Universidade de São Paulo
Instituto de Matemática e Estatística
Bacharelado em Ciência da Computação

Segurança de *CAPTCHAs* além da obscuridade

Carybé Gonçalves Silva

Monografia Final

mac 499— Trabalho de Formatura Supervisionado

Supervisor: Prof. Dr. Ronaldo Fumio Hashimoto

São Paulo
2022

Ao meu orientador, por sua paciência infinita e todo suporte,
à minha mãe e irmã pela força quando mais precisei,
à minha namorada por todo o carinho e apoio,
e ao meu pai, minha eterna inspiração.

Resumo

Carybé Gonçalves Silva. **Segurança de *CAPTCHAs* além da obscuridade**. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2023.

*CAPTCHA*s (*Completely Automated Public Turing test to tell Computers and Humans Apart*), se baseiam na apresentação de desafios que devem ser facilmente realizados por humanos e difíceis de serem feitos por máquinas; são usados para restringir acesso a conteúdo sensível ou controlar atividades em um ambiente computacional. Por outro lado, muitos *CAPTCHA*s são desenvolvidos sem se atentarem a seus princípios básicos de segurança, discretamente pressupondo o desconhecimento do atacante sobre seu funcionamento. Nesse trabalho explora-se o que acontece se essa pressuposição for questionada e é aplicado um ataque baseado na capacidade de reconhecimento de padrões de Redes Neurais a um modelo de *CAPTCHA* textual que não parte desse pressuposto.

Palavras-chave: Segurança. Prova de interação humana. Redes neurais. Obscuridade. Taxonomia.

Abstract

Carybé Gonçalves Silva. ***CAPTCHAs' security beyond obscurity***. Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2023.

CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart), are based on presenting challenges that should be easily faced by humans and difficult for machines to crack. They are used to restrict access to sensitive content or control activities in a computing environment. On the other hand, many CAPTCHAs are developed without paying attention to their basic security principles, quietly assuming the attacker's ignorance of the systems workings. This paper explores what happens if this assumption is questioned and an attack based on the pattern recognition capability of Neural Networks is applied to a textual CAPTCHA model that does not make such assumption.

Keywords: Security. Human interaction proof. Neural Networks. Obscurity. Taxonomy.

Lista de Figuras

3.1 1ª Geração de Captchas textuais 8
3.2 Evolução do re*CAPTCHA* 9
3.3 Diagrama de Taxonomias por categorias de Desafio 11

B.1 Generalizações com n=2 27
B.2 Generalizações com n=4 28
B.3 Generalizações com n=8 28
B.4 Generalizações com n=16 29

Lista de Tabelas

3.1 Categorias com respectivos exemplos 12

4.1 Características de Segurança 16
4.2 Taxa de acerto para o *ConfirmEdit* 19
4.3 Taxas de acerto para diferentes extensões de texto (n) 19

A.1 Taxonomia de CAPTCHAS 23

Sumário

1 Introdução **1**

1.1 Contextualização . . . 1

1.2 Objetivo . . . 1

1.3 Motivação . . . 1

2 Segurança **3**

2.1 Segurança por obscuridade e o Princípio de Kerckhoffs . . . 3

2.1.1 Modelos de ameaça . . . 4

2.2 Reflexão . . . 4

3 *CAPTCHA*s **7**

3.1 O que são *CAPTCHA*s? . . . 7

3.1.1 Histórico e Exemplos . . . 7

3.2 Taxonomia dos *CAPTCHAS*s . . . 9

3.2.1 Metodologia . . . 9

3.2.2 Desenvolvimento . . . 9

3.3 *CAPTCHA*s como sistemas de segurança . . . 13

3.3.1 Considerações econômicas . . . 13

3.4 Resultados . . . 14

4 Prova de Conceito **15**

4.1 Metodologia . . . 15

4.1.1 Características de Segurança . . . 16

4.1.2 Outras variações exploradas . . . 16

4.2 Desenvolvimento . . . 17

4.2.1 Recursos computacionais em números . . . 17

4.2.2 Tecnologias . . . 18

4.2.3 Desafios . . . 18

4.3 Resultados . . . 19

5 Conclusão **21**
5.1 Trabalhos Futuros 21

Apêndices

A Taxonomia **23**

B Generalizações **27**

Referências **31**

Capítulo 1

Introdução

1.1 Contextualização

CAPTCHAs (do inglês: *Completely Automated Public Turing test to tell Computers and Humans Apart*[1]), são sistemas que se baseiam na apresentação de desafios que devem ser facilmente realizados por humanos e difíceis (de preferência, impossíveis) de serem realizados por máquinas. Seus usos costumam ser associados à restrições de acesso a conteúdo sensível ou ao controle e regulamento de certas atividades em um ambiente computacional; por esse aspecto restritivo, não é difícil a associação de tais tecnologias com a teoria de segurança da informação.

Por outro lado, observa-se que muitos dos *CAPTCHA*s da atualidade aparentam pautar-se em técnicas de segurança por obscuridade, as quais são, historicamente [19], no contexto da segurança da informação, questionadas e criticadas [15] tanto pela ausência de auditabilidade, quanto à vulnerabilidade do cliente para ataques de entidades que possuam acesso à informação segredada.

1.2 Objetivo

Atestar a segurança dos *CAPTCHA*s disponíveis na literatura dada uma avaliação sistemática pautada no princípio da Máxima de Shannon: "o inimigo conhece o sistema"[20], assim como o uso de técnicas recentes de reconhecimento de padrões que possam oferecer risco aos demais *CAPTCHA*s mesmo que partam desse pressuposto.

1.3 Motivação

O trabalho foi motivado pela aparente inconsistência entre o modelo de *CAPTCHA*s praticados e a definição original dos mesmos [1] (que tem como pilar seu caráter *Público*), além de a percepção de que o cenário atual de implementações no mercado apresenta um viés monopolista que não se propõe a solucionar o primeiro ponto, muito pelo contrário, e ainda reforçando um contextos de concentração de *poder de computação humana* (definido pelo uso de pessoas como unidades computacionais) [26, 14].

Capítulo 2

Segurança

2.1 Segurança por obscuridade e o Princípio de Kerckhoffs

Segurança por obscuridade é o nome dado a modelos de sistema de segurança que partem do pressuposto do desconhecimento do atacante sobre parte ou totalidade do funcionamento dos mesmos, ou sobre um conjunto de dados ou regras essenciais para seus funcionamentos. Até o começo do século XX essa era a única forma de segurança conhecida, pois as técnicas rudimentares de criptografia existentes dependiam de tabelas de símbolos, códices ou apenas conhecimento sobre seu funcionamento para serem quebradas; ainda assim, nesse contexto, ao final do séc. XIX, Auguste Kerckhoff definiu seu desiderato [19] que se tornou o bastião da criptografia moderna a qual, por sua vez, seria revolucionada com o advento da computação; nele eram definidos 6 pré-requisitos para o desenvolvimento de cifras seguras, eram eles:

1. O sistema deve ser materialmente, se não matematicamente, indecifrável;
2. É necessário que o sistema em si não requeira sigilo, e que não seja um problema se ele cair nas mãos do inimigo;
3. Deve ser possível comunicar e lembrar da chave sem a necessidade de notas escritas, e os interlocutores devem ser capazes de modificá-la a seu critério;
4. Deve ser aplicável à correspondência telegráfica;
5. O sistema deve ser portátil, e não deve exigir a participação de múltiplas pessoas na sua operação e manuseio;
6. Por fim, o sistema deverá ser simples de usar e não exigir conhecimentos profundos ou concentração dos seus usuários nem um conjunto complexo de regras.

Desses pontos, os itens 2 e 3 resistiram ao efeito do tempo e são padrões de design na maioria dos sistemas de segurança. Particularmente, o item 2 foi reforçado por Claude

Shannon em 1949 [20] (no surgimento dos computadores digitais) em uma definição que passou a ser referida como a *Máxima de Shannon*, traduzido abaixo por mim:

> Para tornar o problema [proteção de sistemas sigilosos] matematicamente tratável, devemos assumir que o inimigo conhece o sistema em uso.

Ou, de forma resumida, diz-se: *O inimigo conhece o sistema*

2.1.1 Modelos de ameaça

Para o design de sistemas de segurança é sempre adequado identificar os atores e os contextos em que a segurança do sistema pode ser comprometida [22], *i.e.* se o atacante possui acesso físico ao equipamento, ou se há compartilhamento da estrutura computacional que possa ser explorada, ou até se o próprio detentor da infraestrutura em que se está hospedado o sistema tem acesso às informações que circulam pelo mesmo. Nesse último caso, com o advento das técnicas de criptografia assimétrica, tornou-se possível serviços que ofertariam criptografia ponta-a-ponta em que apenas os usuários, e àqueles com quem desejam compartilhar, têm acesso a dados ali gerenciados.

