humans. This is particularly concerning, as it implies that even a naive approach is sufficient to bypass the tool when static features are present in the browser.

	Baseline	*Abstract*
S_H	84.3%	99.6%
S_L	20.8%	70.1%

Table 1: Evasion Rates for Website C [12]

On Website A, used to test the *Abstract* agent, the researchers did not have access to the score provided by reCAPTCHA v3, as they were not responsible for hosting the tool. To that end, the agent was assessed on its evasion rate instead — whether or not it successfully fooled reCAPTCHA v3. The results, depicted in Table 1, are striking. S_H denotes high starting sessions, or browsing sessions where the agent was initially flagged as a human based on the static data in the User-Agent request header, while S_L denotes low starting sessions. Across low-starting sessions, there is a clear gap between the naive algorithm and the *Abstract* agent's performance, with *Abstract* passing as a human across the majority of browser requests. Its performance only improved further in high-starting sessions, almost perfectly evading detection in the vast majority of cases.

6 Analysis

6.1 Implications

The results from [12] show a vulnerability in reCAPTCHA v3, largely due to the score value that is provided to website owners when user requests are made. Using this score, the researchers were able to train an RL model capable of mimicking human browsing habits without additional feedback. This suggests that CAPTCHAs using behavioral metrics may be insecure if they rely on such a quantifiable way of determining bot traffic. While this score value is an essential feature of reCAPTCHA v3, not all invisible CAPTCHA challenges provide a score. Some, such as Cloudflare's Turnstile, send only a Boolean "pass" or "fail" response to websites.

However, a measurable CAPTCHA score may come with its own advantages over a simple "pass" or "fail." The score allows website owners to control how potential unwanted traffic is handled, especially considering that the default threshold of 0.5 may not be sufficient, a claim supported by Figure 5 and Table 1. If a similar metric is used by CAPTCHA frameworks that invariably determine this threshold for the website, it limits how strict the tool can be, potentially allowing more unwanted traffic, rather than preventing it.

Additionally, any behavioral metrics CAPTCHAs may be at risk, regardless of whether or not they provide website owners with a score. If an RL model can be trained to bypass reCAPTCHA v3 with considerable consistency, particularly when static features contribute strongly to the score, the same model can likely be deployed on other websites using different CAPTCHA tools taking similar metrics. In fact, later research conducted by Sateur et al. [10] demonstrated this with Cloudflare's Turnstile.

6.2 Limitations

While the findings in Section 5 are striking, the research is not without its limitations. Since the study focuses almost exclusively on reCAPTCHA v3, it is difficult to exactly determine how it may impact other CAPTCHA tools. However, behavioral metrics-based CAPTCHAs largely work on the same principles of measuring static and dynamic features, so the possibility of using a similar RL model to break other tools seems promising for future studies.

As mentioned in Section 4.2, reCAPTCHA v3 likely introduces random noise in its score calculation to deter machine learning. This means that the true meaning of the score for assessing the RL agent's progress is less clear than it would be otherwise, and it is a highly imperfect metric for knowing what the tool is measuring behind the scenes. It is likely that the RL agents in the study still could have learned more and achieved greater evasion consistency, however, they were limited quite heavily in their activity, and could not remain active for long periods of time. This is because overly frequent or repetitive requests are known to have a strong negative impact on reCAPTCHA v3's score, as attackers commonly do this to send spam or overload the website's servers with fake activity.

Finally, the discussion here would benefit from further studies observing how reCAPTCHA v3 has changed over time, particularly because the tool is, itself, evolutionary in nature. Google is known to collect user data from reCAPTCHA v3 to improve the service, likely using it to train a "defender" model that seeks to correctly identify bot traffic, through a process known as adversarial learning. Since reCAPTCHA v3 changes over time, it is difficult to determine how exactly the results of [12], conducted in 2022, would be applicable against the tool today. However, it is worth noting that the tool was previously studied using a similar approach, with similar alarming results achieved [1].

6.3 Ethical Considerations

An important part of ethical hacking is responsible disclosure, addressing newly-discovered security concerns by notifying the developers of the security feature at risk, and ensuring that a proper response is received or a fix is implemented before the vulnerability is made known to the public. Tsingenopoulous et al. [12] made sure Google was aware of the risks of reCAPTCHA v3 and their RL model by opening a support ticket. The issue was marked as "intended behavior/won't fix," likely due to how reCAPTCHA v3 is designed to be able to combat these threats over time, as mentioned in Section 6.2.

