Computers & Security 97 (2020) 101947

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

SoK: Machine vs. machine – A systematic classification of automated machine learning-based CAPTCHA solvers

Antreas Dionysiou*, Elias Athanasopoulos

Department of Computer Science University of Cyprus P.O. Box 20537, 1678 Nicosia, Cyprus

ARTICLE INFO

Article history:
Received 21 November 2019
Revised 30 March 2020
Accepted 22 June 2020
Available online 23 June 2020

Keywords:
CAPTCHA
Web security
Deep learning
Web attacks
Systemization of knowledge

ABSTRACT

Internet services heavily rely on CAPTCHAs for determining whether or not a user is a human being. The recent advances in ML and AI make the efficacy of CAPTCHAs in strengthening Internet services against bots questionable. In this paper, we conduct a systematic analysis and classification of the state-of-the-art ML-based techniques for the automated text-based CAPTCHA breaking problem. The current state and robustness of text-based CAPTCHAs as are utilized by modern Internet applications, against ML-based automated breaking tools, is examined and reported. Our study suggests that ML can be very effective in increasing: (a) accuracy, (b) speed, and (c) abstraction in CAPTCHA solving. Especially, as far as (c) is concerned, ML-based techniques are easier to be applied in different classes of text-based CAPTCHA schemes. To assess the importance of ML in breaking CAPTCHAs, we build our own ML-only classifiers. Surprisingly, an ML-only approach for solving CAPTCHAs is not sufficient. Overall, our study suggests that fundamentally different ways of conducting reverse Turing test, that will be painless for legitimate users (i.e., humans) but at the same time challenging for automated systems (i.e., software), should be considered for ensuring the healthy operation of current Internet services.

1. Introduction

Internet services heavily rely on effectively distinguishing human from machine visitors. Nonetheless, determining whether or not a user is a human being is not a trivial task. In 1950, Alan Turing proposed the Turing Test (Turing, 2009) for testing a machine's ability to exhibit intelligent behavior equivalent to, or indistinguishable from, that of a human. Nowadays, inferring whether or not an activity is the outcome of a human being is a critical task due to exactly the rise of automated software that misuses and corrupts public Internet services (harvesting email addresses, automatically signing up for or making use of web sites, blogs or forums, etc.). In response, the *Completely Automated Public Turing test to tell Computers and Humans Apart* (CAPTCHA) was proposed in 2003 by Ahn et al., 2003. This is a critical technology for defending Internet services against automated interaction. As mentioned in their article (Ahn et al., 2003), *any program that has high success over a CAPTCHA can be used to solve an unsolved Artificial Intelligence (AI) problem*. This means that CAPTCHAs can be used not only for security reasons (i.e., for telling humans and bots apart), but also as a benchmark for AI and Machine Learning (ML) technologies. CAPTCHAs are sometimes called *reverse Turing tests* because they are intended to allow a computer, instead of a human, to determine if a remote client is human or not (Bursztein et al., 2011b). Thus, CAPTCHAs should effectively defend Internet services against the advances in ML and AI.

CAPTCHAs realize a vital security mechanism that effectively eliminates automated interaction with Internet computing services and applications, particularly deterring attackers from abusing online services (Stark et al., 2015). The vast majority of Internet services deploy CAPTCHAs in order to effectively limit bot systems from misusing and/or exploiting their valuable resources. In order to grasp the importance of blocking automated systems from exploiting valuable Internet resources, we can consider many services and applications that, if successfully exploited, benefit attackers, significantly. For instance, consider (a) poisoning an online poll by artificially submitting hundreds of malicious responses, (b) attacking online accounts, such as bank (Li et al., 2010; Zhang and Wang, 2010) or email accounts, by brute-force (i.e., repeatedly attempting a massive amount of different passwords), (c) signing up for large amounts of email or social media accounts, (d) spamming blogs and news sites with dozens of bogus news/comments

* Corresponding author.
E-mail addresses: adiony01@cs.ucy.ac.cy (A. Dionysiou), eliasathan@cs.ucy.ac.cy (E. Athanasopoulos).

https://doi.org/10.1016/j.cose.2020.101947

and search-engine links, and (e) falsifying torrent seed counts and positive feedback in order to lure people into downloading a trojan payload. Furthermore, according to Motoyama et al. (2010), an underground and robust CAPTCHA solving ecosystem has emerged reselling both automated solving technology and real-time human labor to bypass these protections. As they clearly mention in their paper, automated solvers have zero cost, compared to human-based solvers, so ML-based solutions effectively tackling the automated CAPTCHA-breaking problem will dramatically increase this underground economy (Motoyama et al., 2010). Consequently, CAPTCHAs stand as a critical mechanism for effectively protecting Internet computing services and applications from large-scale automated malicious interaction as well as limiting the economic growth of the underground economy selling CAPTCHA-breaking solutions.

With CAPTCHAs in place as a security policy, the user is requested to perform a specific action in order to be claimed by the system as a human (grant access) or a bot (deny access). The action requested by the system is directly related to the CAPTCHA version being used. There are plenty of versions that exist nowadays depending on the type of the requested input. The most popular ones are: (a) text-based CAPTCHAs, (b) Google's No CAPTCHA reCAPTCHA, (c) Google's Invisible reCAPTCHA (which is an update to the previous version of No CAPTCHA reCAPTCHA), (d) simple math problem CAPTCHAs, (e) honeypot CAPTCHAs, (f) social account sign-in CAPTCHAs, (g) time-based CAPTCHAs, (h) confident CAPTCHAs, (i) biometrics CAPTCHAs, and (j) audio CAPTCHAs.

Since the invention of CAPTCHAs, several design techniques, which trade security with usability have been proposed. Their security is not based on their algorithm's secrecy as the algorithm used to create instances of a proposed CAPTCHA scheme must be made public (as *P* implies in CAPTCHA), though it may be covered by a patent. Instead, their security mainly depends on the visual interference effects (Chellapilla et al., 2005) including rotation, waving, adhesion, and overlap. Ideally, a useful CAPTCHA should be solvable by humans at least 80% of the times while programs using reasonable resources should succeed in less than 0.01% of the cases (Chellapilla and Simard, 2004). Text-based CAPTCHAs, whose users include Google, Yahoo, and Wikipedia, block automated systems, which cannot read the distorted characters in the CAPTCHA's graphic representation. In particular, text-based CAPTCHAs require users to type the characters contained in a distorted image, sometimes with the addition of an obscured sequence of letters or digits, that appears on the screen. These puzzles can be trivially generated by a computer program, however detecting machine-wise the actual text embedded in the image, which is fairly visible to a human eye, is not straight-forward. Therefore, text-based CAPTCHAs have been massively deployed as an impediment to large-scale automated interaction with Internet services. Despite several attacks and numerous alternatives that have been proposed for text-based CAPTCHAs, many websites and applications still use them as their main security and authentication mechanism (Ye et al., 2018). These include the majority of the *top-50* popular websites ranked by *alexa.com* as of April 2018, including Google, Microsoft, Baidu, and many others (Ye et al., 2018). Simply making text-based CAPTCHAs hard (see Section 6) is not optimal since this action has a direct negative impact on CAPTCHAs' usability as legitimate users will struggle to solve them.

Breaking a specific CAPTCHA scheme [1] requires a solution to a difficult problem in the field of AI and ML rather than revealing the algorithm itself (which could be done through reverse engineering or other means (Hindle et al., 2008)). The main reason for which attackers automate CAPTCHA solving is the value of automatically collecting resources that can be sold, such as e-mails, accounts, likes, and re-tweets (Motoyama et al., 2010). The remarkable evolution of techniques that make software capable of mimicking Human Visual System (HVS), especially the advances in AI and ML, has given adversaries the necessary tools for breaking sophisticated/improved-security CAPTCHA schemes. Breaking text-based CAPTCHA using ML-related (Bursztein et al., 2011, 2014; Chellapilla and Simard, 2004; Cruz-Perez et al., 2012; Dileep et al., 2017; Fiot and Paucher, 2009; Gao et al., 2013, 2017; Hindle et al., 2008; Hussain et al., 2016; Li et al., 2010; Rui et al., 2013; Stark et al., 2015; Starostenko et al., 2015; Tang et al., 2016; Ye et al., 2018; Zhang and Wang, 2010) or Computer Vision-related (Ahmad et al., 2010; 2011; Chandavale et al., 2009; Gao et al., 2012; 2014; Huang et al., 2010; Mori and Malik, 2003; Moy et al., 2004; Nachar et al., 2015; Sano et al., 2015; Shih-Yu et al., 2008; Yan and Ahmad, 2007; Yan and El Ahmad, 2008) techniques has been excessively studied. A large number of ML algorithms, mainly based on different Artificial Neural Network (ANN) architectures, have been proposed and successfully applied to solving specific text-based CAPTCHA schemes, some of them reaching 100% success rate (Ye et al., 2018). Note that the threshold for which a CAPTCHA scheme to be considered ineffective is 1% recognition success rate (Bursztein et al., 2011b). As such, the recent advances in ML and AI make the efficacy of CAPTCHAs questionable.

In this paper, we perform a systematic classification of the state-of-the-art ML-based techniques used for breaking text-based CAPTCHA schemes by comparing them in terms of *accuracy, performance*, and *generality*. After examining and comparing those techniques, our final conclusions and reasons for why some specific techniques fail, in terms of CAPTCHA's contents recognition accuracy, are given. Also, the current state and robustness of text-based CAPTCHAs, against ML-based automated breaking tools, is examined and reported. Furthermore, we analyze typical end-to-end scenarios for breaking text-based CAPTCHAs identifying the role of ML in the attack process. Moreover, we show that holistic/non-segmentation based approaches incorporating features from systems neuroscience and HVS in general are the state-of-the-art methodologies on trying to solve complex object recognition problems. This research provides (among the rest) insights, guidelines, and suggestions on how to improve the security of text-based CAPTCHAs by offering a series of effective design security features. Non ML-based techniques will not be examined as many of these attacks are hard-coded for a few specific CAPTCHA schemes making the tuning of these attacking methodologies a non-applicable process as the need for heavily expert involvement, labor-intensive, and time-consuming processes is required. Finally, we experimentally show that an ML-only approach is not sufficient on tackling the automated text-based CAPTCHA breaking problem.

For our large-scale evaluation on automated text-based CAPTCHA breaking, we identify the relevant literature by querying the most accredited scholarly databases. In all, we review 51 papers. Each paper is carefully analysed for determining the *ML-related* or *Computer Vision-related* methodology used in each proposed automated text-based CAPTCHA solver. It has to be noted that a quantitative comparison (e.g., comparing the success rates achieved from different solvers) does not impose a direct metric among all reported techniques. Additionally, such a comparison is extremely challenging as almost all methods are applied to particular versions of CAPTCHAs. Nonetheless, a qualitative discussion about the most powerful techniques (i.e., techniques achieving the *highest success rates* as well as requiring the *minimum breaking times*) is a more applicable and effective approach on trying to report the current state and situation of text-based CAPTCHAs.

[1] In this paper, the phrase *CAPTCHA breaking* refers to automatically solving the text-based CAPTCHA problem using a computer program, i.e., recognizing the characters within a text-based CAPTCHA image representation using software.

Table 1
The total number and references of the scientific papers studied regarding automated text–, image–, audio–, video–, and other–based CAPTCHA solvers. Note that our focus is mainly given on text-based automated CAPTCHA solvers focusing on ML-related techniques.