Uma analogia para *CAPTCHA*s com esse grau de segurança, seria justamente o empenho de técnicas não-dependentes de obscuridade, que garantiriam o pé de igualdade entre os usuários, potenciais atacantes, pela perspectiva do sistema, e as plataformas que também são vistas como suspeitas de ataque em um sistema robusto[1].

Cabe uma última ressalva: não há nada inerentemente errado em *sistemas baseador em segredo*, desde que haja confiança de seus usuários de que eles são adequadamente geridos e auditados[22].

2.2 Reflexão

*CAPTCHA*s são intrinsecamente sistemas de segurança, tanto pelo caráter de restrição de acesso, proteção de sistema computacional e autenticidade de requisição, quanto pelo modelo atacante-defensor observado em inúmeras técnicas de ataque apresentadas na literatura que acarretam em mudanças no panorama de implementações de defesa no mercado (e.g. [6]), tal qual sistemas de segurança clássicos, como é o caso da criptografia.

Ainda assim, na literatura, diferentemente da criptografia, sempre se pressupõe o desconhecimento do atacante sobre o funcionamento do sistema e sobre os dados utilizados no mesmo (em técnicas de ataque *caixa-preta*). Nesse trabalho, portanto, definiu-se o mesmo critério, respeitando a *Máxima de Shannon*, partindo-se do pressuposto de máximo conhecimento e acesso do atacante, tal qual definido por Von Ahn [1]:

> Our definitions imply that an adversary attempting to write a program that has high success over a CAPTCHA knows exactly how the CAPTCHA works. The only piece of information that is hidden from the adversary is a small amount of randomness that the verifier uses in each interaction.

Ou seja: a única informação que se assume o atacante não possuir é o dado aleatório usado na geração do desafio (observa-se que, ainda assim, o sistema poderia estar vulnerável

à ataques físicos, mesmo que sem acesso ao sistema, um agente malicioso poderia realizar um ataque de canal lateral).

Capítulo 3

*CAPTCHA*s

3.1 O que são *CAPTCHA*s?

*CAPTCHA*s são programas feitos para avaliar e gerar desafios que a maioria dos humanos são capazes de realizar e que sistemas computacionais da atualidade não são capazes de fazê-lo; dessa forma, sendo os pontos anteriores satisfeitos, esses programas são capazes de diferenciar humanos de computadores, permitindo:

- limitar o acesso de recursos computacionais apenas à pessoas naturais;
- servir de apoio a plataformas de autenticação, prevenindo ataques de força bruta ou ataques de negação de serviços;
- restringir o escopo de plataformas públicas online para o acesso automatizado;
- dar ao gestor de páginas web mais segurança na veracidade dos dados de análise de fluxo em suas páginas.

Observa-se que, diferentemente de protocolos criptográficos, não há como quantificar objetivamente a dificuldade desses desafios, pois são dependentes da dificuldade em se replicar e automatizar programaticamente o comportamento humano, assim, supõe-se que o conjunto dos desafios adequados pertence à área de Inteligência Artificial, e, portanto, deve ser, analogamente ao contexto criptográfico, de difícil solução e, ainda assim, simples o suficiente para a maioria dos humanos solucioná-lo facilmente.

3.1.1 Histórico e Exemplos

A primeira proposta de sistema que utiliza um teste de Turing reverso para identificar acesso humano na internet é de Moni Naor em 1996 [18], para combater os cada vez mais frequentes ataques de *spam* em plataformas gratuitas na internet; ele sugere o uso de:

1. desafios de reconhecimento de caracteres;
2. classificação e rotulação semântica de imagens;

3. reconhecimento de fala;

4. avaliação lógica e semântica em texto escrito.

Desafios estes que ramos da Inteligência Artificial, como visão computacional e processamento de linguagem natural se propõem solucionar.

No ano seguinte, o site de busca AltaVista, para evitar uma inundação de submissões em sua plataforma, implementou o sistema, que hoje reconheceríamos como um *CAPTCHA* de texto, com uma sequência de caracteres distorcidos a ser reconhecido pelo usuário; a segurança desse modelo advinha da incapacidade dos sistemas de reconhecimento de caracteres (*OCR*) da época não serem capazes de interpretar letras distorcidas satisfatoriamente. 3 anos depois, nos anos 2000, o site Yahoo, acometido dos mesmos males em suas salas de bate-papo, utiliza a fórmula conhecida, dessa vez com outras deformações e pedindo ao usuário a identificação de várias palavras.

Em 2001 os pesquisadores da AltaVista patenteiam [13] a sua versão do *CAPTCHA* textual e, por fim, em 2003, Luis Von Ahn cunha o termo *CAPTCHA* (do inglês: *Completely Automated Public Turing test to tell Computers and Humans Apart*, "teste de Turing público completamente automatizado para diferenciação entre computadores e humanos") em um artigo [1] reforçando todos os pontos ditos antes por Naor: como a necessidade de publicização do algoritmo, assim como da base de dados usados para sua construção, e a definição do *CAPTCHA* como um protocolo criptográfico cuja segurança advém da suposição de se basear em um problema "difícil"da inteligência artificial.

Abaixo seguem exemplos dos sistemas da AltaVista e Yahoo:

(a) *AltaVista (1996)* **(b)** *Yahho - EZ-Gimpy (2000)*

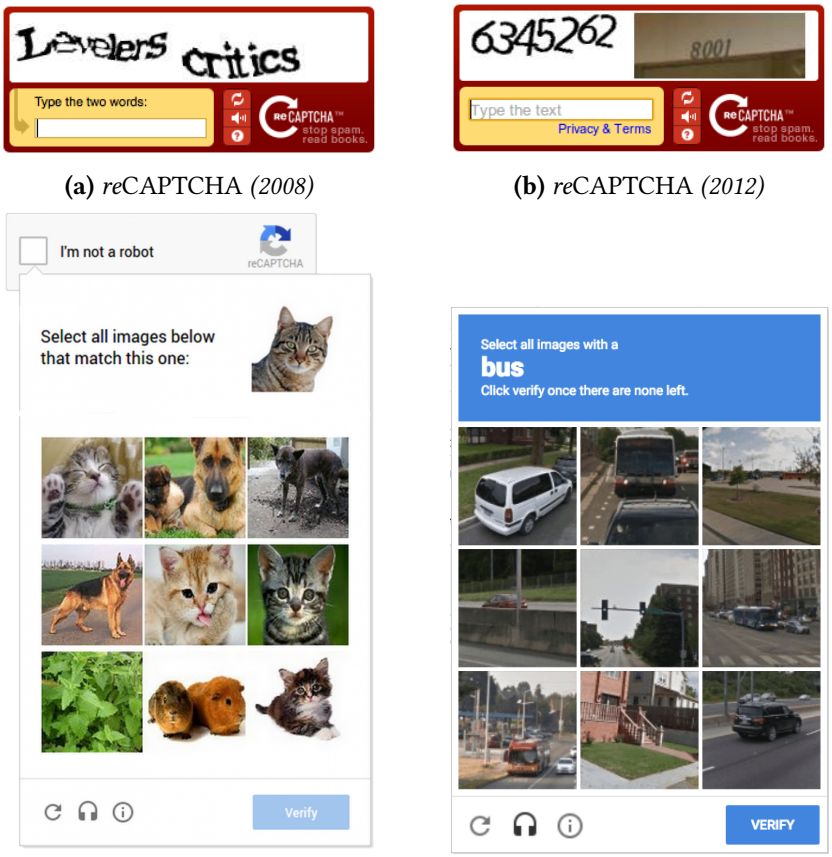
Figura 3.1: *1ª Geração de Captchas textuais*

Em 2008 foi proposta uma nova versão para o *CAPTCHA*, o re*CAPTCHA*[27], idealizado por Von Ahn motivada pela ideia de computação humana [26] – usar a capacidade humana de reconhecimento e rotulação de imagens – para auxiliar projetos de digitalização de livros acumulando essa função ao sistema anterior.

No ano seguinte, a Google compra o sistema e em 2012, substitui a digitalização de livros pela identificação de números de endereços fotografados pela sua ferramenta de mapa e navegagação, o Google StreetView.

Em 2013 treinam uma Rede Neural Convolucional[6] com acuidade de 99.8%, maior do que qualquer humano e no ano seguinte a Google substitui o *CAPTCHA* textual pelo re*CAPTCHA*V2, um sistema baseado no comportamento que, a depender dos dados coletados do usuário, pode apresentar um *CAPTCHA* de classificação de imagem, ou apenas um botão com os dizeres "*Não sou um robô*".

