A Generative Adversarial Learning Framework for Breaking Text-Based CAPTCHA in the Dark Web

Ning Zhang
Department of Management Information Systems
University of Arizona
Tucson, USA
zhangning@email.arizona.edu

Mohammadreza Ebrahimi
Department of Management Information Systems
University of Arizona
Tucson, USA
ebrahimi@email.arizona.edu

Weifeng Li
Department of Management Information Systems
University of Georgia
Athens, USA
weifeng.li@uga.edu

Hsinchun Chen
Department of Management Information Systems
University of Arizona
Tucson, USA
hsinchun@arizona.edu

***Abstract*—Cyber threat intelligence (CTI) necessitates automated monitoring of dark web platforms (e.g., Dark Net Markets and carding shops) on a large scale. While there are existing methods for collecting data from the surface web, large-scale dark web data collection is commonly hindered by anti-crawling measures. Text-based CAPTCHA serves as the most prohibitive type of these measures. Text-based CAPTCHA requires the user to recognize a combination of hard-to-read characters. Dark web CAPTCHA patterns are intentionally designed to have additional background noise and variable character length to prevent automated CAPTCHA breaking. Existing CAPTCHA breaking methods cannot remedy these challenges and are therefore not applicable to the dark web. In this study, we propose a novel framework for breaking text-based CAPTCHA in the dark web. The proposed framework utilizes Generative Adversarial Network (GAN) to counteract dark web-specific background noise and leverages an enhanced character segmentation algorithm. Our proposed method was evaluated on both benchmark and dark web CAPTCHA testbeds. The proposed method significantly outperformed the state-of-the-art baseline methods on all datasets, achieving over 92.08% success rate on dark web testbeds. Our research enables the CTI community to develop advanced capabilities of large-scale dark web monitoring.**

***Index Terms*—automated CAPTCHA breaking, dark web, generative adversarial networks, cyber threat intelligence**

I. Introduction

The dark web [1] hosts a plethora of illicit platforms such as Dark Net Markets (DNMs) and carding shops, where hackers exchange malware, hacking tools, and stolen financial information. Recognizing the enormous value of the dark web data, cybersecurity companies (e.g., FireEye) develop Cyber Threat Intelligence (CTI) from the dark web to identify cyber threats [2]. For example, financial institutions can leverage CTI to learn about stolen financial information sold in carding shops and DNMs to prepare their customers for potential risks [3]. As such, CTI necessitates automated monitoring of the dark web on a large scale [4] [5].

While automated web crawling is an integral part of the dark web CTI monitoring, the dark web extensively employs anti-crawling measures to prevent automated data collection. One major anti-crawling measure is text-based CAPTCHA (Completely Automated Public Turing Test to tell Computers and Human Apart) [6]. Text-based CAPTCHA determines whether the user is a web crawler by presenting a heavily obfuscated image of characters and examining the user's ability to identify the characters shown in the CAPTCHA image. The obfuscation generally includes the change of characters, such as font, color, and rotation. When navigating through dark web platforms, automated crawlers are frequently disrupted by text-based CAPTCHA challenges, which require human involvement and therefore hamper large-scale dark web collection. Furthermore, dark web platforms increase the difficulty of algorithmic CAPTCHA breaking by adding noisy background consisting of colorful curves and dots and varying the number of characters in the CAPTCHA image.

Machine Learning (ML) methods have been developed and demonstrated promising results in automated CAPTCHA breaking; nonetheless, text-based CAPTCHA patterns in the dark web present two nontrivial technical challenges [7]. First, CAPTCHA backgrounds are rendered particularly noisy with random curves and dots. Such noisy backgrounds are challenging for existing ML-based methods because they need to learn and distinguish a variety of randomly generated background patterns, in addition to recognizing the characters. Second, while the character length of text-based CAPTCHA patterns is a key parameter for training ML-based CAPTCHA breaking methods, dark web platforms rarely use CAPTCHA patterns with the same character length, causing most ML-based CAPTCHA breaking methods ineffective. This is mainly

Acknowledgments: This material is based upon work supported by the National Science Foundation (NSF) under Secure and Trustworthy Cyberspace (grant No. 1936370), Cybersecurity Innovation for Cyberinfrastructure (grant No. 1917117), and Cybersecurity Scholarship-for-Service (grant No. 1921485).

because pre-trained ML models have difficulty in breaking CAPTCHA patterns with different character lengths from what they have been trained on. In addition to these challenges, there is a lack of dark web-specific text-based CAPTCHA datasets that have been labeled for training ML-based CAPTCHA breaking methods. As such, existing CAPTCHA breaking methods cannot effectively facilitate the large-scale dark web CTI monitoring.