CAPTCHA Category	Number of papers examined	References
Text-based	33	Ahmad et al. (2010, 2011); Bursztein et al. (2014a, 2011b); Chandavale et al. (2009); Chellapilla and Simard (2004); Chen et al. (2018); Cruz-Perez et al. (2012); Fiot and Paucher (2009); Gao et al. (2017, 2012, 2014, 2013); Dileep et al., 2017; Hindle et al. (2008); Huang et al. (2010); Hussain et al. (2016); Li et al. (2010); Mori and Malik (2003); Moy et al. (2004); Nachar et al. (2015); Rui et al. (2013); Sano et al. (2015); Shih-Yu et al. (2008); Stark et al. (2015); Starostenko et al. (2015); Tang et al. (2018, 2016); Yan and Ahmad (2007); Yan and El Ahmad (2008); Ye et al. (2018); Zhang and Wang (2010); Zi et al. (2020)
Image-based	7	Chew and Tygar (2004); Elson et al. (2007); Golle (2008); Goswami et al. (2014); Hernández-Castro et al. (2009); Sivakorn et al. (2016); Zhu et al. (2010)
Audio-based	8	Bock et al. (2017); Bursztein et al. (2011a); Bursztein and Bethard (2009); Darnstdt et al. (2014); Solanki et al. (2017); Soupionis et al. (2009); Tam et al. (2008a,b)
Video-based	2	Shireesha and Gaikwad (2013); Xu et al. (2014)
Other	2	Google (2019); Sivakorn et al. (2016)

1.1. *Contribution & main points*

Here is a list of the **contributions** and **main points** stemming from this paper.

- **Contribution 1.** We review 51 papers for automated text-based CAPTCHA breaking which show that Internet services cannot effectively defend themselves against the advances in ML/AI. Our study suggests that, in addition to pure visual-recognition techniques, ML can be very effective in increasing: (a) *accuracy*, (b) *speed*, and (c) *abstraction* of solution. Especially, as far as (c) is concerned, ML-based techniques are easier to be applied in different *classes* of text-based CAPTCHA schemes. Furthermore, this paper makes a contribution by presenting a *systematic classification* of the related literature which is largely absent in these overview papers. The overall classification scheme for this research and the corresponding literature is presented in Table 2 on page 4. It has to be noted, that this article surveys the most critical related literature covering all related papers published from 2003 to date. This gives us a strong indication about the overall situation regarding text-based CAPTCHAs.
- **Contribution 2.** We further focus on ML-related automated text-based CAPTCHA solvers and compare all relevant papers across different dimensions. Our results suggest that three effective approaches are based on Reinforcement Learning (RL), Generative Adversarial Networks (GANs), and Recursive Cortical Networks (RCNs). Our large-scale evaluation also indicates that techniques not based on ANNs, neither are inspired by systems neuroscience or HVS, are considered weak.
- **Conclusion 1.** Our large-scale evaluation of the related literature clearly shows that the reassessment of CAPTCHAs and their design fundamentals is considered critical. The scientific community should develop new methods and techniques that will be *challenging* for automated solvers, but at the same time *painless* for legitimate users, so that the healthy operation of current Internet services is ensured.
- **Conclusion 2.** Our study suggests that automated text-based CAPTCHA solving is an attractive target for ML. Therefore, we build our own classifiers based solely on ML techniques. Surprisingly, the results are not supportive, highlighting that a *single type ML-only* approach is likely *not enough*.
- **Conclusion 3.** The most effective design security features stemming from an extensive analysis of the related literature are summarized. The different site operators that still deploy text-based CAPTCHAs as their main human or bot authentication mechanism should carefully consider these security features for further enhancing their systems' security.

1.2. *Paper organization*

The rest of this paper is organized as follows. The core automated text-based CAPTCHA breaking aspects are discussed in Section 2. In Section 3, we discuss the state-of-the-art pre-processing techniques that take place in automated text-based CAPTCHA breaking emphasizing on DL-based pre-processing techniques and methods. In Section 4, we stress the role of advances in ML for CAPTCHA breaking and, in Section 5, we argue that the use of ML algorithms on directly recognizing the text contained in a CAPTCHA is unlikely to be enough for successful attacks. Later on, in Section 6, we discuss some generic design principles stemming from this paper, that can contribute to the design and implementation of more robust text-based CAPTCHA schemes. We discuss our future research directions in Section 7 and we conclude in Section 8.

2. CAPTCHA breaking core aspects

Text-based CAPTCHAs are the most widely used ones (Ye et al., 2018), however, several different schemes exist. We therefore group the different automated CAPTCHA solvers using five (5) main categories (shown in Table 1): (a) Text-based CAPTCHAs, (b) Image-based CAPTCHAs, (c) Audio-based CAPTCHAs, (d) Video-based CAPTCHAs, and (e) Other. The *Other* category refers to CAPTCHA schemes that minimize (e.g., reCAPTCHA) or totally omit (e.g., invisible reCAPTCHA) the user input. Having this specific grouping in place, we proceed and discuss how CAPTCHAs are automatically solved (Section 2.1), the role of ML in the process (Section 2.2), and alternatives to automatic solving through human involvement (Section 2.3).

2.1. *Automated text-based CAPTCHA breaking*

The process of breaking text-based CAPTCHAs can be divided into a number of phases/stages. The most common are: (a) *pre-processing*, (b) *segmentation*, and (c) *recognition*. Furthermore, all deployed methodologies can be classified into one of the two categories, namely: (a) methods based on segmentation, and (b) holistic approaches that do not employ segmentation. We now discuss the rationale behind this classification.

Chellapilla and Simard (2004) were the first to propose an automated attack on text-based CAPTCHA schemes by leveraging ML. Nonetheless, the segment (i.e., isolating characters contained in the CAPTCHA) and then recognize the characters approach was not new. Plamondon and Srihari (2000) presented a survey of holistic- and segmentation-based approaches for handwriting recognition in

Table 2
Comparative evaluation and classification of the related literature. The related literature is divided into two main categories namely: (a) *Non ML-related*, and (b) *ML-related* techniques, also sorted in ascending order based on the publication year. Columns represent different techniques and phases that the studied methodologies may be based on (✔) or not (–). The symbol | denotes that the specified methodology deploys two kinds of techniques for automated text-based CAPTCHA breaking. The following encoding holds for the table shown below: SP = *Supervised Learning*, ANNs = *Artificial Neural Networks*, CV = *Computer Vision*, OR = *Object Recognition*, IP = *Image Processing*, KNN = *K-Nearnest Neighbors*, UL = *Unsupervised Learning*, RCN = *Recursive Cortical Network*, OCR = *Optical Character Recognition*, LSTM-RNN = *Long-Short Term Memory Recurrent Neural Network*, SVM = *Support Vector Machine*, HMM = *Hidden Markov Model*, MLP = *Multi-Layer Perceptron*, CNNs = *Convolutional Neural Networks*, LGF = *Log-Gabor Filters*, AI = *Artificial Intelligence,* and NA = *Not Applicable*.

			Segmentation Phase		**Recognition Phase**			
Category	**Pre-processing**	**Dictionary-Based**	**Segmentation-Based**	**Segmentation Method**	**Recognition Method**	**ML-Based Recognition Method**	**Year**	**Reference**
Non ML-related	–	✔	–	–	CV/OR	NA	2003	Mori and Malik (2003)
	✔	✔	–	–	CV/OR		2004	Moy et al. (2004)
	✔	✔ | –	✔	CV	IP/PR		2007	Yan and Ahmad (2007)
	✔	–	✔	CV	–		2008	Yan and El Ahmad (2008)
	✔	–	✔	IP	–		2008	Shih-Yu et al. (2008)
	✔	–	✔	IP | Heuristics	–		2010	Ahmad et al. (2010)
	✔	–	✔	IP	–		2010	Huang et al. (2010)
	✔	–	✔	IP | Heuristics	IP/Heuristics		2009	Chandavale et al. (2009)
	✔	✔ | –	✔	PR/Heuristics	CV/OCR		2011	Ahmad et al. (2011)
	✔	–	✔	IP/PR	CV/OCR		2012	Gao et al. (2012)
	✔	–	✔	IP/PR	CV/OCR		2014	Gao et al. (2014)
	✔	–	✔	IP/PR & Heuristics	AI/Fuzzy Logic		2015	Nachar et al. (2015)
	✔	–	✔	PR/HMM	PR/HMM		2015	Sano et al. (2015)
ML-related	✔	✔ | –	✔	IP/PR	ML/SL/ANNs	CNNs	2005	Chellapilla and Simard (2004)
	✔	–	✔	CV | ML/UL/K-Means	IP | ML/SL	KNN	2008	Hindle et al. (2008)
	✔	–	✔	IP | Heuristics	ML/SL	SVMs & Active Learning	2009	Fiot and Paucher (2009)
	✔	–	✔	ML/UL/K-Means	IP/PR	–	2010	Li et al. (2010)
	✔	–	✔	IP	ML/SL	KNN	2010	Zhang and Wang (2010)
	✔	✔ | –	✔	IP	ML/SL	KNN | SVMs	2011	Bursztein et al. (2011b)
	✔	–	✔	IP | Heuristics	ML/SL	SVMs	2012	Cruz-Perez et al. (2012)
	✔	–	✔	IP | Heuristics	ML/SL/ANNs	CNN	2013	Gao et al. (2013)
	✔	–	–	–	ML/SL/ANNs	LSTM-RNN	2013	Rui et al. (2013)
	–	–	✔	CV | IP/Heuristics	ML/SL	KNN	2014	Bursztein et al. (2014a)
	✔	–	✔	CV | IP	ML/SL	SVMs	2014	Starostenko et al. (2015)
	✔	–	–	–	ML/SL/ANNs	CNN & Active Learning	2015	Stark et al. (2015)
	✔	–	–	–	ML/SL/ANNs	MLP	2016	Hussain et al. (2016)
	✔	–	–	–	LGF & ML/SL	KNN	2016	Tang et al. (2016)
	–	–	✔	ML/SL/RCN	ML/SL	RCN	2017	Dileep et al., 2017
	✔	–	✔	CV | IP	ML/SL/ANNs	CNN	2017	Gao et al. (2017)
	✔	–	–	–	ML/SL/ANNs	CNN & Active Learning	2018	Ye et al. (2018)
	✔	–	✔	ML/SL/CNN	ML/SL/ANNs	CNN	2018	Tang et al. (2018)
	✔	–	✔	IP | ML/SL/KNN	ML/SL/ANNs	CNN	2018	Chen et al. (2018)
	–	–	–	–	ML/SL/ANNs	CNN & LSTM	2019	Zi et al. (2020)

early 2000s giving an in-depth explanation of the most common phases along with a brief definition for each one of them. Observing Plamondon and Srihari's paper (Plamondon and Srihari, 2000), the segmentation process is described as *breaking a given input into meaningful units* which in our case is breaking a text-based CAPTCHA into a set of individual characters. Furthermore, the definition given in Plamondon and Srihari's paper (Plamondon and Srihari, 2000) for the recognition phase is described as *the task of transforming a language represented in its spatial form of graphical marks into its symbolic representation*.

Moreover, as Bursztein et al. (2011b) suggest, the choice of classifier does not matter greatly because most of the work is done before recognition. Thus, the main concern for the recognition phase is to be as stable and as fast as possible. Based on this statement and the insights gathered during our evaluation, we divide the automated text-based CAPTCHA solvers in two major categories, namely *segmentation-* and *non-segmentation based*. The first one tries to segment the text-based CAPTCHA into a set of individual characters before executing the recognition phase. The second one predicts the text contained in a text-based CAPTCHA scheme, holistically, without relying on specific heuristics or pattern-recognition algorithms for individual character detection. More details regarding these two major approaches are given in Section 4.

2.2. Applying ML on CAPTCHA breaking

According to Ahn et al., 2003, breaking CAPTCHAs is a difficult problem in the field of AI and ML. This is mainly because improved CAPTCHA schemes cause segmentation-based breaking techniques to fail on detecting the individual characters contained in an image. However, humans are still able to solve improved CAPTCHA schemes relatively easily. As a result, the scientific community supports that ML-based techniques are attractive for automatically solving text-based CAPTCHAs.