Com o tempo, a classificação de imagens foi se especializando para a área de transportes autônomos e em 2018 foi lançada a versão re*CAPTCHA*V3 em que os dados telemétricos coletados são classificados e atribuídos pontos e, em função disso, são ou não apresentados os desafios para os usuários.



(a) *re*CAPTCHA *(2008)* **(b)** *re*CAPTCHA *(2012)*

(c) *re*CAPTCHA*v2 (2014)* **(d)** *re*CAPTCHA*v2 (2017)*

Figura 3.2: *Evolução do re*CAPTCHA

3.2 Taxonomia dos *CAPTCHASs*

3.2.1 Metodologia

Avaliou-se o impacto da publicização dos dados utilizados na geração de *CAPTCHA*s sobre a vulnerabilidade dos mesmos às técnicas de busca reversa e reconhecimento de padrões, assim como técnicas de ataques específicos a determinados sistemas utilizando-se apenas processamento de imagem e de sinais.

3.2.2 Desenvolvimento

Nesse trabalho, foi criada uma taxonomia (do apêndice, Tabela A.1) pelos tipos de desafio [2, 8], e foram incluídas categorizações pelo funcionamento, entre modelos:

- "baseado em dados"(coluna **Dados**), se são modelos que dependem de (e, por consequência, dependem da inacessibilidade dos atacantes a) bases de dados, pesos de modelos treinados, conjunto específico de regras, algoritmos, tabelas, etc, qualquer dado que se assume apenas o modelo de *CAPTCHA* ter acesso;
- "generativos"(coluna de mesmo nome na tabela de taxonomia), se são modelos que sintetizam completamente os dados usados no desafio, ou se aplicam um conjunto de transformações nos dados (observa-se que a prática de adicionar ruídos aos dados pode ser feita em quase todos os modelos, porém a taxonomia parte do pressuposto a configuração dos modelos sugeridos na literatura ou presentes no mercado.

Além disso, para garantir uma melhor representatividade dos desafios de cada categoria, foram definidas categorias próprias, ou seja, ao invés de utilizar as categorias *(Texto, Imagem-{Click, Seleção, Slide}, Áudio, Vídeo, Interativo, Desenho, Matemático, ...)* comuns na literatura [32, 2, 8] – observa-se que, pela taxonomia proposta, muitos são de categoria *Imagem + Comportamental*, ou seja, para uma avaliação de segurança do modelo proposto, a mesma será limitada por aquele de maior vulnerabilidade.

Assim, os 4 tipos base, classificados em função da coleta e apresentação dos dados do desafio, foram:

- **Visual** e **Sonoro**: Compreendendo os desafios baseados nas faculdades visuais e auditivas humanas;
- **Telemétrico**: Referente aos desafios baseados na coleta de dados dos dispositivos que a pessoa usar;
- **Token**: Técnicas novas em que o próprio dispositivo intermediário realiza a certificação do acesso.

As 12 categorias taxonômicas, associadas aos 4 tipos de coleta/apresentação de dados, são:

- Textual;
- Lógico;
- Imagético de Seleção;
- Imagético de Transformação;
- Vídeo;
- Reconhecimento de fala;
- Reprodução de fala;
- Auditivo;
- Sensorial;
- Comportamental;
- Token Criptográfico;
- Prova de Trabalho (baseado em algoritmos de *Proof-of-Work*).

Cada categoria, mais facilmente visualizada no diagrama da Figura 3.3, define as demais características do desafio, como acessibilidade, vulnerabilidade conhecida, etc. Na Tabela 3.1 foram organizadas as descrições de cada categoria de desafio, assim como exemplos de suas interfaces ou de seus funcionamentos.

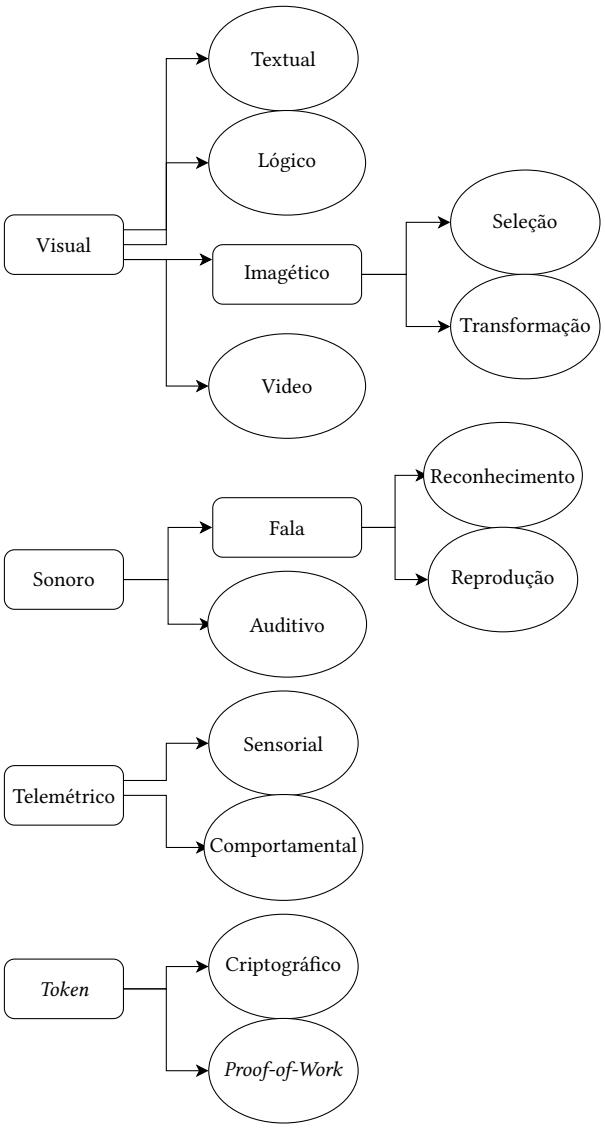


Figura 3.3: *Diagrama de Taxonomias por categorias de Desafio*

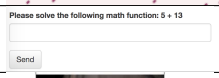
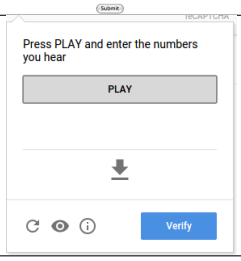
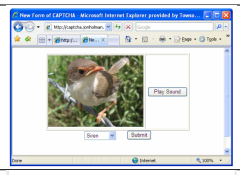
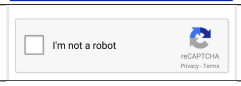
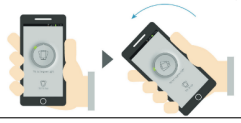
Categoria	**Desafio**	**Interface**
Textual	Identificação de caracteres	
Lógico	Resolução de problema lógico	
Imagético de Seleção	Identificação de objetos em imagens	
Imagético de Transformação [7]	Reconhecimento de transformações em imagens	
Vídeo [12]	Interpretação de conteúdo em vídeo	
Reconhecimento de Fala	Reconhecimento de caracteres ditados	
Reprodução de Fala [21]	Reprodução verbal de palavras	
Auditivo[10]	Reconhecimento de sons	
Comportamental	Análise de dados de uso	
Sensorial [9]	Análise de dados de uso	
Proof-of-work [11]	Comprovação de computação de carga	
Criptográfica [16]	Autenticação de chave eletrônica por estrutura certificadora	

Tabela 3.1: *Categorias com respectivos exemplos*

3.3 *CAPTCHA*s como sistemas de segurança

Feita a devida taxonomia (Tabela A.1), nota-se que, para a maioria dos modelos, não há como se avaliar adequadamente sua segurança se a mesma for "baseada em dados"(lê-se: baseada em obscurantismo) e, nesse caso, foram rotuladas com grau de segurança indefinido, principalmente se pretendermos ter um modelo de ameaça que restrinja o acesso do detentor do sistemas aos dados de seus usuários.

Os modelos que sobram, e que pode-se fazer alguma avaliação concreta sobre a segurança são:

- Considerados Inseguros:
  - Textual: devido aos inúmeros ataque já conhecidos;
  - Lógico: a depender de seu funcionamento, pois se for de lógica matemática, basta um sistema simples de Reconhecimento óptico de caracteres (*OCR*) para atacá-lo, por outro lado, se forem desafios de lógica semântica, como não parece haver formas de gerá-los "generativamente", então seria dependente de uma base de dados e, por sua vez, do segredo sobre os mesmos;
  - Reconhecimento de Fala: se baseado em base de dados, tal qual nos itens passados, é de segurança indeterminada e, se for estritamente generativo/sintetizado, o espaço de busca será limitado, potencialmente expondo o sistema a maior vulnerabilidade.
  - Reprodução de Fala: inconclusivo pela insipiência de modelos assim na literatura.
- Considerados Seguros:
  - *Proof-of-Work* e o modelo de *Token Criptográfico* podem ser considerados seguros, uma vez que ambos possuem bases sólidas e exemplos de modelos semelhantes no contexto da criptografia.