Motivated by these challenges, we propose a novel framework that leverages the state-of-the-art deep learning techniques for breaking text-based CAPTCHA in the dark web. The proposed framework comprises three major components:

- **Automated background removal** seeks to remove the dark web-specific background noise to improve CAPTCHA breaking performance. To this end, we propose a Generative Adversarial Network (GAN) model, CAPTCHA GAN, that learns to generate CAPTCHA patterns with relatively clean background from the original patterns with noisy backgrounds. Instead of using millions of labeled CAPTCHA images for training, our CAPTCHA GAN leverages adversarial learning to generate training data.
- **Character segmentation** addresses the challenge of variable character length by extracting image segments from the CAPTCHA image, each of which contains one single character. We extend the state-of-the-art image segmentation technique for dark web-specific CAPTCHA.
- **Character recognition** detects the character from each CAPTCHA image segment. We utilize the state-of-the-art Convolutional Neural Network (CNN) to recognize characters from CAPTCHA image segments.

The remainder of this paper is organized as follows. First, we review CAPTCHA and text-based CAPTCHA, background denoising, character segmentation, and character recognition techniques. Subsequently, we detail the components of our proposed method. Lastly, we compare the success rate of our proposed method to the state-of-the art automated text-based CAPTCHA breaking methods and highlight promising future directions.

II. Literature Review

A. CAPTCHA

CAPTCHA is a type of Turing test designed for distinguishing bots and human beings [8]. In general, there are four major types of CAPTCHA: Text-based, image-based, video-based, and audio-based. Text-based CAPTCHA requires users to correctly identify alphanumeric characters shown in a deliberately obfuscated pattern. Image-based CAPTCHA asks users to perform some actions on the images provided (e.g., move them to a specified location). Video-based CAPTCHA provides a video file and requires users to choose an option that best describes the video. Audio-based CAPTCHA plays an audio and asks the user to enter the characters mentioned in the audio. Text-based CAPTCHA has been the most commonly used type of CAPTCHA in the dark web [9].

Automated breaking of text-based CAPTCHA is nontrivial for two reasons:

- Security measures: Intentionally added complex patterns to the CAPTCHA image makes CAPTCHA breaking difficult [10].
- Variable character length: Automated solvers that are trained to break CAPTCHA with a specific length are often not generalizable to CAPTCHAs with more characters [7].

In particular, security measures can be categorized into two groups: foreground security measures and background security measures [11]. Foreground security measures are mainly deployed to distort characters using font change (varying the typeface), rotation (altering the orientation), and color change (diversifying the foreground color of each character). On the other hand, background security measures introduce additional obfuscation to the background of the CAPTCHA pattern, including dot noise (applying extraneous pixels), curve noise (adding irregular curvatures intersecting characters), and noise color change (varying the color of noise). Figure 1 shows two examples of dark web CAPTCHA with intense background and foreground security measures.

Fig. 1. Dark Web CAPTCHA Samples: (a) CAPTCHA pattern with dot noise and curve noise; (b) CAPTCHA pattern with curve noise and color change

B. Text-based CAPTCHA Breaking Methods

Extant CAPTCHA breaking methods generally employ three stages to address the aforementioned challenges: image preprocessing, character segmentation, and character recognition [9]. First, image processing performs a series of computer vision preprocessing techniques to remove background and foreground noise. Next, character segmentation splits CAPTCHA images to enable character-level CAPTCHA breaking. Lastly, character recognition applies character recognition algorithms to identify foreground characters. We summarize selected significant prior work based on the methods they employ at each stage (Table I). In the ensuing subsections, we examine image processing, character segmentation, and character recognition techniques in detail.