For example, as mentioned in Ye et al. (2018), the proposed ML-based technique *outperforms four state-of-the-art text-based CAPTCHA solvers by not only delivering a significantly higher accuracy on all testing schemes, but also successfully attacking schemes where others have zero chance*. The proposed ML-based technique used in Ye et al. (2018), namely Convolutional Neural Network (CNN), has become the state-of-the-art ANN-based technique for solving complex image and object recognition related problems. CNNs are a class of deep, feedforward ANNs that have successfully been applied to analysing visual imagery (Dionysiou et al., 2018). CNNs were inspired by the HVS, where individual cortical neurons respond to stimuli, only in a restricted region of the visual field, known as the receptive field. CNNs have been exclusively designed to face feature extraction as well as immense complexity sequence and pattern recognition problems (Dionysiou et al., 2018; Simonyan and Zisserman, 2014; Srinivas et al., 2016).

In order to understand the usefulness of ML in solving text-based CAPTCHAs, we have reviewed 51 related papers. Table 2 shows the major characteristics and attributes of each paper studied. Each methodology is initially classified as *ML-* or *Non ML-related*. Each column in Table 2 represents a phase or a technique used by the specified methodologies. Each cell contains the ✔ symbol if the corresponding automated CAPTCHA solver adopts that phase/technique or the – symbol otherwise. In case the proposed methodology deploys a segmentation or a recognition technique, the exact specific algorithm used, as well as the field/subfield that belongs to, is given. Note that the attack success rates for each methodology are not given as all the examined techniques manage to break the 1% barrier for a text-based CAPTCHA scheme to be considered ineffective (Bursztein et al., 2011b).

2.3. Software-based vs human-based CAPTCHA solvers

Motoyama et al. are the first and only, to the best of our knowledge, to study the various CAPTCHA solving services in an economic context (Motoyama et al., 2010). They mention that as a robust CAPTCHA solving ecosystem has emerged, reselling both automated solving technology and real-time human labor for bypassing these protections should be evaluated in purely economic terms. In particular, evaluating the market price of a solution versus the monetizable value of the asset being protected. Nonetheless, their focus is mainly given in human labor-based solvers analyzing the behavior and dynamics of human-based CAPTCHA-solving service providers, their price performance, and the underlying labor markets driving this economy. They mention that the cost of such CAPTCHA-solving services is as low as *$1 per 1,000 CAPTCHAs* and they state that this cost will continue to dramatically decline.

Another major advantage of human-based solvers, compared to automated solving tools, is the *adaptability* issue. Humans can easily adapt, with minimum or no training experience, on different CAPTCHA schemes. In their study, Motoyama et al. (2010) have concluded that a specific solving service, namely *ImageToText*, displayed a remarkable adaptability when solving the Asirra CAPTCHA. In fact, the service was successful on average 39.9% of the time, which was a high percentage back then. However, automated solvers have $0 cost, can be general applicable, if correctly designed, and nowadays they demonstrate extremely high accuracy rates on difficult CAPTCHA schemes. Motoyama et al. paper's main purpose though is to demonstrate that CAPTCHAs should be viewed, not only as a technological impediment, but also as an *economic impediment* to an attacker; a fact, that is now widely accepted by the scientific community. For the rest of the paper, we focus only on automated CAPTCHA solving using algorithms and not human labor.

3. Pre-processing methods

Most, if not all, automated CAPTCHA solvers require some image pre-processing. This phase is usually performed before segmentation (if segmentation phase exists) and recognition. For most of the existing ML-based CAPTCHA solvers this phase is critical. Pre-processing highlights the information related to the characters in the image and weakens or eliminates any interfering information (Bursztein et al., 2011; Dileep et al., 2017; Ye et al., 2018). In general, the different pre-processing techniques aim to reformat the CAPTCHA's image representation in a simpler form. This form can be effectively analysed at a later segmentation or recognition stage. The pre-processing phase is optional and in current state-of-the-art methods includes mainly *image binarization, image thinning*, and *denoising* (Bursztein et al., 2011; Dileep et al., 2017; Ye et al., 2018). Additionally, the pre-processing phase may be based on Deep Learning (DL). In this section, we review and briefly explain each of the aforementioned pre-processing techniques.

3.1. Image binarization

Image binarization is the process of converting a digital image to a binary one. A binary image has only two possible values for each pixel (i.e., black or white). It has been experimentally shown that image recognition techniques work better and faster on binary features so binarizing the CAPTCHAs in black and white is recommended in most of the times (Bursztein et al., 2011b).

3.2. Image thinning

Image thinning, also known as skeletonization, highlights characters' contour without affecting their adhesion. It is a morphological operation that deletes black foreground pixels iteratively layer by layer until one-pixel width skeleton is obtained (Abhishek, 2017). There are different types of thinning algorithms, some of them including: (a) sequential, (b) parallel, and (c) non-iterative (Lam et al., 1992). This technique is effectively used to simplify any subsequent processing. However, as mentioned in Abhishek (2017), no thinning algorithm that fulfils the basic set of requirements exists yet.

3.3. Image noise removal

Most text-based CAPTCHA schemes use noise and interference lines in order to resist automated breaking. This distortion (noise, interference lines, etc.), if not removed, prevents the recognition technique used from achieving higher success rates. As a result, most of the state-of-the-art ML-based CAPTCHA breaking methodologies try to *de-noise*[2] the image representation before issuing the recognition phase. The most common de-noising methods are: (a) filter-based in the spatial domain method, (b) Gibbs and Hough

[2] In this paper, the term *de-noise* means removing all kind of noise and interference contained in a text-based CAPTCHA image representation.

transform-based method (Lutton et al., 1994), (c) morphology-based method, (d) connected component-based method, and (e) wavelet transform-based method. Nonetheless, each de-noising technique should be carefully deployed considering the type of distortion included in a CAPTCHA scheme. It has been shown experimentally that removing the distortion from a CAPTCHA image before issuing the recognition phase leads to higher success rates, rather than directly issuing the recognition phase on the distorted image (Ye et al., 2018).

3.4. Deep learning

Many proposed ML-based techniques successfully break specific text-based CAPTCHA schemes (Bursztein et al., 2011, 2014; Dileep et al., 2017; Stark et al., 2015; Ye et al., 2018). These attacks have led the scientific community to the development of more robust text-based CAPTCHAs that include advanced security features, like occluding lines and distorted hollow fonts. These improved schemes cause past techniques that are heavily based on pre-processing (Gao et al., 2013; Yan and El Ahmad, 2008) to fail. The state-of-the-art, in terms of attack success rates, automated text-based CAPTCHA solvers use DL for pre-processing (i.e., removing noise, occluding lines from the background, filling hollow parts of characters, and widening and standardizing the gap between two characters).

For example, the DL-based Pix2Pix image-to-image translation framework (Isola et al., 2017) has been used in Ye et al. (2018). The Pix2Pix framework transforms an image from one style to another. In Ye et al. (2018)'s case, the images to be transformed are the distorted CAPTCHA images (i.e., images with security features applied). As mentioned in Ye et al. (2018), the pre-processing model deployed is a Generative Adversarial Network (GAN) and consists of a generator and a discriminator. GANs were proposed by Goodfellow et al. (2014) in 2014 for estimating generative models via an adversarial process. According to Ye et al. (2018), their goal was to train a generator on removing the security features and standardizing the font style. In contrast, the discriminator tries to distinguish the pre-processed from the clean CAPTCHAs that are produced by their CAPTCHA synthesizer, which is also based on GANs. This deployed pre-processing GAN becomes better and better on removing security features, as training time passes. Deploying the fully trained GAN on distorted images leads to *clean* CAPTCHAs. Furthermore, the discriminator is trained on recognizing even the smallest, in terms of impact, security features of the text-based CAPTCHAs given. This *adversarial* process of fine cleaning the distorted CAPTCHA images using GANs terminates when the discriminator fails to identify more than 5% of the generated CAPTCHAs from the clean counterparts. In other words, GAN-based pre-processing techniques are able to remove all the distortion, noise, and interference from the CAPTCHA images in 95% of all cases.

For the time being, the *DL-based pre-processing* methods are considered to be the strongest and most effective pre-processing techniques as they manage to filter out almost all the distortion from text-based CAPTCHA images. Referring to GANs, Facebooks AI research director Yann LeCun called adversarial training *the most interesting idea in the last 10 years in ML* (Avyuct and Avyuct, 2019). GANs seem to be a great step in deploying DL techniques as they essentially ease the use of DL-based approaches by: (a) effectively updating the weights of the ANN-based *generator* based on the recognition ability of another ANN-based *discriminator*, and (b) constructing a large dataset (needed for training DL-based techniques) from synthetic, but similar to real ones, CAPTCHAs. Nonetheless, for GAN-based techniques to be deployed, a small amount of real-world training examples, in our case real text-based CAPTCHAs, is required (Ye et al., 2018).

4. ML-based CAPTCHA solvers

CAPTCHA solvers can be augmented with ML techniques for increasing effectiveness and efficiency (Bursztein et al., 2011, 2014; Dileep et al., 2017; Gao et al., 2017; Stark et al., 2015; Ye et al., 2018). Additionally to ML, a particular solver may incorporate segmentation techniques for making character recognition easier and achieving higher success rates.

A generic sequential approach, based on segmentation (Bursztein et al., 2011b; Chellapilla and Simard, 2004; Gao et al., 2017) and used by many state-of-the-art automated text-based CAPTCHA solvers, is the three-stage approach. In particular, this method firstly pre-processes the CAPTCHA to make it easier to analyse, secondly, a segmentation phase takes place, where the solver tries to segment the CAPTCHA into chunks that contain exactly one character, and thirdly, the recognition phase kicks in, where a classifier, based on ML, is used to recognize which character is contained in each chunk. For such segmentation-based approaches it has become well established that a CAPTCHA's ability to withstand automated solving lies in the difficulty of segmenting the image into individual characters, rather than recognizing the characters themselves (Bursztein et al., 2014a) (individual character recognition can be solved with high success rates such as 99.1% precision LeCun et al., 1995b).

Several ML-based solvers are not based on segmentation (Bursztein et al., 2014; Dileep et al., 2017; Ye et al., 2018). Such techniques aim at directly predicting the text contained in a given CAPTCHA, sometimes after issuing some CAPTCHA pre-processing techniques. Omitting segmentation is much more attractive for developing ML-based CAPTCHA solvers as in many cases the use of specific security features (e.g., collapsing or lines crossing upon characters) hardens the development of algorithms for reliably segmenting specific CAPTCHA schemes. Thus, ML-based solvers simplifying the CAPTCHA breaking process by limiting the number of stages required are preferred (Bursztein et al., 2014; Dileep et al., 2017; Ye et al., 2018).

We now thoroughly discuss the two major approaches for developing ML-based automated CAPTCHA solvers while also explaining the reason why the text-based CAPTCHA breaking problem is an attractive target for ML technologies.

4.1. Segmentation-based approach

Trying to break a CAPTCHA using segmentation techniques (i.e., splitting the CAPTCHA image into segments that contain individual characters) can only be effective on non-adherent CAPTCHAs as in this case individual characters are obtained using vertical projection and connected component with good effect. In particular, the success rates achieved for non-adherent character CAPTCHAs range from 57% to 100% (Mori and Malik, 2003; Moy et al., 2004; Ye et al., 2018). Nevertheless, in case of adherent characters, these techniques have little success as they fail to detect the characters contained in a text-based CAPTCHA scheme, due to overlapping characters, lines drawn over all characters, background confusion, etc. Deploying more and more sophisticated anti-segmentation security features[3] in CAPTCHA schemes makes the process of extracting individual characters harder and harder. Until 2011, the success rates of automated solvers for adherent characters CAPTCHAs were generally low, with only a few higher than 80% (Bursztein et al., 2011b). The failure of segmentation-based techniques on CAPTCHA schemes containing adherent characters made the scientific community to realise that more sophisticated techniques and method-

[3] In this paper we use the term *anti-segmentation security features* to describe image/text manipulations that aim at preventing the solver from splitting the CAPTCHA into individual characters.

ologies had to be deployed for effectively tackling this hard to be beaten security feature.