Observa-se que as demais categorias, por não poderem ser avaliadas em modelo de caixa-branca (quando o atacante detém o conhecimento completo do funcionamento do mesmo), possuem avaliação de segurança inconclusiva e que, na definição dada de CAPTCHAs, talvez sequer possam ser classificadas como tal, afinal não são modelos *Públicos*.

3.3.1 Considerações econômicas

A análise de segurança dos modelos de CAPTCHA não estaria completa se não fosse feita a análise do contexto econômico no qual estão inseridos, afinal, um dos tipos de ataque mais comuns, e que exige o mínimo conhecimento do atacante sobre o contextos, é chamado, em inglês, de *Human Relay Solver*[8] e consiste em enviar o desafio para ser resolvido por outras pessoas, mediante uma baixa remuneração, e receber a solução pronta, que possa ser autenticada com um serviço, por exemplo.

Esse tipo de ataque pode então ser automatizado com *scripts* e rodar paralelamente sendo apenas limitado pelo preço da contratação do serviço [17].

A maior dificuldade de se combater esse tipo de ataque é que o CAPTCHA cumpre adequadamente sua função, na outra extremidade há, de fato, um ser humano solucionando o desafio, mas como a obtenção dessa solução foi automatizada, voltam a ser possíveis todos os mesmos problemas que se propunham resolver. A solução nesse caso, costuma ser o uso de um protocolo de segurança adequado que limite o número de requisições de clientes, ou que avalie o conjunto de informações dos acessos e auxilie na classificação de acessos automatizados ou não, tal qual nos desafios de CAPTCHAs Comportamentais.

3.4 Resultados

Observou-se que a maioria dos sistemas da literatura partem de um pressuposto de inacessibilidade de um atacante à relação de etiquetas e desafios, pois, mesmo se aplicadas transformações ao desafio proposto, o mesmo poderá manter-se vulnerável a técnicas de detecção de características invariantes às mesmas.[8]

A partir da taxonomia sugerida, propõe-se reavaliar a forma que se avalia e desenha CAPTCHAs, de preferência levando-se em consideração o desenho original [1, 18], respeitando-se os mesmos princípios de segurança que protocolos de criptografia procuram fazer.

Capítulo 4

Prova de Conceito

Como visto no capítulo 3, poucos são os modelos de desafio que poderiam ser considerados seguros, apenas os baseados em *Tokens*, e a avaliação desses é incipiente, pois, para o caso do modelo *Proof-of-Work*[11], poder-se-ia argumentar sequer se tratar de um CAPTCHA, afinal, não se propõe a diferenciar o acesso humano do automatizado, e sim a limitar os acessos indevidos; já o modelo de *Token Criptográfico* [25], é bem promissor, mas exige uma configuração de hardware específica e ainda não está adequadamente estabelecido para poder se avaliar em todo seu escopo.

Assim, optou-se por desenvolver uma prova de conceito de um ataque genérico de reconhecimento de padrões para ataque aos sistemas generativos conhecidamente inseguros (vide a seção 3.3, mais especificamente, do sistema de CAPTCHA textual, por ser o mais amplamente implementado, com inúmeras variações e uma literatura extensa de implementações [24, 29, 5, 4, 28, 31] e técnicas de ataque [30].

Em um primeiro momento, pretendeu-se utilizar Redes Neurais Generativas Adversárias (GANs), pois, exigiriam uma base menor de imagens rotuladas [31], mas, como iria se explorar o caso de CAPTCHAs textuais disponíveis publicamente, optou-se por uma abordagem mais ingênua: de se gerar todo o *dataset* (arbitrariamente selecionado em 1 milhão de elementos, por ser metade da quantidade de elementos dita ser necessária [31]) e aplicar uma Rede Neural Convolucional adequada para solucioná-los.

4.1 Metodologia

Usando modelos de geração de *CAPTCHA* textual, gerou-se conjuntos de dados com 1 milhão de elementos para o treinamento de redes neurais convolucionais genéricas realizando variações no modelo geracional: tamanho de alfabeto e dimensões das palavras, avaliando-se a generalidade dos modelos treinados às variações.

Foram escolhidos 2 modelos de CAPTCHA: a biblioteca captcha.py[1] e o modelo usado na extensão *ConfirmEdit* do projeto MediaWiki (base para a Wikipédia), sendo que, no primeiro foram exploradas as variações apelidadas de **Default** (configuração padrão da

[1] https://pypi.org/project/captcha/

biblioteca), **Wheezy** (modelo depreciado com cores e taxas de deformação fixas), **Fonts** (que utiliza o mesmo conjunto de transformações que o **Default**, mas, para cada caractere sorteia a fonte a ser utilizada na renderização do mesmo); e no segundo foram exploradas as variações **Wiki** (configuração padrão da extensão) e **Old** (configuração antiga, considerada menos segura).

4.1.1 Características de Segurança

A Tabela 4.1 exibe as variações exploradas de ambos os modelos estudados, descrevendo o conjunto de características de segurança [3, 31] usado em cada um, essas são determinantes para a definição de segurança e acessibilidade de um CAPTCHA textual.

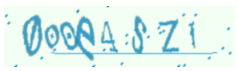
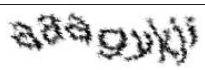
Modelo	**Exemplos**	**Características de segurança referentes a**	
		Segmentação	Reconhecimento
Default		Sobreposição de caracteres, linha de oclusão, ruído, fundo sólido	Sorteio de cor, variação de tamanho de fonte, rotação, distorção, ondulação
Wheezy		Linha de oclusão, ruído, fundo sólido	Variação de tamanho de fonte, rotação, ondulação
Fonts		Sobreposição de caracteres, linha de oclusão, ruído, fundo sólido	Sorteio de cor, variação de tamanho e tipo de fonte, rotação, distorção, ondulação
Wiki		Ruído, fundo ruidoso	Variação de cor, distorção, ondulação
Old		Ruído, fundo sólido	Distorção, ondulação

Tabela 4.1: *Características de Segurança*

4.1.2 Outras variações exploradas

Além nas variações de características de segurança, foi explorada a variação de acuidade das redes treinadas para atacar CAPTCHAs textuais de comprimento **2**, **4**, **8** e **16** caracteres, assim como modelos com conjuntos de caracteres (estritamente pertencente ao alfabeto latino sem diacríticos) referenciados por: **a** (letras minúsculas), **A** (letras maiúsculas) e **n** (números), e todas as possíveis combinações (e.g. **Ana** = maiúsculas, minúsculas e números, somando 62 elementos). Devido a incompatibilidades com a estrutura de geração utilizado pela extensão *ConfirmEdit*, não foram feitas variações nos modelos **Wiki** e **Old**.

Com esses 2 modelos e todas suas variações foram exploradas:

- A capacidade de se treinar uma Rede Neural para solucionar cada variação;
- A diferença que cada mudança gera na acuidade da Rede Neural;

- A possibilidade de generalização dos modelos treinados para serem aplicados em outras variações e a taxa de acerto dessa generalização .

4.2 Desenvolvimento

Explorou-se arquiteturas de Redes Convolucionais genéricas para reconhecimento de imagem e foi escolhido a VGG16 [23], pela capacidade e eficiência em lidar com larga escala de dados.

De forma empírica (realizando alguns testes e selecionando aqueles que treinaram redes mais acuradas) foram escolhidos como parâmetros de treinamento:

- Fatia de **20%** do *dataset* destinada à validação do treinamento;
- Fatia de **10%** do *dataset* destinada à avaliação da rede treinada;
- **20** *épocas* (número de vezes que a rede passa pela fatia de treinamento do *dataset*) (normalmente convergindo para uma alta taxa de acerto antes de serem completadas todas as épocas);
- **50** amostras por lote (número de imagens treinadas por vez);

Buscou-se códigos livres de implementações já feitas[2] de solução de CAPTCHAs textuais e o mesmo foi adequado para, dados parâmetros configurados, treinar uma rede para solucionar cada variação dos modelos de CAPTCHA textual escolhidos.

Também buscou-se códigos livres de implementações de geradores de CAPTCHAs textuais, mas, como quase todos estavam depreciados, limitou-se a utilizar, além da biblioteca *captcha.py*, a extensão de *ConfirmEdit*[3] da *MediaWiki*, modificando o código da mesma para gerar o *dataset* de acordo com as variações desejadas. Foi criado também um *script* para gerar todas as variações desejadas do modelo da biblioteca *captcha.py*.