C. Image Preprocessing and Background Denoising

Preprocessing CAPTCHA patterns involves applying computer vision techniques to enhance the foreground and denoise the background. Three image preprocessing techniques have been commonly utilized in past research: pixel normalization, gray-scale conversion, and Gaussian smoothing. Pixel normalization scales a fragment of pixel value into a certain range to enhance inconspicuous features [16]. Gray-scale conversion transforms colored images into gray scale to eliminate the impact of different colors [9]. Gaussian smoothing applies Gaussian function to remove details with the goal of lowering the impact of curves and dots on characters' edges [9]. While

TABLE I
SELECTED MAJOR ML-BASED CAPTCHA BREAKING STUDIES

Year	Author	Focus	Testbed	Targeted Security Measure		Image Processing Techniques	Segmentation	Character Recognition
				Foreground	Background			
2020	Nouri and Rezaei [12]	Utilize CNN to break synthesized CAPTCHA for vulnerability study	500,000 randomly synthesized CAPTCHA	Font, rotation, color	Noise, curve , color	Grayscale Conversion, Gaussian Smoothing	No	CNN
2019	Wu et al. [9]	CAPTCHA breaking for numerous Chinese characters	Chinese National Enterprise Credit Information Publicity System (10k)	Font, rotation, color	Noise,color	Grayscale Conversion, Gaussian Smoothing	Yes	CNN
2018	Ye et al. [7]	Use generative adversarial network to remove security features	33 Captcha sets (MS, eBay, Sina ,etc.)	Font, rotation, color	Noise, curve , color	GAN-Based Background Removing	No	CNN
2018	Tang et al. [13]	Leverage segmentation and deep learning to break English and Chinese CAPTCHA.	11 Captcha sets	Font, rotation, color	Color	Grayscale Conversion,Gaussian Smoothing	Yes	CNN
2017	Le et al. [14]	Utilize synthetic data to train a deep learning model to break CAPTCHA	7 platforms (Baidu, eBay, Yahoo etc.)	Font, rotation	-	-	No	CNN and RNN
2016	Hussain et al. [15]	Leverage segmentation and deep learning to break the CAPTCHA with overlapping characters	Taobao, eBay, and MSN (3,500 CAPTCHA)	Font, rotation	-	Normalization, Grayscale Conversion	Yes	ANN

Note: ANN: Artificial Neural Network; CNN : Convolutional Neural Network; GAN: Generative Adversarial Network; CFS: Color Filling Segmentation; RNN: Recurrent Neural Network.

these techniques help enhance foreground characters, denoising background in complex patterns remains a major challenge [9]. Background denoising is difficult for dark web-specific CAPTCHA with complex noisy backgrounds because removing curve noise in the background cannot be addressed by common image preprocessing methods. As curve patterns are intended to be mistaken as foreground characters, removing the curve noise may result in the removal of actual foreground characters. Moreover, while machine learning models need to be trained on a large number of unseen background patterns, there lacks enough training data for such models. Recently, Ye et al. (2018) have shown that Generative Adversarial Network (GAN) has the potential to address this issue by automatically generating background patterns with eliminated curve noise [7].

D. Character Segmentation for Variable Length

In addition to the noisy background, variable character length presents another unique challenge of breaking dark web-specific CAPTCHA for two reasons. First, image-level methods that are trained on a pre-specified length are not applicable to CAPTCHAs with variable length of characters. Second, for these methods, the number of class labels grows exponentially with the number of characters (e.g., 10^3 for 3-digit numerical CAPTCHA and 10^4 for 4-digit numerical CAPTCHA). Consequently, a large number of CAPTCHA images is required for model training [7]. In contrast, character-level CAPTCHA solvers use a smaller number of class labels (e.g., 10 different classes (i.e., 0,. . . , 9) for numerical CAPTCHA) because character segmentation is utilized to separate the CAPTCHA pattern into single characters prior to character recognition [9]. Four segmentation methods are commonly adopted in CAPTCHA breaking research: Color Filling Segmentation (CFS), interval-based segmentation, pixel distribution-based segmentation, and contouring. CFS fills hollow shapes with a different color to differentiate characters [17]. Interval-based segmentation splits CAPTCHA images by fixed intervals [13]. Pixel distribution-based segmentation crops the characters based on the variance of the pixel values [15]. Contour detection identifies an area containing one single character based on the contour features of the character [18]. Contour detection is more suitable for breaking the dark web-specific CAPTCHA since it can deal with changes in font, color, and rotation of the characters.