Early state-of-the-art CAPTCHA solvers were heavily based on segmentation and used in most cases the aforementioned three-stage approach consisting of a *pre-processing*, a *segmentation*, and a *classification/recognition* stage (Bursztein et al., 2011b; Chellapilla and Simard, 2004; Gao et al., 2017). In 2011, Bursztein et al. (2011b) proposed Decaptcha. Decaptcha was a generic attack adopting an extended five-stage approach consisting of: (a) pre-processing, (b) segmentation, (c) post-segmentation, (d) recognition, and (e) post-processing phases. During the segmentation phase, the authors involve various segmentation techniques, the most common being the CFS (Color Filling Segmentation) (Yan and El Ahmad, 2008). At post-segmentation phase, the segments are processed individually to make recognition easier. At recognition phase, they use two relatively straight-forward ML algorithms for predicting each segment's character. As far as the ML algorithms used are concerned the authors are explicit: *the choice of classifier does not matter greatly because many modern classifiers perform strikingly well (i.e., they achieve 97% - 99.5% success rates) on the MNIST dataset* (Bursztein et al., 2011b). They also mention that for a CAPTCHA scheme to be considered insecure, automated CAPTCHA solvers have to reach only 1% precision (Bursztein et al., 2011b). Finally, Bursztein et al.'s methodology, inspired from previous works (i.e., Ahmad et al., 2011; Mori and Malik, 2003), includes a post-processing phase. Post-processing techniques perform specific actions after issuing the recognition phase on the results returned. For example, some CAPTCHA breaking techniques use a dictionary to dismiss unlikely words (Mori and Malik, 2003), an approach which is not considered to be very optimistic as it is usually trivial for the defender to tune its text-based CAPTCHA scheme for mitigating those attacks. Furthermore, Decaptcha uses spell checking on the classifier's output for a specific text-based CAPTCHA scheme that uses dictionary words.

However, Decaptcha (Bursztein et al., 2011b) failed to break the reCAPTCHA 2011 scheme, whereas simpler attacks that are not based on ML can be successful (Tang et al., 2016). Additionally, the proposed attacks implemented by Decaptcha (Bursztein et al., 2011b) cannot break hollow CAPTCHAs, while non segmentation-based approaches can (Gao et al., 2013). Non segmentation-based techniques (Bursztein et al., 2014a; Ye et al., 2018) achieve higher success rates on all CAPTCHA schemes compared to Decaptcha. Due to the failure of segmentation-based algorithms on reliably segmenting the characters in a text-based CAPTCHA scheme, non segmentation-based approaches have now become the state-of-the-art for developing automated CAPTCHA solvers.

4.2. Non segmentation-based approach

Non-segmentation based techniques rely on directly predicting the text contained in a CAPTCHA scheme, sometimes after some CAPTCHA pre-processing. It is obvious that the success rates for these methods rely on the recognition technique's ability to observe the contents from a distorted CAPTCHA image. In 2003, Mori and Malik (2003) proposed some object recognition techniques based on shape context matching for the CAPTCHA breaking problem. The following year, Moy et al. (2004) proposed two distortion estimation techniques (again for object recognition) for breaking two specific CAPTCHA schemes named EZ-Gimpy and Gimpy-r. However, back at that time, techniques based on segmentation have been shown to achieve higher success rates than the state-of-the-art non-segmentation based techniques. Nowadays, many techniques based on DL (Isola et al., 2017; Pan and Yang, 2010) have been successfully applied to CAPTCHA breaking (Bursztein et al., 2014; Dileep et al., 2017; Stark et al., 2015; Yan and El Ahmad, 2008; Ye et al., 2018), after facing the two major difficulties that prevented the scientific community from deploying such techniques: (a) the hardware performance limitations, and (b) the lack of large datasets required for training DL-based classifiers.

Modern CAPTCHAs employ several anti-segmentation security features forcing the segmentation of the CAPTCHA image to individual image-characters to fail. Amongst the many different DL-based methodologies that have been proposed for automated text-based CAPTCHA solving, most of them utilize CNNs in an attempt to, not only recognize the text contained in a CAPTCHA scheme (Bursztein et al., 2014; Dileep et al., 2017; Stark et al., 2015; Ye et al., 2018), but also for filtering out the noise contained in the puzzle (Ye et al., 2018). In Sections 4.2.1, 4.2.2 and 4.2.3 we discuss in detail the three *state-of-the-art*, in terms of effectiveness, efficiency and generality, approaches for text-based CAPTCHA breaking, that follow the non-segmentation based approach (Bursztein et al., 2014; Dileep et al., 2017; Ye et al., 2018). The examination of each paper will be done in ascending order based on the publication date. As we show in our evaluation, the models become more and more biologically realistic as they try to implement CAPTCHA breaking techniques heavily based on and inspired from HVS (Thorpe et al., 1996). As the HVS is the ultimate solution on trying to solve visual recognition problems, the scientific community is trying to develop recognition algorithms imitating basic structural parts of HVS (Dileep et al., 2017). For example, the probabilistic generative model for vision proposed in Dileep et al., 2017's paper (namely RCN) is heavily inspired by systems neuroscience and demonstrates excellent generalization, utilizing DL-based techniques for handling recognition, segmentation and reasoning in a unified way.

We choose to give greater focus on the following three text-based CAPTCHA solvers as they seem to have great performance achieving surprisingly *high success rates on difficult text-based CAPTCHA schemes* (i.e., CAPTCHA schemes deploying state-of-the-art security features) as well as being *generic* enough at the same time (i.e., achieving high success rates on radically different text-base CAPTCHA schemes).

4.2.1. Bursztein et al.'s reinforcement-based approach

Bursztein et al. (2014a) paper's title is *The End is Nigh: Generic Solving of Text-based CAPTCHAs*. The authors are directly stating that the life of text-based CAPTCHAs is coming to an end. This statement is due to the fact that their automated CAPTCHA solver (as well as many other techniques (Bursztein et al., 2011; Dileep et al., 2017; Moy et al., 2004; Ye et al., 2018)) based on Reinforcement Learning (RL) achieves success rates $\geq$ 1% (threshold for which a CAPTCHA scheme is considered to be ineffective (Bursztein et al., 2011b)). Nevertheless, Bursztein et al.'s proposed technique achieved recognition accuracy results ranged from 3.67% to 55.22% on specific text-based CAPTCHA schemes, which are considered to be extremely high for such security oriented technologies. RL (Kaelbling et al., 1996) is considered one of the core three ML paradigms, alongside supervised and unsupervised learning. In RL, an agent with initially no-knowledge acts in a specified environment and receives a reward or penalty for every action performed in each step. In this way, after a certain amount of epochs the agent acts in a certain way to maximize some notion of cumulative reward (short-term or long-term). In each step, the agent has a dilemma to solve which involves: (a) exploring the environment, or (b) exploitation of its current knowledge taking an action that has high expected reward (Kaelbling et al., 1996).

Bursztein et al. (2014a) introduce a novel method of solving text-based CAPTCHAs in a single step, that uses ML to attack the segmentation and the recognition problems simultaneously (Bursztein et al., 2014a). They clearly mention that when segmentation and recognition techniques are jointly performed allows for improved information and context exploitation that couldn't be done if those phases were issued sequentially. Their technique removes any hand-crafted components generalizing their approach to new text-based CAPTCHA schemes. Their algorithm's main idea is to deploy ML to assign a score to all possible ways to segment a text-based CAPTCHA and finally select the segmentation giving the highest score. Their algorithm is composed of four components namely: (a) cut-point detector; finds all the potential ways to segment a CAPTCHA, (b) slicer; extracts the segments and combining them into a graph, (c) scorer; assigns a recognition confidence score to each segment, and (d) arbiter; processes the scores to determine what are the most likely characters contained. As their algorithm produces a huge number of segments, which most of them are garbage, they use RL to ask humans to annotate segments that have been misclassified and then their algorithm learns from the feedback. Using this methodology they managed to break many prominent real-world text-based CAPTCHA schemes (e.g., Baidu 2011 and 2013, CNN, eBay, etc.) that make use of both negative kerning and occluding lines, without tuning or making any modification to the algorithm. They conclude their article underlying that combining segmentation and recognition is the next evolution of text-based CAPTCHA solving while also suggesting that significant effort may be needed to rethink the way we perform the reverse Turing tests.

4.2.2. Ye et al.'s holistic GANs-based approach

Ye et al.'s paper, starts by saying that despite the several attacks proposed, text-based CAPTCHAs are still being widely used as a security mechanism (Ye et al., 2018). One of the main reasons that text-based CAPTCHAs are still widely deployed by site operators is that prior automated attacks are scheme-specific and require a labor-intensive and time-consuming process to construct. In this way, prior attacks achieve zero success when site operators tweak their text-based CAPTCHAs design. As a result, the authors proposed a generic, yet effective, text-based CAPTCHA solver based on GANs (Goodfellow et al., 2014; Ye et al., 2018). GANs are composed of two ANNs, namely *generator* and *discriminator*. The *generator* ANN generates new data instances, whereas the *discriminator* ANN evaluates them for authenticity (i.e., the discriminator decides whether each instance of data belongs to the actual training dataset or not). GANs belong to the set of generative models which means that they are able to produce/generate new/synthetic samples that are very similar to real ones. Thus, the need for a large volume of training data is also eliminated as they again deploy GANs to construct *similar* to real-world text-based CAPTCHA training samples. After creating their large synthetic corpus, and before presenting a CAPTCHA image to a solver, they deploy and train a GAN-based pre-processing model (called Pix2Pix (Isola et al., 2017)) to erase the text-based CAPTCHAs' security features. After that, they train their base solver (i.e., a CNN) using their synthetic corpus created by their GAN-based CAPTCHA synthesizer. More specifically, their CNN-based solver takes in a preprocessed text-based CAPTCHA image and outputs the corresponding characters.

Finally, they apply transfer learning to refine the CNN-based solver using a small set of manually labeled text-based CAPTCHAs. As one can easily notice, Ye et al. employ an holistic all-ML-based methodology using GANs for CAPTCHA synthesis and preprocessing as well as a CNN for recognizing the text contained in a preprocessed text-based CAPTCHA (i.e., clean CAPTCHA – without any security features). Their approach seems to have an advantage over the other computer vision- and ML-related CAPTCHA breaking techniques as it requires significantly fewer real-world CAPTCHA samples to construct and train their automated solver. Furthermore, their method requires minimum human involvement when targeting a new CAPTCHA scheme as their attack can be easily adjusted, demonstrating in this way their attack's general applicability. As they clearly state, applying advanced security features on CAPTCHA schemes only make it difficult for legitimate users, failing to stop automated programs. Furthermore, their holistic DL-based technique succeeds to solve all 33 tested CAPTCHA schemes, including 11 schemes that are currently being used by 32 of the *top-50* popular websites including Microsoft, Wikipedia, eBay and Google. Finally, such holistic ML-based approaches seem to be the most powerful solutions for the automated text-based CAPTCHA breaking problem demonstrating general applicability and high efficiency. For example, Ye et al.'s (Ye et al., 2018) method can solve a text-based CAPTCHA scheme within 0.05 seconds using a desktop GPU.