Assim, a fim de otimizar o espaço e tempo de computação da geração, treinamento e avaliação desses códigos, foram feitos *scripts* para paralelizar e automatizar cada uma dessas etapas.

4.2.1 Recursos computacionais em números

Ambiente de computação

Os códigos foram todos rodados em um servidor disponível pela Rede IME (apelidado de *BrucutuVI*) que possui 80 núcleos, 520 GB de memória RAM, 20 TB de armazenamento e uma GPU Tesla V100 com 16 GB de memória.

Espaço e tempo

Como cada variação gera uma imagem com, em média 10KB, o *dataset* referente a cada variação tem, aproximadamente, 9.5GB, e como há 5 variações de modelo, 7 de conjunto

[2] https://github-com.translate.goog/ypwhs/captcha_break

[3] https://github.com/wikimedia/mediawiki-extensions-ConfirmEdit

de caracteres e 4 de extensão de CAPTCHA textual, e considerando que a variação de conjunto de caracteres só foi aplicada a 3 dos modelos, o conjunto total de dados gerado foi, aproximadamente, de 900 GB.

Para cada variação, levou-se, em média, 300 minutos (5 horas) para se fazer o treinamento da rede correspondente, ou seja, com as 92 variações ($3 * 7 * 4 + 2 * 4$), o treinamento levaria, aproximadamente, 19 dias de tempo de computação, por isso, rodando em paralelo, foi possível diminuir para pouco mais de 1 semana;

Para a análise de generalizações (avaliação de modelos de solução (Rede Neural) para diferentes contextos de modelos de geração (CAPTCHA)), foram necessárias 1764 ($(7 * 3)^2 * 4$) avaliações (pois só foram feitas para as variações dos modelos **Default**, **Wheezy** e **Fonts** por serem baseados na biblioteca *captcha.py*), permitindo avaliar melhor a diferença que cada variação tem sobre o resultado final. A avaliação leva muito menos tempo que o treinamento, em média 30 minutos, o que levaria até 36 dias, porém, excluindo incompatibilidades de modelos (*e.g.* a rede treinada para reconhecer números nunca acerta um modelo generativo de apenas letras) e paralelizando as avaliações, pôde ser finalizado em 1 semana.

4.2.2 Tecnologias

O código[4] de treinamento e avaliação da Rede Neural foi completamente desenvolvido em *Python3* com o auxílio das bibliotecas *TensorFlow* e *Keras*.

Para a geração das imagens, também se utilizou *Python3* (necessitando atualizar o código depreciado da extensão *ConfirmEdit*), com o auxílio da biblioteca *Pillow* para síntese e manipulação de imagens.

Os demais *scripts* para gestão do fluxo do dados e escalonamento de execução foram feitos em *BASH*.

4.2.3 Desafios

O principal desafio foi a gestão dos dados, em retrospecto, teria sido melhor utilizar o formato hdf5 para encapsular os *datasets* (pois sistemas de arquivo têm dificuldade em gerenciar muitos arquivos pequenos), por outro lado, o fato de estar organizado em arquivos legíveis e individuais, facilitou bastante a depuração e compreensão do funcionamento da rede, além de permitir treinar uma rede solucionadora de CAPTCHAs textuais com um conjunto qualquer de imagens de treinamento (desde que seguindo a mesma organização).

Outra dificuldade foi que, por motivos ainda desconhecidos, o treinamento da rede rodava indefinidamente e, independente de seu critério de parada (atingir uma taxa de erro mínima), mas sem relatar nenhuma anormalidade (ou registrar algo no registro de épocas), como se estivesse parado, mas consumindo processamento, logo como se estivesse em um *deadlock*.

[4] Disponível em https://gitlab.com/carybe/hip_tcc

4.3 Resultados

Para os modelos baseado na extensão *ConfirmEdit*, foi gerada a Tabela 4.2 com as taxas de acerto dos modelos de rede neural treinados quando aplicados aos respectivos conjuntos de teste.

Extensão do texto (n)	**Wiki**	*Old*
2	97.602%	98.844%
4	96.522%	98.610%
8	94.687%	98.611%
16	54.082%	50.858%

Tabela 4.2: *Taxa de acerto para o ConfirmEdit*

Já as taxas de acerto dos modelos baseados na biblioteca *captcha.py*, podem ser vistos nas tabela de generalizações ou na Tabela 4.3 (também, quando avaliado o conjunto de testes).

Charset	**Default**	**Wheezy**	**Fonts**
A	99.784%	99.884%	99.002%
n	99.954%	99.992%	99.864%
a	99.896%	99.936%	98.938%
An	96.898%	99.522%	96.570%
na	99.552%	99.872%	96.324%
Aa	96.452%	99.272%	90.526%
Ana	95.536%	99.050%	89.892%

(a) *Taxa de acerto para o captcha.py com n=2*

Charset	**Default**	**Wheezy**	**Fonts**
A	99.182%	99.314%	98.000%
n	99.858%	99.942%	99.756%
a	99.570%	99.796%	96.922%
An	95.214%	98.944%	93.196%
na	99.074%	99.732%	92.832%
Aa	93.448%	98.614%	83.154%
Ana	91.286%	98.464%	80.922%

(b) *Taxa de acerto para o captcha.py com n=4*

Charset	**Default**	**Wheezy**	**Fonts**
A	97.603%	96.009%	95.198%
n	99.651%	99.761%	99.593%
a	97.978%	99.244%	94.522%
An	88.751%	96.204%	86.927%
na	96.401%	98.890%	86.938%
Aa	83.894%	96.318%	70.533%
Ana	80.784%	96.045%	65.326%

(c) *Taxa de acerto para o captcha.py com n=8*

Charset	**Default**	**Wheezy**	**Fonts**
A	22.813%	50.059%	21.618%
n	38.186%	19.308%	38.658%
a	23.834%	85.220%	12.412%
An	17.308%	66.616%	15.345%
na	18.318%	82.232%	8.358%
Aa	10.920%	69.191%	4.851%
Ana	1.187%	67.180%	4.339%

(d) *Taxa de acerto para o captcha.py com n=16*

Tabela 4.3: *Taxas de acerto para diferentes extensões de texto (n)*

Assim, observa-se que:

- A variação de maior impacto na acurácia da rede foi a extensão do texto do CAPTCHA gerado, isso pois, de forma análoga ao contexto de chaves em segurança da informação, aumenta-se o espaço de busca do atacante em proporção ao número de classes (*charset*).

- Mudanças no *charset* também afetam substancialmente a capacidade da rede atacante em solucionar o CAPTCHA, na mesma analogia do ponto anterior, assemelha-se ao uso de caracteres especiais ou números em uma senha para aumentar a entropia da mesma e diminuir a chance de um atacante adivinhá-la;
- O tamanho do charset tem mais impacto sobre taxa de acerto do modelo neural do que as características individuais de seus elementos; por exemplo: observa-se que as taxas de acerto para CAPTCHAS gerados com o conjunto das letras maiúsculas (A) são semelhante aos gerados com o conjunto das letras minúsculas (a);
- Como esperado, quanto maior o número de características de segurança do modelo, maior a dificuldade apresentada ao atacante e, assim, as evoluções dos modelos **Old** para **Wiki** e **Wheezy** para **Default** nas suas respectivas bases de código aparentam ser justificáveis, ao mesmo tempo percebe-se que conferem muito menos segurança que variações de aumento de extensão do texto e do *charset*;
- Fica também evidente que as variações que conferem maior segurança, prejudicam também a acessibilidade do desafio, indo de encontro ao propósito original, por prejudicar mais a capacidade de pessoas passarem no desafio do que máquinas;
- Para o caso das generalizações, observou-se na aplicação dos modelos treinados para conjuntos de dados distintos uma maior capacidade de generalização dos modelos com maior variabilidade e uma razoável incompatibilidade entre o modelo depreciado com o padrão, como era esperado.

Capítulo 5

Conclusão

Concluiu-se que, apesar de *CAPTCHA*s, terem sido concebidos como um teste público, as alternativas propostas na literatura, assim como disponíveis no mercado, dificilmente partem desse pressuposto, falhando em apresentar sistemas comprovadamente seguros.

Os sistemas generativos textuais, como o explorado no estudo de caso, são vulneráveis a ataques de sistemas de reconhecimento de padrões e, apesar de estratégias comumente usadas para aumentar a segurança de senhas em sistemas criptográficos (aumento de alfabeto e de extensão) contribuírem significativamente para a segurança do sistema, as mesmas diminuem a acessibilidade do mesmo.