E. Character Recognition with CNN

After identifying the character boundaries via segmentation, the next stage involves correctly recognizing the characters within each boundary [9]. As shown in Table 1, Convolutional Neural Networks (CNNs) have been widely used for the character recognition task [6] [10]. CNNs have shown promising results in counteracting foreground security measures such as rotation, color change, and font change of the characters. Specifically, CNN is composed of three main components: convolution layer, sampling layer, and fully connected layer. Each CNN component contributes to a certain aspect of CAPTCHA Character recognition. Convolution layer extracts geometrical salient features from local regions of the input image. In CAPTCHA breaking, this layer can extract key features from characters despite rotation. Sampling layer combines features from local regions to achieve higher-level features. Specifically for CAPTCHA breaking, this layer helps identify character features despite differences in their font and sizes. Fully connected (FC) layer weights the extracted features and assigns a probability to the output. This layer predicts the CAPTCHA pattern's class label based on the

extracted features. Given these reasons, we expect that CNN can effectively contribute to the dark web-specific CAPTCHA character recognition after proper segmentation.

III. Research Gaps and Questions

Two major research gaps are identified from reviewing prior studies. First, few studies offer methods for breaking CAPTCHA with complex noisy backgrounds. Second, existing methods cannot address CAPTCHAs with variable character lengths and noisy backgrounds under a unified framework, and thus are not directly applicable to dark web-specific CAPTCHA. Based on these gaps, we propose the following research question:

- *How can we design an automated CAPTCHA breaking framework to address the noisy background and variable character length in the dark web?*

IV. Research Design

We propose Dark Web-GAN (DW-GAN), a GAN-based method that utilizes background denoising, character segmentation, and character recognition to automatically break dark web CAPTCHA. Our design aims to counteract background security measures and address the problem of variable length for dark web CAPTCHA. Consistent with the analytical stages in prior literature, DW-GAN comprises three major components (Figure 2).

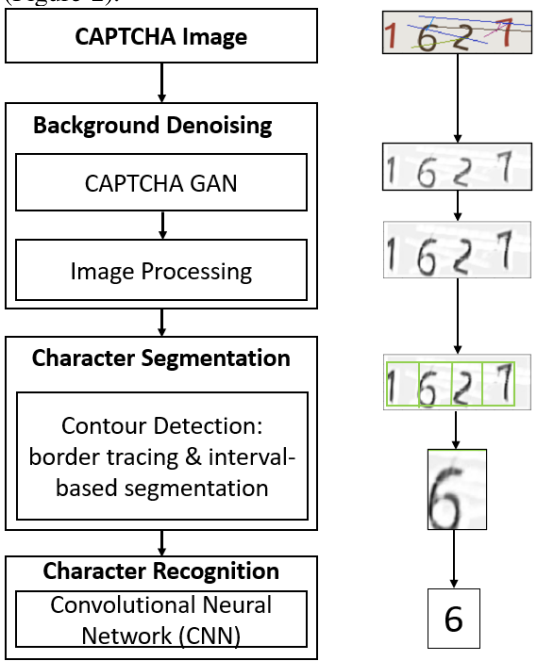

Fig. 2. DW-GAN Framework for Breaking Dark Web-specific CAPTCHA Images (left) and Corresponding Illustration (right)

First, the GAN-based background denoising component removes the noisy background from a given dark web CAPTCHA pattern. Then, the denoised CAPTCHA is further processed using Gaussian smoothing to remove the residual noise. Next, characters are segmented using an enhanced contour detection algorithm. We enhance traditional border tracing with a subsequent region enlargement procedure to get more accurate segmented regions. Lastly, segments of the original CAPTCHA pattern are fed to a CNN for character recognition. Next, we elaborate the details of each component.