4.2.3. Dileep et al.'s RCN-based approach

Dileep et al., 2017 propose a model that is heavily inspired by human visual intelligence and systems neuroscience. As mentioned in their paper, they introduce *a probabilistic generative model for vision in which message-passing based inference handles recognition, segmentation and reasoning in a unified way*. In this way, their model has the ability to learn from few examples and generalize to dramatically different situations. Their proposed hierarchical model, called Recursive Cortical Network (RCN), fundamentally breaks the defence of the modern text-based CAPTCHA schemes, without integrating any CAPTCHA-specific heuristics in their model's methodology. Their approach mainly tries to mimic HVS's behavior to avoid the limitations of the DL-based techniques that require millions of labelled examples to be trained on, and computer vision algorithms that usually make use of hand-crafted style-specific heuristics to segment out the characters contained in a text-based CAPTCHA scheme. For these reasons, their model is able to generalize beyond the training distribution to new text-based CAPTCHA schemes without explicit training (i.e., as humans do).

In RCN, objects are modelled as a combination of contours and surfaces. Contours appear at the boundaries of surfaces; surfaces being modelled using a Conditional Random Field (CRF). This factored representation of contours (shape) and surfaces (appearance) allows their model to recognise objects with dramatically different shapes and styles without being trained exhaustively on every possible object. In contrast with Dileep et al.'s RCNs, a state-of-the-art CNN (Goodfellow et al., 2013) used for multi-digit number recognition from street view imagery required a way too larger (50,000-fold larger) training set of actual CAPTCHA images while also being less robust to any adjustments on the input (e.g., string lengths not present during training, and/or minor perturbations to the spacing of characters). The authors clearly state that the incorporation of RCN's inductive biases from systems neuroscience (i.e., RCN's object and part-based compositionality) into neural network models as well as combining RCN with Bayesian Program Learning (BPL) (Lake et al., 2015) seem to be interesting research directions that can lead to robust and generalizable ML models with high data efficiency. Finally, the authors conclude their article stressing the fact that site operators should deploy more robust mechanisms for blocking automated bot attacks. This approach tries to incorporate HVS's capabilities on a probabilistic model to create in this way an effective and efficient object recognition model. Moreover, their technique does not adopt the serial segment-then-recognize tactic but instead it tackles segmentation and recognition in a unified way. The authors, throughout their paper try to pass the message that the path to the ultimate solution in such problems lies in the integration of mammalian optical cortex mechanisms to any proposed model. Thus they inspire the research on improved models of cortical circuits (Dileep and Hawkins, 2009; Litvak and Ullman, 2009) and techniques that combine the power of ANNs and structured probabilistic models toward general AI systems (Dileep et al., 2017).

5. Solving CAPTCHAs using ML only

All of the techniques discussed in this paper make use of some kind of pre-processing, segmentation, post-segmentation and even post-recognition techniques for successfully breaking CAPTCHAs. This fact made us rethink the strength of using solely ML techniques in directly predicting the text contained in a given text-based CAPTCHA. Tang et al. (2018) suggested a CNN-based attack which however differs from our approach as they use two different CNNs, one for predicting the number of characters contained in a sample having the limitation of bounded length prediction (e.g., 5-7 characters prediction range), and one for recognizing the actual character contained in a single character frame after segmenting the CAPTCHA sample into *C* equally distributed segments (where *C* is the number of characters contained in a sample as estimated by the CNN). Moreover, they apply different pre-processing techniques for increasing the two CNNs' prediction success rates. Zi et al. (2020) have managed to achieve high attack success rates using an end-to-end ML-only (i.e., no pre-processing, no segmentation) methodology, which is however, a *hybrid* ML model consisting of a combination of CNNs and LSTMs. Also, it has already been reported that such hybrid models demonstrate excellent performance on the most difficult text-based CAPTCHA schemes (not tested in Zi et al., 2020) (Bursztein et al., 2014; Dileep et al., 2017; Ye et al., 2018).

In our case, we are concerned about the application of a single type of ML model (i.e., one type of ML algorithm) for directly attacking the *most difficult* text-based CAPTCHAs. By performing such an experiment we can evaluate the *robustness, generality* and *autonomy* of ML methodologies on solving hard AI problems, such as the automated text-based CAPTCHA breaking problem (Ahn et al., 2003). As a result, we have selected two ML algorithms that are the most popular choices based on all the articles studied: (a) CNNs (ANNs-based) and (b) SVMs. Our choice was also based on the fact that CNNs constrain their architecture's design specifically for image recognition and immense complexity problems (Dionysiou et al., 2018), whereas SVMs require minimum ML-related knowledge as well as minimum parameter tuning and can be easily trained on solving a wide-range of problems while also achieving high-enough accuracy rates. All the aforementioned reasons make CNNs and SVMs attractive to potential attackers.

5.1. Convolutional neural networks (CNNs)

CNNs are biologically-inspired variants of Multi-Layer Perceptrons (MLPs). CNNs is a class of deep, feedforward ANNs, inspired by the HVS (LeCun and Bengio, 1998), that have successfully been applied to analysing visual imagery (Krizhevsky et al., 2012; Rawat and Wang, 2017), and explicitly designed for complex feature extraction from two dimensional (2D) and three dimensional (3D) input volumes (Dionysiou et al., 2018). The CNN architecture consists of an input layer (inactive), multiple hidden layers and an output layer. The hidden layers of a CNN typically consist of convolutional layers, pooling layers and fully connected layers. CNNs take advantage of the fact that the input would be like an image so they constrain their architecture in a more sensible way. It is worth noting that every layer of a CNN transforms one volume of activations to another through a differential function. In this paper, we utilize CNNs in order to evaluate the robustness of the current state-of-the-art text-based CAPTCHA schemes (i.e., schemes that deploy effective security features) to single type ML-only attacks where the only task of a potential adversary is to deploy an *off-the-shelf* CNN architecture for directly predicting the characters contained in a specific text-based CAPTCHA scheme.

5.2. Support vector machines (SVMs)

On the other hand, SVM is an ML-based statistical model that tries to non-linearly map input vectors to a higher dimensional feature space using a specific kernel function so that a linearly inseparable problem in the input space becomes linearly separable in the new feature space (i.e., a linear decision surface can be constructed) (Vapnik, 1999). CNNs are considered to be a more powerful solution, in terms of complex feature extraction and classification capabilities, than SVMs. Nevertheless, some major advantages of SVMs are: (a) high generalization ability (Meyer and Wien, 2001), (b) robust performance with respect to sparse and noisy data (Haussler et al., 2000), and (c) small number of parameters to be tuned (in contrast to CNNs where parameter tuning is a painful task).

5.3. Motivation

Our main goal of directly applying CNNs and SVMs for CAPTCHA text recognition is to evaluate whether or not the other common phases used, in CAPTCHA breaking process, are essential as well as to measure the performance of single type ML models. Achieving high success rates using a single type of ML model essentially denotes serious security implications for online systems that still deploy text-based CAPTCHAs as their main user authentication mechanism. This is because the high attack success rates achieved in combination with the simplicity of the ML model used, and thus the *minimum* solving time needed for a given CAPTCHA sample, annihilate the security of current text-based CAPTCHA schemes.

5.4. Hyper-parameters optimization

As the tuning of the CNN's hyper-parameters is a very challenging task as well as considering our experiments' main purpose, we choose to deploy an *improved/enhanced* version of the well-known LeNet-5 architecture proposed by LeCun et al. in 1995, for handwritten and machine-printed character recognition (LeCun et al., 1995a). LeNet-5 is now considered obsolete and possibly outdated, however it is *still* the baseline of almost all state-of-the-art image recognition ML models, such as ImageNet (Krizhevsky et al., 2012), ZFNet (Zeiler and Fergus, 2014), and GoogLeNet (Szegedy et al., 2015). We therefore build on top of this architecture in order to create an enhanced version of the LeNet-5, *specifically designed* for text-based CAPTCHA breaking. Our improved version of the LeNet-5, shown in Fig. 2, takes as input the whole CAPTCHA image and directly recognises the text contained in it. The optimal CNN parameters that lead to the highest recognition accuracy results are shown in Fig. 2.

After performing several experiments using different kernels, misclassification penalty parameters (C) (Cortes and Vapnik, 1995) and Gamma values (G) (Cortes and Vapnik, 1995) we have also decided for the optimal SVM parameters that lead to the highest recognition accuracy results and which are: (a) Kernel = Radial Basis Function, (b) C = 1, and (c) G = 0.001.

5.5. Dataset

Following the methodology proposed in Bursztein et al. (2011b) we have created a corpus of real-world CAPTCHAs to evaluate the effectiveness of our ML models. We use the *CNN text-based CAPTCHA scheme*, shown in Fig. 1, for training our CNN and SVM classifiers. This text-based CAPTCHA scheme has also been utilized in Bursztein et al. (2011b), Bursztein et al. (2014a) and Ye et al. (2018). The motivation behind the selection of this CAPTCHA scheme for conducting

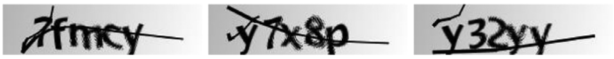

Fig. 1. In this figure, three samples of the **CNN** text-based CAPTCHA scheme are shown. Each sample, contains many security features, i.e., collapsing, occluding lines, waving, etc.

our experiments lies on the fact that it reports the lowest attack success rates in all three (3) aforementioned papers (16% in Bursztein et al., 2011b, 51.6% in Ye et al., 2018 and 51.09% in Bursztein et al., 2014a). Furthermore, this CAPTCHA scheme has been selected as it contains a large number of advanced security features (discussed in Section 6) namely: (a) negative kerning (collapsing), (b) occluding lines, (c) character overlapping, and (d) character rotating, distortion or waving, which seem to be among the most effective security features used in text-based CAPTCHA designing.

5.6. Evaluation

In order to conclude about the recognition success rates for the two selected classifiers we conduct numerous experiments using different sets of parameters as well as adapting the models' architectures. The highest recognition accuracy results obtained on CNN text-based CAPTCHA scheme for our CNN and SVM classifiers are *0.7%* and *0.1%* respectively. However, it has been experimentally shown that very high attack success rates can be achieved utilizing the other common phases, such as pre-processing, segmentation, etc., using either CNNs (Chellapilla et al., 2005; Chen et al., 2018; Gao et al., 2017; 2013; Stark et al., 2015; Tang et al., 2018; Ye et al., 2018) or SVMs (Bursztein et al., 2011b; Fiot and Paucher, 2009; Starostenko et al., 2015). The low recognition accuracy results achieved indicate that directly applying single type ML-based models, such as CNNs and SVMs, for recognizing the text contained in a text-based CAPTCHA scheme is likely not enough. This conclusion is further strengthened from the fact that CNNs seem to be the most powerful ML-based models when it comes to image recognition related problems (Krizhevsky et al., 2012; LeCun and Bengio, 1998; Rawat and Wang, 2017), whereas SVMs are among the most easy to be deployed ML models (i.e., SVMs are easy to be trained and tuned) successfully been applied to a wide-range of applications while also achieving high enough success rates when it comes to automated text-based CAPTCHA solving (Bursztein et al., 2011b). Moreover, showing that directly applying ML-models on text-based CAPTCHA schemes does not seem to work, is a significant contribution as the community cannot develop simple straight-forward techniques for directly solving text-based CAPTCHAs, and thus minimizing the required breaking time, which indicates that breaking text-based CAPTCHAs requires effective and efficient solutions.

The other common phases used, such as pre-processing, segmentation, and post-processing, are *essential* phases for successfully tackling the text-based CAPTCHA breaking problem. However, breaking methodologies, like Ye et al. (2018)'s, deploy ML algorithms and more specifically ANN-based techniques (i.e., GANs) for all phases used, providing in this way a complete all-ML-based solver. As many techniques that can effectively and efficiently solve the automated text-based CAPTCHA breaking problem exist, different doubts regarding the security of text-based CAPTCHAs arise. New design directions and principles for improving the security of text-based CAPTCHAs have to be taken. Also, a detailed security and usability assessment of text-based CAPTCHA schemes is considered critical to be performed by every site operator.