Behavioral metrics-based CAPTCHAs have been criticized by consumers for being invasive of user privacy, with the tools often being likened to spyware. To give users more rights to how their personal data is used online, the European Union passed the General Data Protection Regulation (GDPR) in 2018, which requires websites to obtain user consent before using cookies or tracking any data. ReCAPTCHA v3 is compliant with the GDPR. Google claims that the data collected by the service is not used for advertising, and is only used for providing and maintaining the service [9].

7 Conclusion

CAPTCHA tools are an integral yet often overlooked part of web security, as they protect millions of websites from malicious automated traffic in a time dominated by the development of AI. Despite this, they face scrutiny today, with mounting public concern regarding their usefulness. This paper investigated current forms of CAPTCHA that rely on behavior metrics and assessed their effectiveness against developments in AI tools, and found that they were, in their current form, somewhat lacking in necessary security quality. Yet still, being a tool designed to evolve over time, CAPTCHAs may continue to change and lead efforts to protect the internet, while also being invaluable for measuring progress in the rapidly-growing field of AI. Because of this, it is important to be aware of how they could be improved for the future through techniques such as adversarial learning, or integrating more complex yet non-invasive CAPTCHA challenges, such as interactive puzzles.

Acknowledgments

I sincerely thank Dr. Elena Machkasova for her work in not only instructing the Fall 2025 Senior Seminar course, but also guiding me through this process. I also thank Melissa Helgeson, Pierce Richards and Peter Dolan for providing helpful feedback on this paper through the drafting process, and I appreciate the University of Minnesota Morris for providing me the opportunity to conduct this work and present it to my peers.

References

[1] Ismail Akrout, Amal Feriani, and Mohamed Akrout. 2019. Hacking Google reCAPTCHA v3 using Reinforcement Learning. arXiv:1903.01003 [cs.LG] https://arxiv.org/abs/1903.01003

[2] Pieter Arntz. 2025. ChatGPT solves CAPTCHAs if you tell it they're fake. https://www.malwarebytes.com/blog/news/2025/09/chatgpt-solves-captchas-if-you-tell-it-theyre-fake

[3] BuiltWith. 2025. ReCAPTCHA v3 Usage Statistics. https://trends.builtwith.com/widgets/reCAPTCHA-v3

[4] Anthony Cuthbertson. 2025. Most internet traffic is now bots. https://www.the-independent.com/tech/bots-internet-traffic-ai-chatgpt-b2733450.html

[5] Imperva. 2025. *2025 Bad Bot Report*. Technical Report. Imperva. https://www.imperva.com/resources/resource-library/reports/2025-bad-bot-report/

[6] Luis von Ahn, Nicholas J. Hopper, Manuel Blum, and John Langford. 2003. CAPTCHA: Using Hard AI Problems for Security. In *Advances in Cryptology — EUROCRYPT 2003*. Springer Berlin Heidelberg, 294–311.

[7] Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. 2008. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science* 321, 5895 (2008), 1465–1468. arXiv:https://www.science.org/doi/pdf/10.1126/science.1160379 doi:10.1126/science.1160379

[8] Fabien A. P. Petitcolas. 2011. *Kerckhoffs' Principle*. Springer US, Boston, MA, 675–675. doi:10.1007/978-1-4419-5906-5_487

[9] Badr Salmi and David Lenehan. 2023. reCAPTCHA Enterprise and the importance of GDPR compliance. https://cloud.google.com/blog/products/identity-security/recaptcha-enterprise-and-the-importance-of-gdpr-compliance/

[10] Maxime Sateur, Javier Martínez Llamas, Davy Preuveneers, and Wouter Joosen. 2025. Evaluating Turnstile as a Privacy-Conscious Alternative to reCAPTCHA. In *Availability, Reliability and Security*, Bart Coppens, Bruno Volckaert, Vincent Naessens, and Bjorn De Sutter (Eds.). Springer Nature Switzerland, Cham, 235–252.

[11] Hamid R. Tizhoosh and Graham W. Taylor. 2006. Reinforced Contrast Adaptation. *International Journal of Image and Graphics* 06, 03 (2006), 377–392. doi:10.1142/S0219467806002379

[12] Ilias Tsingenopoulos, Davy Preuveneers, Lieven Desmet, and Wouter Joosen. 2022. Captcha me if you can: Imitation Games with Reinforcement Learning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroSP)*. 719–735. doi:10.1109/EuroSP53844.2022.00050

[13] Alan Turing. 1950. Computing Machinery and Intelligence. *Mind* 59, 236 (10 1950), 433–460. arXiv:https://academic.oup.com/mind/article-pdf/LIX/236/433/61209000/mind