A. Background Denoising: CAPTCHA GAN

Our GAN model aims to reduce various background curve noises from text-based CAPTCHA images without human intervention. Specifically, the generative nature of GAN allows us to train the DW-GAN with only a small labeled dataset (e.g., 500 dark web CAPTCHA patterns) and achieve a high performance in background denoising. GAN for background removal comprises two competing neural networks: generator and discriminator (Figure 3). The generator aims to generate CAPTCHA patterns with denoised background from original CAPTCHAs whereas the discriminator seeks to determine whether the generator has completely removed the background. The learning process for background removal in DW-GAN includes iterative execution of four major steps :

- *Step 1:* The generator creates a pattern g with denoised background from the original CAPTCHA pattern.
- *Step 2:* The generated pattern g and the corresponding original CAPTCHA pattern are fed to discriminator to assess whether the background noise has been completely removed.
- *Step 3:* The generator improves by learning from the loss function $\mathcal{L}_{\mathcal{G}}$ that compares the discriminator's output and the correct generated CAPTCHA pattern g.
- *Step 4:* The discriminator improves by learning from the loss function $\mathcal{L}_{\mathcal{D}}$ that compares the true label and the discriminators' output.

Steps 1-4 repeat until the generator and discriminator cannot improve further, known as the equilibrium condition [19].

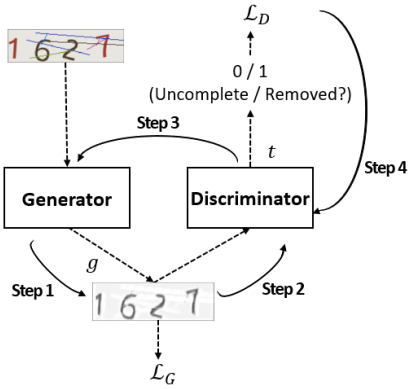

Fig. 3. Abstract View of GAN-based Background Denoiser

B. Background Denoising: Image Processing

While CAPTCHA GAN removes curve noise, some residual noise may still remain in the background. The residual noise could potentially interfere with the subsequent segmentation and character recognition. Therefore, we utilize a combination of three image processing techniques to further remove the remaining background noise. First, gray-scale conversion reduces the variance of color in background and foreground by converting the image to gray scale. Second, Gaussian

smoothing decreases the visibility of the residual background noise by blurring it. Third, pixel normalization is applied to distinguish the foreground from the background color.

C. Character Segmentation

Character segmentation focuses on identifying the boundary of characters. The border tracing algorithm is suited for segmenting the dark web CAPTCHA because characters in text-based CAPTCHA patterns often have distinguished borders. Border tracing can effectively identify the boundary pixels of the character region and separate the characters from the background [20]. Border tracing algorithm has two main steps [21]. First, the image is converted to binary color and scanned from the upper left to the bottom right. Then, for each pixel, the algorithm searches a square neighborhood (e.g. 3x3 pixels) to find the direction of the edges and define minimal regions to bound the character.

Despite the effectiveness of border tracing, the minimal detected regions might not be large enough to encompass the entire character (e.g., digit '6' in Figure 4(b)). Incorrect segmentation affects the subsequent recognition of characters from the segmented CAPTCHA pattern. To address this issue, we enhance the border tracing with region enlargement to encompass the entire character via a two-step process. In the first step, the initial character regions are detected with border tracing algorithm (Figure 4(b)). In the second step, the maximal regions that bounds the character are achieved by dividing the CAPTCHA pattern into fixed intervals. The previously detected regions are then overlapped with these maximal regions to attain the resultant segmentation (Figure 4(c)).