Nonetheless, most of the site operators have realised the limitations of text-based CAPTCHAs switching in this way their basic scheme to other CAPTCHA versions like image-based or game-based CAPTCHA schemes. Furthermore, most of the site operators, especially the popular ones, deploy heavier-weight secondary defences (e.g., SMS validation, account shutdowns, additional CAPTCHA screens, etc.) (Fleizach et al., 2007) as an extra security measure but only after a text-based CAPTCHA is passed and some usage threshold is triggered (e.g., multiple sign-ups from the same IP address) (Motoyama et al., 2010). Moving forward, inspired by Motoyama et al. (2010), CAPTCHAs have to be viewed, not only as a technological impediment to an attacker, but also as an economic one. However, Motoyama et al. (2010) mentioned this considering the cost of human-based labor CAPTCHA solving services, whereas in this article the cost (economic impediment) is due to the specialized and highly skilled staff needed for developing such automated text-based CAPTCHA solvers.

6. Design principles for improved CAPTCHA security

After an extensive study on a large number of scientific papers related to text-based CAPTCHA breaking and security (Ahmad et al., 2010, 2011, Bursztein et al., 2011, 2014; Chandavale et al., 2009; Chellapilla and Simard, 2004; Cruz-Perez et al., 2012; Dileep et al., 2017; Fiot and Paucher, 2009; Gao et al., 2012, 2013, 2014, 2017; Hindle et al., 2008; Huang et al., 2010; Hussain et al., 2016; Li et al., 2010; Mori and Malik, 2003; Moy et al., 2004; Nachar et al., 2015; Rui et al., 2013; Sano et al., 2015; Shih-Yu et al., 2008; Stark et al., 2015; Starostenko et al., 2015; Tang et al., 2016; Yan and Ahmad, 2007; Yan and El Ahmad, 2008; Ye et al., 2018; Zhang and Wang, 2010), several guidelines and suggestions on designing improved text-based CAPTCHA schemes to resist automated breaking resulted. Those text-based CAPTCHA designing suggestions will

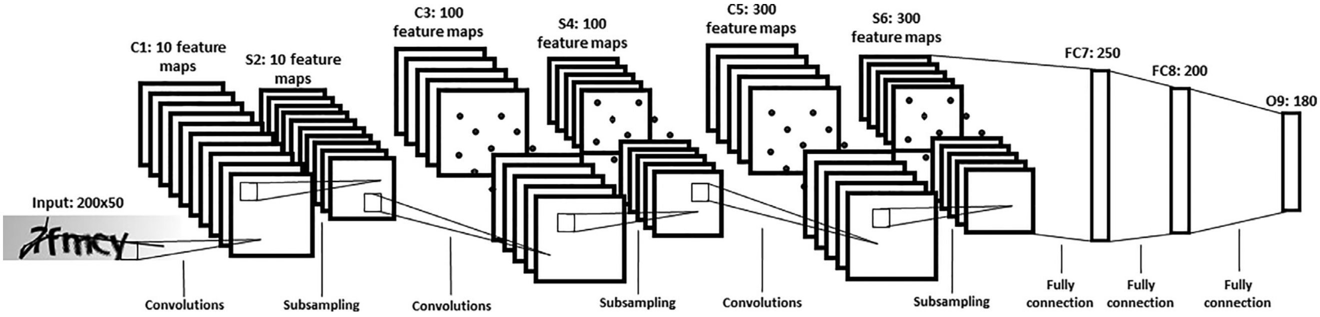

Fig. 2. The **CNN** architecture used for our experiments, which is an *improved/enhanced* version of the LeNet-5 architecture used for handwritten and machine-printed character recognition (LeCun et al., 1995a). As shown in the figure, it is composed by 9 layers, among which there are 3 convolutional layers (C1, C3 and C5), 3 sub-sampling (pooling) layers (S2, S4 and S6), and 2 fully connected layers (FC7 and FC8), that are followed by a softmax output layer (O9). Convolutional layers use 2 by 2 convolutions with stride 1 and ReLU as an activation function. Sub-sampling layers use max-pooling with 2 × 2 filters and stride 2.

be useful to CAPTCHA designers as well as CAPTCHA attackers. For example, randomizing the CAPTCHA length and individual relative character size makes text-based CAPTCHA recognition painless for humans, but at the same time a way too hard task for automated attacks (Bursztein et al., 2011b). Similarly, if the characters in a text-based CAPTCHA scheme are of the same size, then the segmentation and recognition techniques used are extremely benefited. While carefully chosen anti-recognition[4] techniques help slow down the learning process and reduce recognition technique's success rates, they are not sufficient by themselves. Based on our large-scale evaluation, we have concluded that for a text-based CAPTCHA scheme to reach the maximum level of security; *anti-segmentation* and *anti-recognition* techniques must be *combined*. Note that many techniques explained in this section serve both anti-recognition and anti-segmentation. Furthermore, different security features that are not considered to improve the security of text-based CAPTCHA schemes are also stated. However, in the last section of this research we argue why text-based CAPTCHA schemes must be discontinued as a technique for determining if a user is a human or not.

It has to be noted that *we do not* provide empirical evidence for the security of the following text-based CAPTCHA design security features as all the design guidelines provided in this paper are *extensively evaluated in the related literature* (Ahmad et al., 2010, 2011, Bursztein et al., 2011, 2014; Dileep et al., 2017; Gao et al., 2013) and thus their is *no need for further experimentation*. Furthermore, these techniques are considered to be the most generic ones as they can be easily applied to all text-based CAPTCHA schemes. However, we suggest site operators to carefully adopt each design security feature as the text-based CAPTCHA designing reflects a *trade-off* between *security* and *usability*. Moreover, further experiments should be conducted for validating the security of the improved text-based CAPTCHA scheme.

6.1. Confusion lines

Using confusion lines that cross upon multiple characters prevents segmentation-based techniques from effectively detect and segment characters. This leads the recognition technique used in a later stage to poor success rates. As mentioned in Bursztein et al. (2011b), two types of lines used to prevent segmentation are: (a) small lines that cross the CAPTCHA's characters, and (b) large lines of the same width as the characters' lines that cross the entire CAPTCHAs. We recommend using curved lines, instead of straight lines, with random angles and same width as the characters' width, crossing upon a random number of characters with random slopes. The length of these lines should be random and their colour should match the characters' colour. In case the CAPTCHA scheme uses different colours for the characters then random coloured lines should be used. This technique causes preprocessing and segmentation algorithms to fail on distinguishing the characters from the confusion lines.

6.2. Negative Kerning (Collapsing)

As mentioned in Bursztein et al. (2011b), Bursztein et al. (2014a), and Ye et al. (2018) collapsing (or negative kerning) anti-segmentation technique is used by the most prominent text-based CAPTCHA schemes as a general algorithm for breaking this type of security feature does not exist. Collapsing, uses negative space between characters to resist segmentation by ensuring that each character is occluded by its neighbours (Bursztein et al., 2014a). We recommend using collapsing as the main anti-segmentation security feature as it provides an efficient defence against segmentation. It is also advised though to deploy collapsing in combination with random CAPTCHA length, characters' size and font types in order to increase the effectiveness of this security feature (Bursztein et al., 2011b).

6.3. Random CAPTCHA length

Attackers can easily exploit fixed length CAPTCHA schemes as ML-based techniques able to break those schemes, with high success rates, exist. In other words, using fixed length CAPTCHA schemes gives too much information to the attacker while also limiting the training space of ML-based solvers. Instead, randomizing the CAPTCHA length has a major negative impact on automated text-based CAPTCHA solvers' success rates, requiring a significant higher overhead for training holistic/non-segmentation ML-based solvers, one per each CAPTCHA length. We highly recommend site operators on using this technique/security feature as it causes the larger portion of automated ML-based CAPTCHA solvers to fail.

6.4. Wave CAPTCHAs

Waving the different text-based CAPTCHA schemes increases the difficulty of finding cut points in case of collapsing and helps mitigate the risk of the attacker finding the added line based on its slope when using lines (Bursztein et al., 2011b). Furthermore, waving the text-based CAPTCHAs hardens the problem for the recognition technique been used.

6.5. Random characters' size and font types

Randomizing characters' size as well as using several fonts causes segmentation-based techniques to fail as a general highly effective known method for segmenting the characters contained in a text-based CAPTCHA scheme does not exist. In addition, using several fonts reduces the classifier's accuracy and scheme's learnability. We recommend using this technique for the same reasons described in Section 6.3 as it has a major negative impact on automated text-based CAPTCHA solvers also being easy to deploy.

6.6. Rotate CAPTCHA's characters

It is well-known that DL-based techniques, which are the state-of-the-art for image recognition, fail when rotating the input image. Rotating the CAPTCHA's characters in a considerably small number of degrees can effectively cause most automated ML-based CAPTCHA solvers to fail and still be relatively painless to be solved by humans. However, we suggest site operators to carefully use rotation as it may negatively affect the user experience.

6.7. Combination of security features

Many CAPTCHA schemes combine different security features for resisting automated solving. Combining multiple security features indeed improves the robustness of a text-based CAPTCHA scheme (Bursztein et al., 2011b; Ye et al., 2018). This can be easily seen from the success rates achieved when using different security features. For example, in Ye et al. (2018) a table (i.e., Table 5) showing how the combination of different security features affects the accuracy of the proposed GAN-based solver, exists. On the other hand, using more and more security features negatively affects the usability of text-based CAPTCHA schemes. Again, as mentioned in Ye et al. (2018), their annotators struggled to recognize a specific CAPTCHA scheme that combines overlapping, rotation, distortion, and waving security features.

[4] In this paper, the term *anti-recognition techniques* is used to describe the image/text manipulations that aim at preventing the recognition of individual characters or a sequence of characters.

6.8. Non-effective security features

Background confusion techniques can be inferred and filtered (for example using pre-processing techniques) so the use of this kind of distortion is not considered to be effective. These non-effective anti-segmentation techniques include: (a) complex background, (b) background and characters color similarity, and (c) random noise. As we have already explained in Section 6.2, collapsing can be an effective defence if combined with random CAPTCHA length, characters' sizes, and fonts. Nevertheless, it is recommended that site operators should carefully use collapsing as it has been shown experimentally that if the overlapping area is greater than 6 pixels, the resulting text-based CAPTCHA will significantly affect user experience because it will become painful for humans to discriminate the characters from the CAPTCHA's image representation (Bursztein et al., 2014a).

7. Future directions

In this paper, an extensive and systematic evaluation and classification of the related literature regarding automated text-based CAPTCHA solvers has been conducted, mainly focusing on ML-based solvers. After examining and evaluating a large number of related articles published on the last two decades, we have concluded that holistic (i.e., non segmentation-based) ML-based approaches, incorporating, or not, mechanisms inspired by systems neuroscience and HVS, can lead to robust and general applicable automated text-based CAPTCHA solvers that are highly effective and efficient in all terms. To sum up, we have initially classified each paper studied in one of the two main categories, i.e., ML-related and Non ML-related, based on whether or not the proposed methodology is mainly based on ML. After performing the classification process, we have tracked down whether or not, the proposed methodologies make use of specific phases and techniques in the text-based CAPTCHA breaking process and thus receiving as a result the classification Table 2. Observing the resulting table (i.e., Table 2), we have identified that the latest automated solvers use CNNs or other techniques and algorithms heavily inspired from HVS (e.g., RCN) while also adopting an holistic approach. Furthermore, we have tracked down the three state-of-the-art solvers that have successfully solved the automated text-based CAPTCHA breaking problem and examined the motivation and insights behind their approach.