Alternativas recentes baseadas em *token*s criptográficos apresentam o melhor equilíbrio entre segurança e acessibilidade impondo, porém, barreiras econômicas ao seu uso.

5.1 Trabalhos Futuros

Os seguintes pontos surgiram durante o desenvolvimento do trabalho, mas não puderam ser adequadamente averiguados:

- Observou-se, durante a pesquisa, alguns modelos que não seriam possíveis de ser encaixados em algumas categorias da taxonomia: sistemas baseados em avaliação semântica de imagens, de interfaces físicas variadas, com regras transitórias; então a taxonomia deverá ser revista para buscar acomodar estratégias menos comuns.
- Acredita-se ser possível, através de estratégias de busca reversa, utilizando descritores invariantes a transformações (como a família do SIFT, ORB, SURF, KAZE, ...) atacar de forma eficiente um CAPTCHA Imagético de Seleção, mesmo com adição de ruído e transformações, pois haveria uma correspondência 1-para-1 entre a imagem do desafio e a presente na base de dados utilizada; porém, não foi encontrada pesquisa que se propusesse a fazê-lo, assim essa é uma tarefa perfeita a ser realizada futuramente.

- Para tornar o código utilizado na Prova de Conceito mais genérico, é necessário que fossem feitas alterações na rede que permitissem a identificação de CAPTCHAs textuais de extensão arbitrária (ou limitada a um valor definido); o uso de CTC (*Connectionist temporal classification*, uma técnica de pontuação para classes em saídas de Redes Neurais que permite o treinamento de redes com *datasets* com elementos de dimensões variadas) possibilitaria isso, mas ao custo de prejudicar a taxa de acerto da rede. Seria adequado avaliar esses pontos para encontrar uma melhor estratégia.
- Seria interessante testar a rede proposta na Prova de Conceito com outros modelos generativos e também ver os resultados do treinamento da mesma com conjuntos de regras variadas para se verificar se há melhores resultados na avaliação de generalização.
- No atual contexto de modelos generativos de difusão, seria interessante o desenvolvimento de um sistema CAPTCHA que auxiliasse o treinamento do modelo através de reconhecimento semântico de mídias ou textos gerados, comparando-os com a semântica utilizada de entrada para a geração.

Apêndice A

Taxonomia

Tabela A.1: *Taxonomia pela perspectiva de segurança do desafio proposto*

Tipo	Categoria	Desafio	Generativo	Dados	Vulnerabilidade	Acessibilidade	Obs	Seguro
Visual	Textual	Identificação de caracteres	Sim	Não[1]	OCR e CNNs	Média	Mais difundido e com mais variações	Não
Visual	Lógico	Resolução de Problema Lógico	Sim	Não	OCR	Média	Se usado sozinho, oferece pouca proteção	Não
Visual	Imagético de seleção	Reconhecimento de objetos em imagens	Não	Sim	CNNs e SIFT	Média	Suscetível a ataques se o atacante tiver acesso à base de dados	Não ou Indefinido

[1] Pode ser que use um dicionário, facilitando bastante o ataque [24]

Tipo	Categoria	Desafio	Generativo	Dados	Vulnerabilidade	Acessibilidade	Obs	Seguro
Visual	Imagético de transformação	Reconhecimento de transformações em imagens	Sim	Sim	CNNs e SIFT	Média	Suscetível a ataques se o atacante tiver acesso à base de dados	Não ou Indefinido
Visual	Vídeo	Interpretação de conteúdo em vídeo	Não	Sim	CNN + LSTM	Baixa	As diferenças linguísticas entre países dificultam o uso de modelos genéricos, necessitando de lterações regionais	Não ou Indefinido
Sonoro	Reconhecimento de fala	Reconhecimento de textos ditados ou sonoramente sintetizados	Sim	Sim[2]	TTS (CTC + LSTM)	Média	As diferenças linguísticas entre países dificultam o uso de modelos genéricos, necessitando de lterações regionais	Não ou Indefinido
Sonoro	Reprodução de fala	Reprodução verbal de palavras ou letras	Sim	Não[3]	CTC + LSTM	Média	As diferenças linguísticas entre países dificultam o uso de modelos genéricos, necessitando de iterações regionais	Não ou Indefinido
Sonoro	Auditivo	Reconhecimento de sons	Não	Sim	CNN	Média	Diferenças culturais influenciam no arcabouço de sons que as pessoas têm familiaridade com	Não ou Indefinido
Telemétrico	Comportamental	Avaliação de dados de telemetria	Não	Sim	RNN	Alta	Suscetível a ataques se o atacante tiver acesso aos critérios utilizados	Não ou Indefinido

[2] Se for completamente sintetizado, então deve ser "Não-dependente de dados"

[3] Se o modelo de reconhecimento de fala for limitado, pode ser que haja um conjunto reduzido de fonemas capaz de interpretar

Tipo	Categoria	Desafio	Generativo	Dados	Vulnerabilidade	Acessibilidade	Obs	Seguro
Telemétrico	Sensorial	Análise de dados dos sensores referentes a realização de movimentos	Não[4]	Sim	RNN	Média	Suscetível a ataques se o atacante tiver acesso aos critérios utilizados	Não ou Indefinido
Token	*Proof-of-Work*	Comprovação da máquina cliente da resolução de problema computacional de dificuldade proporcional à carga do servidor	Sim	Não	Botnet	Alta	Pode ocasionar desigualdade na distribuição de dados dadas as diferenças computacionais dos usuários	Sim
Token	Criptográfico	Autenticação de chave eletrônica por estrutura certificadora	Sim	Não	Uso automatizado de múltiplos tokens	Alta	Depende de uma estrutura certificadora	Sim

[4] Dada a baixa acessibilidade de muitos movimentos possíveis, o conjunto de possibilidades é, provavelmente, reduzido