As many automated solvers have managed to break the 1% barrier (Bursztein et al., 2011b) of many popular text-based CAPTCHAs (Bursztein et al., 2014; Dileep et al., 2017; Ye et al., 2018), the reassessment of the way that we perform the reverse Turing tests (i.e., CAPTCHA) is considered critical. Thus, there is the need for more, improved design security features, to be invented preserving the usability for legitimate users unaffected, otherwise as Bursztein et al. stated, *the end is nigh* for the text-based CAPTCHA schemes (Bursztein et al., 2014a). New ways and suggestions for conducting the reverse Turing tests, like the ones proposed in Al-Fannah (2017); Baecher et al. (2010); Bursztein et al. (2014b); Datta et al. (2005); Gossweiler et al. (2009); Hoque et al. (2006); K. et al. (2016); Mohamed et al. (2014); Osadchy et al. (2017), should be carefully examined in order to conclude about the security and usability levels that they offer. The same should be done for widely-deployed CAPTCHA schemes, like Google's reCAPTCHA (Sivakorn et al., 2016). Furthermore, the same systematic evaluation should be performed for all types of CAPTCHA versions shown in Table 1 to aid the scientific community as well as the industry on receiving insights about the usefulness of each approach.

We advise companies and site operators that still use text-based CAPTCHA schemes for which automated CAPTCHA solvers able to break them with high success rates exist, to tweak their CAPTCHA's design according to the design security features reported in Section 6, while also rethinking of their security policies used for blocking automated bot systems attacks. We further suggest site operators that use text-based CAPTCHAs as their only human authentication mechanism that, if an automated CAPTCHA solver exists that is able to break a specific text-based CAPTCHA scheme more than 50% of times then it is highly advised that the specific CAPTCHA scheme should be replaced as, mathematically speaking, it does not provide any security on a particular, protected by the text-based CAPTCHA, resource. This is because in the uniform probability distribution those CAPTCHA schemes will more likely to break than to resist automated attacks. Moreover, we advise site operators to deploy secondary, more aggressive defences for minimizing the impact of automated text-based CAPTCHA solvers. In contrast with many other previous suggestions stating that the immunity of the CAPTCHA design should lie in the segmentation and not in the recognition phase, we suggest that a proper, in terms of security and usability, text-based CAPTCHA scheme should be *both hard to be segmented and hard to be recognized* by automated solvers.

8. Conclusion

Text-based CAPTCHA schemes, even after several proposed security features, remain woefully problematic. In this work, a systematic evaluation and classification of the related literature regarding automated text-based CAPTCHA solvers has been performed. In so doing, we discover that these vulnerabilities affect the security of systems that deploy text-based CAPTCHAs as their main human or bot classification system. An enormous amount of popular text-based CAPTCHA schemes are considered broken, something which is a clear evidence that the reassessment of the way that we perform reverse Turing tests (i.e., CAPTCHA) is considered critical. Moving forward, an in-depth examination of automated ML-based CAPTCHA solvers is conducted, while also examining the motivation and the underlying methodology and architecture of the three state-of-the-art automated text-based CAPTCHA solvers. After demonstrating experimentally that an ML-only approach is not enough for breaking a modern text-based CAPTCHA scheme, the most effective text-based CAPTCHA design security features are summarized for site operators to tweak and improve the security of their CAPTCHA's design. Concluding our research, new, fundamentally different ways of conducting the reverse Turing tests should be invented as ML-based, and more specifically DL-based approaches, are becoming stronger and stronger.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank the anonymous reviewers for helping us to improve the final version of this paper. This work was supported by the European Union's Horizon 2020 research and innovation programme under grant agreements No. 786669 (ReAct), and No. 830929 (CyberSec4Europe), and by the RESTART programmes of the research, technological development and innovation of the Research Promotion Foundation, under grant agreement ENTERPRISES/0916/0063 (PERSONAS).

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cose.2020.101947.

References

Abhishek, L.K., 2017. Thinning approach in digital image processing. In: Special Issue - SACAIM, pp. 326–330. 10.21172

Ahmad, A.S.E., Yan, J., Marshall, L., 2010. The robustness of a new captcha. In: Proceedings of the Third European Workshop on System Security. ACM, pp. 36–41.

Ahmad, A.S.E., Yan, J., Tayara, M., 2011. The robustness of Google CAPTCHA's. Computing Science, Newcastle University.

Ahn, L., Blum, M., Hopper, N.J., Langford, J., 2003. Captcha: using hard ai problems for security. In: International Conference on the Theory and Applications of Cryptographic Techniques. Springer, pp. 294–311.

Al-Fannah, N.M., 2017. Making defeating captchas harder for bots. CoRR. arXiv:1704.02803.

Avyuct, P. b., Avyuct, 2019. Generative adversarial networks (GANs): A step to AI algorithms.

Baecher, P., Fischlin, M., Gordon, L., Langenberg, R., Ltzow, M., Schrder, D., 2010. Captchas: the good, the bad, and the ugly. Sicherheit 2010. Sicherheit, Schutz und Zuverlssigkeit 353–365.

Bock, K., Patel, D., Hughey, G., Levin, D., 2017. uncaptcha: a low-resource defeat of Recaptcha's audio challenge. 11th USENIX Workshop on Offensive Technologies (WOOT 17). USENIX Association, Vancouver, BC.

Bursztein, E., Aigrain, J., Moscicki, A., Mitchell, J.C., 2014. The end is nigh: generic solving of text-based captchas. 8th USENIX Workshop on Offensive Technologies (WOOT 14).

Bursztein, E., Beauxis, R., Paskov, H., Perito, D., Fabry, C., Mitchell, J., 2011. The failure of noise-based non-continuous audio captchas. In: 2011 IEEE Symposium on Security and Privacy, pp. 19–31. doi:10.1109/SP.2011.14. ID: 1

Bursztein, E., Bethard, S., 2009. Decaptcha breaking 75% of ebay audio captchas. 3rd USENIX Workshop On Offensive Technologies. Usenix.

Bursztein, E., Martin, M., Mitchell, J., 2011. Text-based captcha strengths and weaknesses. In: Proceedings of the 18th ACM Conference on Computer and Communications Security. ACM, pp. 125–138.

Bursztein, E., Moscicki, A., Fabry, C., Bethard, S., Mitchell, J.C., Jurafsky, D., 2014. Easy does it: more usable captchas. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, pp. 2637–2646. doi:10.1145/2556288.2557322.

Chandavale, A.A., Sapkal, A.M., Jalnekar, R.M., 2009. Algorithm to break visual captcha. In: 2009 Second International Conference on Emerging Trends in Engineering & Technology. IEEE, pp. 258–262.

Chellapilla, K., Larson, K., Simard, P.Y., Czerwinski, M., 2005. Computers beat humans at single character recognition in reading based human interaction proofs (HIPS).. CEAS.

Chellapilla, K., Simard, P.Y., 2004. Using machine learning to break visual human interaction proofs (hips). In: Proceedings of the 17th International Conference on Neural Information Processing Systems. MIT Press, Cambridge, MA, USA, pp. 265–272.

Chen, J., Luo, X., Hu, J., Ye, D., Gong, D., 2018. An attack on hollow captcha using accurate filling and nonredundant merging. IETE Techn. Rev. 35 (sup1), 106–118. doi:10.1080/02564602.2018.1520152.

Chew, M., Tygar, J.D., 2004. Image recognition captchas. In: Zhang, K., Zheng, Y. (Eds.), Information Security. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 268–279.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20 (3), 273–297. doi:10.1023/A:1022627411411.

Cruz-Perez, C., Starostenko, O., Uceda-Ponga, F., Alarcon-Aquino, V., Reyes-Cabrera, L., 2012. Breaking recaptchas with unpredictable collapse: heuristic character segmentation and recognition. In: Mexican Conference on Pattern Recognition. Springer, pp. 155–165.

Darnstdt, M., Meutzner, H., Kolossa, D., 2014. Reducing the cost of breaking audio captchas by active and semi-supervised learning. In: 2014 13th International Conference on Machine Learning and Applications, pp. 67–73. doi:10.1109/ICMLA.2014.16. ID: 1

Datta, R., Li, J., Wang, J., 2005. Imagination: a robust image-based captcha generation system. In: Proceedings of the 13th ACM International Conference on Multimedia, MM 2005,, pp. 331–334. doi:10.1145/1101149.1101218.

Dileep, G., Hawkins, J., 2009. Towards a mathematical theory of cortical micro-circuits. PLOS Comput. Biol. 5 (10), e1000532.

Dileep, G., Lehrach, W., Kansky, K., Lázaro-Gredilla, M., Laan, C., Marthi, B., Lou, X., Meng, Z., Liu, Y., Wang, H., Lavin, A., Phoenix, D.S., 2017. A generative vision model that trains with high data efficiency and breaks text-based captchas. Science 358 (6368). doi:10.1126/science.aag2612.

Dionysiou, A., Agathocleous, M., Christodoulou, C., Promponas, V., 2018. Convolutional neural networks in combination with support vector machines for complex sequential data classification. In: International Conference on Artificial Neural Networks. Springer, pp. 444–455.

Elson, J., Douceur, J.J., Howell, J., Saul, J., 2007. Asirra: a captcha that exploits interest-aligned manual image categorization. In: Proceedings of the 14th ACM Conference on Computer and Communications Security. ACM, New York, NY, USA, pp. 366–374. doi:10.1145/1315245.1315291.

Fiot, J.-B., Paucher, R., 2009. The Captchacker Project. Ecole Centrale, Paris.

Fleizach, C., Liljenstam, M., Johansson, P., Voelker, G.M., Mhes, A., 2007. Can you infect me now?: Malware propagation in mobile phone networks. In: WORM'07 - Proceedings of the 2007 ACM Workshop on Recurring Malcode doi:10.1145/1314389.1314402.

Gao, H., Tang, M., Liu, Y., Zhang, P., Liu, X., 2017. Research on the security of microsofts two-layer captcha. IEEE Trans. Inf. Forensics Secur. 12 (7), 1671–1685.

Gao, H., Wang, W., Fan, Y., 2012. Divide and conquer: an efficient attack on Yahoo! captcha. In: 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications. IEEE, pp. 9–16.

Gao, H., Wang, W., Fan, Y., Qi, J., Liu, X., 2014. The robustness of "connecting characters together" captchas.. J. Inf. Sci. Eng. 30 (2), 347–369.

Gao, H., Wang, W., Qi, J., Wang, X., Liu, X., Yan, J., 2013. The robustness of hollow captchas. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. ACM, pp. 1075–1086.

Golle, P., 2008. Machine learning attacks against the asirra captcha. In: Proceedings of the 15th ACM Conference on Computer and Communications Security. ACM, New York, NY, USA, pp. 535–542. doi:10.1145/1455770.1455838.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680.

Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., Shet, V., 2013. Multi-digit number recognition from street view imagery using deep convolutional neural networks. arXiv:1312.6082.

Google, 2019. Invisible recaptcha.

Gossweiler, R., Kamvar, M., Baluja, S., 2009. What's up captcha?: A captcha based on image orientation.. In: WWW, pp. 841–850. doi:10.1145/1526709.1526822.

Goswami, G., Powell, B. M., Vatsa, M., Singh, R., Noore, A., 2014. Facedcaptcha: face detection based color image captcha. ID: 271521. 10.1016/j.future.2012.08.013.

Haussler, D., Bednarski, D.W., Schummer, M., Cristianini, N., Duffy, N., Furey, T.S., 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16 (10), 906–914. doi:10.1093/bioinformatics/16.10.906. 2019

Hernández-Castro, C.J., Ribagorda, A., Sáez, Y., 2009. Side-channel attack on labeling captchas. CoRR. arXiv:0908.1185.

Hindle, A., Godfrey, M.W., Holt, R.C., 2008. Reverse engineering captchas. In: 2008 15th Working Conference on Reverse Engineering. IEEE, pp. 59–68.

Hoque, M.E., Russomanno, D.J., Yeasin, M., 2006. 2d captchas from 3d models. In: Proceedings of the IEEE SoutheastCon 2006, pp. 165–170. doi:10.1109/second.2006.1629343. ID: 1

Huang, S.-Y., Lee, Y.-K., Bell, G., he Ou, Z., 2010. An efficient segmentation algorithm for captchas with line cluttering and character warping. Multimed. Tools Appl. 48 (2), 267–289.

Hussain, R., Gao, H., Shaikh, R.A., Soomro, S.P., 2016. Recognition based segmentation of connected characters in text based captchas. In: 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN). IEEE, pp. 673–676.

Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134.

K., K., S., P., A., S.M., 2016. Better the devil you know: a user study of two captchas and a possible replacement technology. NDSS Workshop on Usable Security (USEC)..

Kaelbling, L.P., Littman, M.L., Moore, A.W., 1996. Reinforcement learning: a survey. J. Artific. Intell. Res. 4, 237–285.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. Curran Associates Inc, USA, pp. 1097–1105.

Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B., 2015. Human-level concept learning through probabilistic program induction. Science 350 (6266), 1332–1338. doi:10.1126/science.aab3050. Http://science.sciencemag.org/content/350/6266/1332.full.pdf

Lam, L., Lee, S.-W., Suen, C.Y., 1992. Thinning methodologies-a comprehensive survey. IEEE Trans. Pattern Anal. Mach.Intell. 14 (9), 869–885.

LeCun, Y., Bengio, Y., 1998. Convolutional networks for images, speech, and time series. MIT Press, Cambridge, MA, USA, pp. 255–258.

LeCun, Y., Jackel, L., Bottou, L., Brunot, A., Cortes, C., Denker, J., Drucker, H., Guyon, I., Muller, U., Sackinger, E., et al., 1995. Comparison of learning algorithms for handwritten digit recognition. In: International Conference on Artificial Neural Networks, 60. Perth, Australia, pp. 53–60.

LeCun, Y., Jackel, L., Bottou, L., Cortes, C., Denker, J.S., Drucker, H., Guyon, I., Muller, U.A., Sackinger, E., Simard, P., 1995. Learning algorithms for classification: a comparison on handwritten digit recognition. Neural Netw. 261, 276.

Li, S., Shah, S., Khan, M., Khayam, S.A., Sadeghi, A.-R., Schmitz, R., 2010. Breaking e-banking captchas. In: Proceedings of the 26th Annual Computer Security Applications Conference. ACM, pp. 171–180.

Litvak, S., Ullman, S., 2009. Cortical circuitry implementing graphical models. Neural Comput. 21 (11), 3010–3056. doi:10.1162/neco.2009.05-08-783; 09.

Lutton, E., Maitre, H., Lopez-Krahe, J., 1994. Contribution to the determination of vanishing points using hough transform. IEEE Trans. Pattern Anal. Mach.Intell. 16 (4), 430–438. doi:10.1109/34.277598.

Meyer, D., Wien, T. U., 2001. Support Vector Machines. The Interface to LIBSVM in Package e1071. Online-Documentation of the Package e1071 for r.

Mohamed, M., Sachdeva, N., Georgescu, M., Gao, S., Saxena, N., Zhang, C., Kumaraguru, P., van Oorschot, P.C., Chen, W.-B., 2014. A three-way investigation of a game-captcha: automated attacks, relay attacks and usability. In: Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security. ACM, New York, NY, USA, pp. 195–206. doi:10.1145/2590296.2590298.

Mori, G., Malik, J., 2003. Recognizing objects in adversarial clutter: breaking a visual captcha. In: InProceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1 doi:10.1109/CVPR.2003.1211347. I–I

Motoyama, M., Levchenko, K., Kanich, C., McCoy, D., Voelker, G.M., Savage, S., 2010. Re: Captchas: Understanding captcha-solving services in an economic context. In: Proceedings of the 19th USENIX Conference on Security. USENIX Association, Berkeley, CA, USA. 28–28

Moy, G., Jones, N., Harkless, C., Potter, R., 2004. Distortion estimation techniques in solving visual captchas. In: In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2 doi:10.1109/CVPR.2004.1315140. II–II

Nachar, R.A., Inaty, E., Bonnin, P.J., Alayli, Y., 2015. Breaking down captcha using edge corners and fuzzy logic segmentation/recognition technique. Secur. Commun. Netw. 8 (18), 3995–4012.

Osadchy, M., Hernandez-Castro, J., Gibson, S., Dunkelman, O., Prez-Cabo, D., 2017. No bot expects the deepcaptcha! introducing immutable adversarial examples, with applications to captcha generation. IEEE Trans. Inf. Forensics Secur. 12 (11), 2640–2653. doi:10.1109/TIFS.2017.2718479.

Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22 (10), 1345–1359.

Plamondon, R., Srihari, S.N., 2000. Online and off-line handwriting recognition: a comprehensive survey. IEEE Transactions on Pattern Analysis and MachineIntelligence 22 (1), 63–84. doi:10.1109/34.824821. ID: 1

Rawat, W., Wang, Z., 2017. Deep convolutional neural networks for image classification: a comprehensive review. Neural Comput. 29 (9), 2352–2449. doi:10.1162/neco_a_00990.

Rui, C., Jing, Y., Rong-gui, H., Shu-guang, H., 2013. A novel LSTM-RNN decoding algorithm in captcha recognition. In: 2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control. IEEE, pp. 766–771.

Sano, S., Otsuka, T., Itoyama, K., Okuno, H.G., 2015. Hmm-based attacks on Google's Recaptcha with continuous visual and audio symbols. J. Inf. Process. 23 (6), 814–826.

Shih-Yu, H., Yeuan-Kuen, L., Bell, G., Ou, Z.-H., 2008. A projection-based segmentation algorithm for breaking MSN and Yahoo captchas 2170.

Shireesha, M., Gaikwad, V., 2013. Performance evaluation of captcha word ranking algorithm to break video captcha. Int. J. Comput. Appl. 75, 30–35. doi:10.5120/13149-0573.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

Sivakorn, S., Polakis, I., Keromytis, A.D., 2016. I am robot: (deep) learning to break semantic image captchas. In: 2016 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 388–403. doi:10.1109/EuroSP.2016.37. ID: 1

Solanki, S., Krishnan, G., Sampath, V., Polakis, J., 2017. In (cyber)space bots can hear you speak: breaking audio captchas using OTS speech recognition. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. ACM, New York, NY, USA, pp. 69–80. doi:10.1145/3128572.3140443.

Soupionis, Y., Tountas, G., Gritzalis, D., 2009. Audio captcha for sip-based voip. In: Gritzalis, D., Lopez, J. (Eds.), Emerging Challenges for Security. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 25–38. ID: 10.1007/978-3-642-01244-0_3

Srinivas, S., Sarvadevabhatla, R.K., Mopuri, K.R., Prabhu, N., Kruthiventi, S.S., Babu, R.V., 2016. A taxonomy of deep convolutional neural nets for computer vision. Front. Robot. AI 2, 36.

Stark, F., Hazrbas, C., Triebel, R., Cremers, D., 2015. Captcha recognition with active deep learning. GCPR Workshop on New Challenges in Neural Computation, 10.

Starostenko, O., Cruz-Perez, C., Uceda-Ponga, F., Alarcon-Aquino, V., 2015. Breaking text-based captchas with variable word and character orientation. Pattern Recognit. 48 (4), 1101–1112.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9.

Tam, J., Hyde, S., Simsa, J., Ahn, L.V., 2008. Breaking audio captchas. In: Proceedings of the 21st International Conference on Neural Information Processing Systems. Curran Associates Inc., USA, pp. 1625–1632.

Tam, J., Simsa, J., Huggins-Daines, D., Von Ahn, L., Blum, M., 2008. Improving audio captchas. Symposium on Usable Privacy and Security (SOUPS).

Tang, M., Gao, H., Zhang, Y., Liu, Y., Zhang, P., Wang, P., 2018. Research on deep learning techniques in breaking text-based captchas and designing image-based captcha. IEEE Trans. Inf. Forensics Secur. 13 (10), 2522–2537.

Tang, M., Zhang, P., Zhou, X., Wang, X., Li, J., 2016. A simple generic attack on text captchas. In: Proceedings of the 2016 Network and Distributed System Security Symposium (NDSS).

Thorpe, S., Fize, D., Marlot, C., 1996. Speed of processing in the human visual system. Nature 381. doi:10.1038/381520a0. 520–2

Turing, A.M., 2009. Computing machinery and intelligence. In: Parsing the Turing Test. Springer, pp. 23–65.

Vapnik, V.N., 1999. An overview of statistical learning theory. IEEE Transactions on Neural Networks 10 (5), 988–999. doi:10.1109/72.788640. ID: 1

Xu, Y., Reynaga, G., Chiasson, S., Frahm, J., Monrose, F., van Oorschot, P.C., 2014. Security analysis and related usability of motion-based captchas: Decoding codewords in motion. IEEE Transactions on Dependable and Secure Computing 11 (5), 480–493. doi:10.1109/TDSC.2013.52. ID: 1

Yan, J., Ahmad, A.S.E., 2007. Breaking visual captchas with naive pattern recognition algorithms. In: Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007). IEEE, pp. 279–291.

Yan, J., El Ahmad, A.S., 2008. A low-cost attack on a microsoft captcha. In: Proceedings of the 15th ACM Conference on Computer and Communications Security. ACM, New York, NY, USA, pp. 543–554. doi:10.1145/1455770.1455839.

Ye, G., Tang, Z., Fang, D., Zhu, Z., Feng, Y., Xu, P., Chen, X., Wang, Z., 2018. Yet another text captcha solver: a generative adversarial network based approach. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. ACM, pp. 332–348.

Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks. In: European Conference on Computer Vision. Springer, pp. 818–833.

Zhang, J., Wang, X., 2010. Breaking internet banking captcha based on instance learning. In: 2010 International Symposium on Computational Intelligence and Design. IEEE, pp. 39–43.

Zhu, B.B., Yan, J., Li, Q., Yang, C., Liu, J., Xu, N., Yi, M., Cai, K., 2010. Attacks and design of image recognition captchas. In: Proceedings of the 17th ACM Conference on Computer and Communications Security. ACM, New York, NY, USA, pp. 187–200. doi:10.1145/1866307.1866329.

Zi, Y., Gao, H., Cheng, Z., Liu, Y., 2020. An end-to-end attack on text captchas. IEEE Trans. Inf. Forensics Secur. 15, 753–766. doi:10.1109/TIFS.2019.2928622.

Antreas Dionysiou received his BSc degree in computer science and his MSc degree in Computer Science with specialization in Intelligent Systems from the University of Cyprus (UCY) in 2018 and 2019 respectively, both with distinction. He is currently pursuing his PhD in computer science at UCY. In addition, he is working as a special scientist at the security research group (SREC) also being a group member of Computational Intelligence and Neuroscience research group (CIN), both at UCY. His current research interests include information/cyber security, computational intelligence, machine learning, neuroscience, and bioinformatics.

Elias Athanasopoulos is currently an assistant professor at University of Cyprus. He received his BSc in Physics from the University of Athens and his Ph.D. in Computer Science from the University of Crete. Before joining University of Cyprus, he was an assistant professor with Vrije Universiteit Amsterdam. His research interests are systems security and privacy. Elias is a Microsoft Research PhD Scholar. He has interned with Microsoft Research in Cambridge and he has worked as a research assistant with FORTH (Greece). Elias is also a Marie Curie fellow with Columbia University and FORTH. He has several publications in all top security conferences like IEEE Security and Privacy, ACM CCS, Usenix Security, and NDSS.