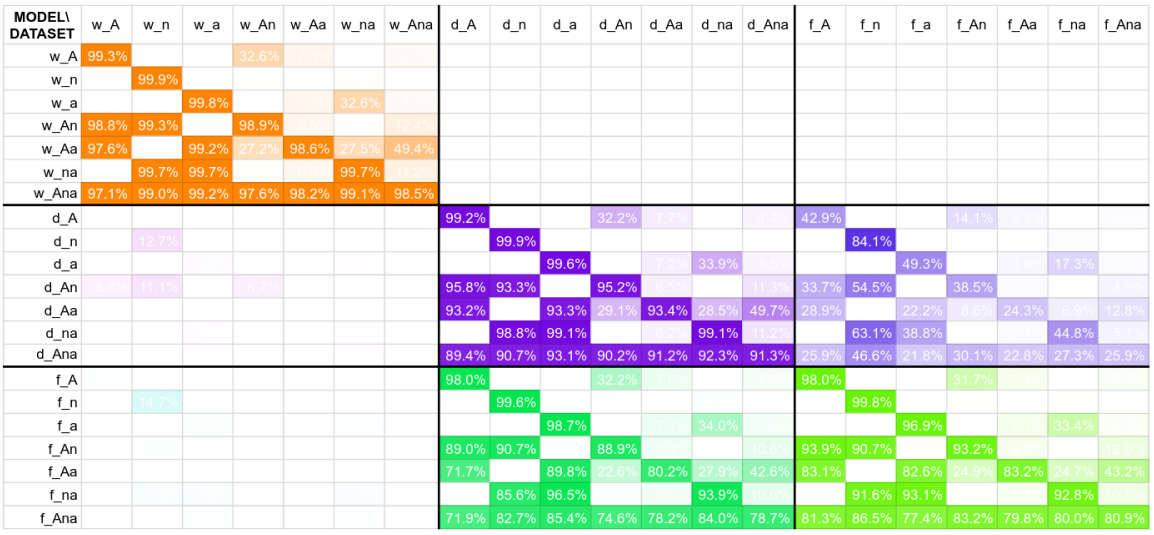
Apêndice B

Generalizações

MODEL\ DATASET	w_A	w_n	w_a	w_An	w_Aa	w_na	w_Ana	d_A	d_n	d_a	d_An	d_Aa	d_na	d_Ana	f_A	f_n	f_a	f_An	f_Aa	f_na	f_Ana
w_A	99.9%			57.3%	27.1%		20.6%														
w_n		100.0%																			
w_a			99.9%		26.9%	56.9%	20.9%														
w_An	99.6%	99.3%		99.5%	26.3%		34.7%														
w_Aa	99.1%		99.5%	52.4%	99.3%	52.3%	70.2%														
w_na		99.9%	99.9%		25.2%	99.9%	33.9%														
w_Ana	98.7%	99.1%	99.5%	98.8%	99.1%	99.4%	99.1%														
d_A	34.1%			20.7%				99.8%			56.9%	27.5%		21.2%	69.9%			40.4%	19.5%		
d_n		28.8%							100.0%							91.1%					
d_a										99.9%		26.8%	58.8%	21.1%			73.9%		20.2%	43.4%	
d_An								98.3%	93.7%		96.9%	24.9%		33.2%	62.1%	67.0%		62.9%	16.7%		
d_Aa								96.3%		96.7%	54.9%	96.5%	55.0%	70.9%	56.6%		51.1%	31.8%	52.8%	28.5%	38.8%
d_na									99.2%	99.7%		24.7%	99.6%	33.3%		78.9%	66.4%		16.8%	70.0%	
d_Ana								94.7%	95.2%	96.4%	94.8%	95.6%	95.9%	95.5%	52.4%	65.0%	49.4%	55.7%	50.3%	53.0%	52.5%
f_A	43.5%							99.2%			56.9%	27.4%			99.0%			56.7%			
f_n									99.8%							99.9%					
f_a										99.7%		26.8%	59.8%	21.4%			98.9%		26.8%	57.9%	
f_An		33.0%						94.3%	95.6%		94.4%				96.8%	95.4%		96.6%			
f_Aa								81.8%		96.7%		88.9%	54.4%	65.1%	88.6%		92.6%		90.5%	50.3%	65.5%
f_na									95.0%	98.1%			97.1%			96.2%	96.2%			96.3%	
f_Ana								84.2%	89.6%	93.9%	85.5%	88.6%	92.5%	88.9%	89.8%	91.5%	89.4%	90.7%	89.5%	90.0%	89.9%

Tabela de generalização de modelos de 2 caracteres com o modelo treinado no eixo y aplicado ao conjunto de dados gerado com o modelo no eixo x, a intensidade da cor refere-se ao percentual de acerto

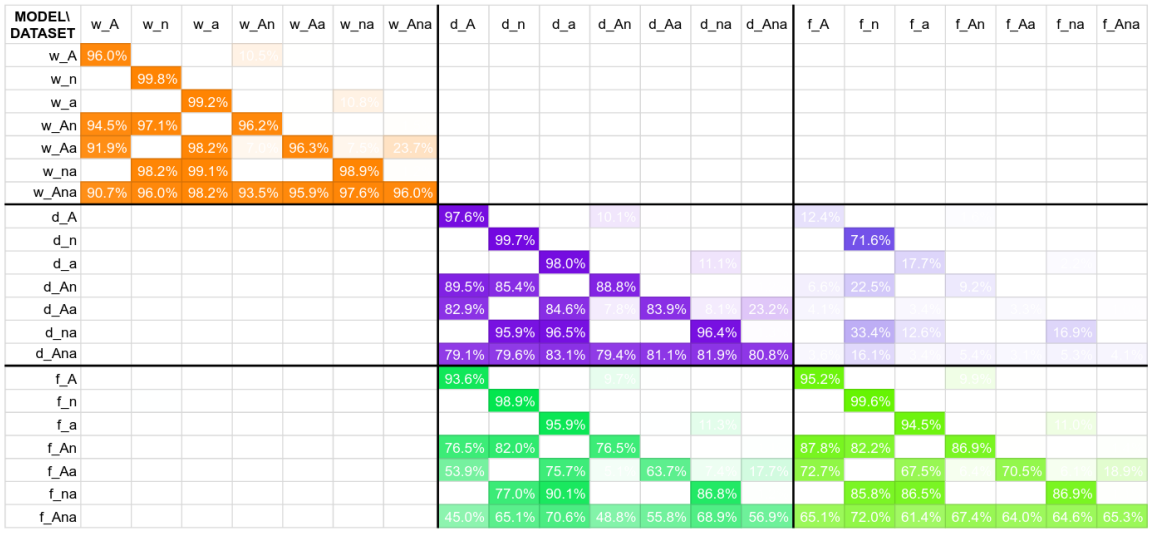
Figura B.1: *Generalizações com n=2*



MODEL\ DATASET	w_A	w_n	w_a	w_An	w_Aa	w_na	w_Ana	d_A	d_n	d_a	d_An	d_Aa	d_na	d_Ana	f_A	f_n	f_a	f_An	f_Aa	f_na	f_Ana
w_A	99.3%			32.6%	7.5%		12.4%														
w_n		99.9%																			
w_a			99.8%		7.5%	32.6%	11.6%														
w_An	98.8%	99.3%		98.9%	27.2%		49.4%														
w_Aa	97.6%		99.2%	27.2%	98.6%	27.5%	49.4%														
w_na		99.7%	99.7%		7.5%	99.7%	11.2%														
w_Ana	97.1%	99.0%	99.2%	97.6%	98.2%	99.1%	98.5%														
d_A								99.2%			32.2%	7.7%		11.4%	42.9%			14.1%	5.8%		6.1%
d_n		12.7%							99.9%							84.1%					
d_a										99.6%		7.2%	33.9%	7.5%			49.3%		5.8%	17.3%	
d_An	5.6%	11.1%		8.7%				95.8%	93.3%		95.2%	6.5%		11.3%	33.7%	54.5%		38.5%			8.6%
d_Aa								93.2%		93.3%	29.1%	93.4%	28.5%	49.7%	28.9%		22.2%	8.6%	24.3%	6.8%	12.8%
d_na									98.8%	99.1%		7.5%	99.1%	11.2%		63.1%	38.8%			44.8%	8.1%
d_Ana								89.4%	90.7%	93.1%	90.2%	91.2%	92.3%	91.3%	25.9%	46.6%	21.8%	30.1%	22.8%	27.3%	25.9%
f_A								98.0%			32.2%	7.5%		11.4%	98.0%			31.7%	5.8%		6.1%
f_n		14.7%							99.6%							99.8%					
f_a										98.7%		7.1%	34.0%	7.5%			96.9%		7.1%	33.4%	
f_An								89.0%	90.7%		88.9%	6.5%		10.6%	93.9%	90.7%		93.2%			12.6%
f_Aa								71.7%		89.8%	22.6%	80.2%	27.9%	42.6%	83.1%		82.6%	24.9%	83.2%	24.7%	43.2%
f_na									85.6%	96.5%		7.5%	93.9%	10.6%		91.6%	93.1%		5.8%	92.8%	10.5%
f_Ana								71.9%	82.7%	85.4%	74.6%	78.2%	84.0%	78.7%	81.3%	86.5%	77.4%	83.2%	79.8%	80.0%	80.9%

Tabela de generalização de modelos de 4 caracteres com o modelo treinado no eixo y aplicado ao conjunto de dados gerado com o modelo no eixo x, a intensidade da cor refere-se ao percentual de acerto

Figura B.2: *Generalizações com n=4*



MODEL\ DATASET	w_A	w_n	w_a	w_An	w_Aa	w_na	w_Ana	d_A	d_n	d_a	d_An	d_Aa	d_na	d_Ana	f_A	f_n	f_a	f_An	f_Aa	f_na	f_Ana
w_A	96.0%			10.5%																	
w_n		99.8%																			
w_a			99.2%			10.5%															
w_An	94.5%	97.1%		96.2%																	
w_Aa	91.9%		98.2%	7.7%	96.3%	7.5%	23.7%														
w_na		98.2%	99.1%			98.9%															
w_Ana	90.7%	96.0%	98.2%	93.5%	95.9%	97.6%	96.0%														
d_A								97.6%			10.1%				12.4%						
d_n									99.7%							71.6%					
d_a										98.0%			11.1%				17.7%				
d_An								89.5%	85.4%		88.8%				5.8%	22.5%		9.2%			
d_Aa								82.9%		84.6%	7.8%	83.9%	5.1%	23.2%	7.1%		12.6%		7.0%		
d_na									95.9%	96.5%			96.4%			33.4%	12.6%			16.9%	
d_Ana								79.1%	79.6%	83.1%	79.4%	81.1%	81.9%	80.8%	7.1%	16.1%	7.1%	9.2%	7.0%	5.8%	7.1%
f_A								93.6%			9.7%				95.2%			9.2%			
f_n									98.9%							99.6%					
f_a										95.9%			11.3%				94.5%			11.0%	
f_An								76.5%	82.0%		76.5%				87.8%	82.2%		86.9%			
f_Aa								53.9%		75.7%	7.8%	63.7%	5.4%	17.7%	72.7%		67.5%	9.2%	70.5%	5.8%	18.9%
f_na									77.0%	90.1%			86.8%			85.8%	86.5%			86.9%	
f_Ana								45.0%	65.1%	70.6%	48.8%	55.8%	68.9%	56.9%	65.1%	72.0%	61.4%	67.4%	64.0%	64.6%	65.3%

Tabela de generalização de modelos de 8 caracteres com o modelo treinado no eixo y aplicado ao conjunto de dados gerado com o modelo no eixo x, a intensidade da cor refere-se ao percentual de acerto

Figura B.3: *Generalizações com n=8*

MODEL\ DATASET	w_A	w_n	w_a	w_An	w_Aa	w_na	w_Ana	d_A	d_n	d_a	d_An	d_Aa	d_na	d_Ana	f_A	f_n	f_a	f_An	f_Aa	f_na	f_Ana
w_A	50.1%																				
w_n		19.3%																			
w_a			85.2%																		
w_An	62.7%	56.2%		66.6%																	
w_Aa	57.9%		69.0%		69.2%																
w_na		76.3%	84.3%			82.2%															
w_Ana	53.5%	57.8%	69.0%	60.7%	67.1%	66.3%	67.2%														
d_A								22.8%													
d_n									38.2%							11.6%					
d_a										23.8%											
d_An								17.2%	15.4%		17.3%										
d_Aa								10.4%		10.5%		10.9%									
d_na									15.8%	17.6%			18.3%								
d_Ana																					
f_A								16.4%							21.6%						
f_n									32.9%							38.7%					
f_a										13.6%							12.4%				
f_An								8.8%	9.8%		9.0%				15.4%	12.6%		15.3%			
f_Aa										7.4%									7.8%		
f_na									8.2%	7.9%			5.6%			8.5%	11.0%			14.4%	
f_Ana																					

Tabela de generalização de modelos de 16 caracteres com o modelo treinado no eixo y aplicado ao conjunto de dados gerado com o modelo no eixo x, a intensidade da cor refere-se ao percentual de acerto

Figura B.4: *Generalizações com n=16*

Referências

[1] Luis von Ahn et al. “CAPTCHA: Using hard AI problems for security”. Em: *International conference on the theory and applications of cryptographic techniques*. Springer. 2003, pp. 294–311 (ver pp. 1, 4, 8, 14).

[2] Darko Brodić e Alessia Amelio. “The CAPTCHA: Perspectives and Challenges: Perspectives and Challenges in Artificial Intelligence”. Em: (2019) (ver pp. 9, 10).

[3] Elie Bursztein, Matthieu Martin e John Mitchell. “Text-based CAPTCHA strengths and weaknesses”. Em: *Proceedings of the 18th ACM conference on Computer and communications security*. 2011, pp. 125–138 (ver p. 16).

[4] Elie Bursztein et al. “The End is Nigh: Generic Solving of Text-based {CAPTCHAs}”. Em: *8th USENIX Workshop on Offensive Technologies (WOOT 14)*. 2014 (ver p. 15).

[5] Kumar Chellapilla e Patrice Simard. “Using machine learning to break visual human interaction proofs (HIPs)”. Em: *Advances in neural information processing systems* 17 (2004) (ver p. 15).

[6] Ian J Goodfellow et al. “Multi-digit number recognition from street view imagery using deep convolutional neural networks”. Em: *arXiv preprint arXiv:1312.6082* (2013) (ver pp. 4, 8).

[7] Rich Gossweiler, Maryam Kamvar e Shumeet Baluja. “What’s up CAPTCHA? A CAPTCHA based on image orientation”. Em: *Proceedings of the 18th international conference on World wide web*. 2009, pp. 841–850 (ver p. 12).

[8] Meriem Guerar et al. “Gotta CAPTCHA’Em all: a survey of 20 Years of the human-or-computer Dilemma”. Em: *ACM Computing Surveys (CSUR)* 54.9 (2021), pp. 1–33 (ver pp. 9, 10, 13, 14).

[9] Meriem Guerar et al. “Invisible CAPPCHA: A usable mechanism to distinguish between malware and humans on the mobile IoT”. Em: *computers & security* 78 (2018), pp. 255–266 (ver p. 12).

[10] Jonathan Holman et al. “Developing Usable CAPTCHAs for Blind Users”. Em: *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*. Assets ’07. Tempe, Arizona, USA: Association for Computing Machinery, 2007, pp. 245–246. ISBN: 9781595935731. DOI: 10.1145/1296843.1296894. URL: https://doi.org/10.1145/1296843.1296894 (ver p. 12).

[11] Ed Kaiser e Wu-chang Feng. “mod kapow: Protecting the web with transparent proof-of-work”. Em: *IEEE INFOCOM Workshops 2008*. IEEE. 2008, pp. 1–6 (ver pp. 12, 15).

[12] Kurt Alfred Kluever e Richard Zanibbi. “Balancing usability and security in a video CAPTCHA”. Em: *Proceedings of the 5th Symposium on Usable Privacy and Security*. 2009, pp. 1–11 (ver p. 12).

[13] Mark D Lillibridge et al. *Method for selectively restricting access to computer systems*. US Patent 6,195,698. 2001 (ver p. 8).

[14] Jonathan Lung. "Ethical and legal considerations of reCAPTCHA". Em: *2012 Tenth Annual International Conference on Privacy, Security and Trust*. IEEE. 2012, pp. 211–216 (ver p. 1).

[15] Rebecca T Mercuri e Peter G Neumann. "Security by obscurity". Em: *Communications of the ACM* 46.11 (2003), p. 160 (ver p. 1).

[16] Thibault Meunier. *Humanity wastes about 500 years per day on CAPTCHAs. It's time to end this madness*. 2021. URL: https://blog.cloudflare.com/introducing-cryptographic-attestation-of-personhood/ (acesso em 10/10/2022) (ver p. 12).

[17] Marti Motoyama et al. "Re:{CAPTCHAs—Understanding}{CAPTCHA-Solving} Services in an Economic Context". Em: *19th USENIX Security Symposium (USENIX Security 10)*. 2010 (ver p. 14).

[18] Moni Naor. "Verification of a human in the loop or Identification via the Turing Test". Em: *Unpublished draft from http://www. wisdom. weizmann. ac. il/˜ naor/PAPERS/human abs. html* (1996) (ver pp. 7, 14).

[19] Fabien Petitcolas. "La cryptographie militaire". Em: *J. des Sci. Militaires* 9 (1883), pp. 161–191 (ver pp. 1, 3).

[20] Claude E Shannon. "Communication theory of secrecy systems". Em: *The Bell system technical journal* 28.4 (1949), pp. 656–715 (ver pp. 1, 4).

[21] Sajad Shirali-Shahreza et al. "Spoken CAPTCHA: A CAPTCHA system for blind users". Em: *2009 ISECS International Colloquium on Computing, Communication, Control, and Management*. Vol. 1. IEEE. 2009, pp. 221–224 (ver p. 12).

[22] Adam Shostack. *Threat modeling: Designing for security*. John Wiley & Sons, 2014 (ver p. 4).

[23] Karen Simonyan e Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". Em: *arXiv preprint arXiv:1409.1556* (2014) (ver p. 17).

[24] Chatpong Tangmanee. "Effects of text rotation, string length, and letter format on text-based captcha robustness". Em: *Journal of Applied Security Research* 11.3 (2016), pp. 349–361 (ver pp. 15, 23).

[25] Reid Tatoris et al. *Announcing Turnstile, a user-friendly, privacy-preserving alternative to CAPTCHA*. 2022. URL: https://blog.cloudflare.com/turnstile-private-captcha-alternative/ (acesso em 10/10/2022) (ver p. 15).

[26] Luis Von Ahn. "Human computation". Em: *Proceedings of the 4th international conference on Knowledge capture*. 2007, pp. 5–6 (ver pp. 1, 8).

[27] Luis Von Ahn et al. "recaptcha: Human-based character recognition via web security measures". Em: *Science* 321.5895 (2008), pp. 1465–1468 (ver p. 8).

[28] Jing Wang et al. "CAPTCHA recognition based on deep convolutional neural network". Em: *Math. Biosci. Eng* 16.5 (2019), pp. 5851–5861 (ver p. 15).

[29] Jonathan Wilkins. *Strong captcha guidelines v1. 2*. 2009 (ver p. 15).

[30] Jeff Yan e Ahmad Salah El Ahmad. "CAPTCHA security: a case study". Em: *IEEE Security & Privacy* 7.4 (2009), pp. 22–28 (ver p. 15).

[31] Guixin Ye et al. "Yet another text captcha solver: A generative adversarial network based approach". Em: *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*. 2018, pp. 332–348 (ver pp. 15, 16).

[32] Yang Zhang et al. “A survey of research on captcha designing and breaking techniques”. Em: *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE. 2019, pp. 75–84 (ver p. 10).