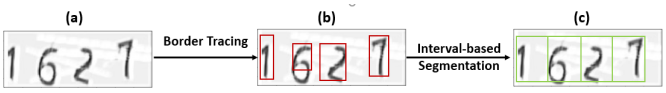

Fig. 4. Our Enhanced Border Tracing with Region Enlargement

D. Character Recognition

Consistent with prior literature [6] [10], we leverage CNN to detect characters in extracted CAPTCHA segments. Our character recognition CNN stacks convolutional layers and sampling layers to detect characters. The convolutional layer extracts features from local regions of the segmented CAPTCHA image and the sampling layer combines extracted features across multiple local regions to identify fine-grained features (e.g., lines and edges). Another convolutional-sampling structure of such kind is then stacked over to extract features from larger regions through combining lines and edges into more abstract features informative of characters. Such a stacked structure addresses CAPTCHA security measures such as rotation and font size change by jointly considering features from multiple local regions. The abstract features are then utilized to detect characters in a fully connected layer. We optimize the cross entropy loss to train the model and utilize ReLU activation function and the dropout mechanism to improve the effectiveness of model training.

V. Evaluation

A. CAPTCHA Testbed

Our CAPTCHA testbed comprises text-based CAPTCHA patterns from three different sources (Table II). The first source is WordPress CAPTCHA plugin, a widely used CAPTCHA with over 700,000 installations. Unlike the dark web CAPTCHA, the CAPTCHA patterns in this dataset does not include background security measures. This dataset serves as a benchmark dataset for examining the performance of CAPTCHA breaking methods. The second source is from the dark web. In particular, two sets of CAPTCHA patterns from carding shops (Rescator Typ1 and Type 2) and one set of CAPTCHA patterns from the DNM (Yellow Brick) were identified and labeled. Rescator platform used two types of CAPTCHA patterns. These sources were selected based on the popularity and scale in the dark web as suggested by CTI experts. For each dataset, a TOR-routed crawler was developed to collect 500 CAPTCHA images. The three datasets were labeled and inspected by two CTI experts.

TABLE II
Summary of our CAPTCHA testbed

Category	Source	Amount	Character Length	Character Type
Benchmark CAPTCHA	**WordPress CAPTCHA**	500	4	Digit+Letter
Dark Web CAPTCHA	**Rescator (type 1)**	500	4	Digit
	Rescator (type 2)	500	5	Digit
	Yellow Brick	500	6	Digit+Letter

B. Experiment Setting

We examined the overall CAPTCHA breaking performance of our proposed framework in comparison with the state-of-the-art baseline methods. The experiment sought to gauge the performance under the dark web-specific condition of lacking labeled CAPTCHA images for training. Accordingly, we sample 500 images from each of the dark web and benchmark CAPTCHA data sources to serve as the evaluation testbed. We compared the CAPTCHA breaking methods' performance for both benchmark CAPTCHA and dark web CAPTCHA patterns. The CAPATCHA breaking performance was measured by success rate, which has been commonly used for evaluation by prior CAPTCHA research [22] [7]. In particular, success rate computes the percentage of CAPTCHA patterns successfully solved by each method [7]. A successful attempt of CAPTCHA breaking is defined as correctly recognizing 'all' characters showing in the CAPTCHA pattern. Moreover, three state-of-the-art methods were used as baselines.

- Image-level CNN only [14]
- Image-level CNN with preprocessing (gray-scale conversion, normalization, and Gaussian smoothing) [12]
- Character-level CNN with interval-based segmentation [13]

C. Experiment Results

As shown in Table III, DW-GAN outperforms the state-of-the-art methods on all datasets with statistically significant

margins measured by t-test (significant at 0.05:*, 0.01:**, 0.001:***). On the dark web CAPTCHA testbed, our DW-GAN achieved 92.08% success rate on Rescator 1, 97.50% success rate on Rescator 2, and 95.98% success rate on Yellow Brick. Among the baseline methods, preprocessing and character segmentation were both effective, consistent with prior literature. Comparing the method in [14] against [12], we find that preprocessing improved the success rate across all datasets. In particular, preprocessing improved CAPTCHA breaking success rate by at least 0.99% (on Rescator 2). Moreover, character segmentation achieved significantly higher success rate on all datasets. Specifically, we observed success rate increases of at least 17.62% by comparing the method in [13] against [12]. In general, the experiment results suggest that our proposed approach of coupling GAN-based background denoising with character segmentation significantly improves the success rate across all datasets and that the improvement are attributed to the background denoising and character segmentation components.

TABLE III
EVALUATION RESULT OF BREAKING DARK WEB CAPTCHAS

Methods	**Benchmark CAPTCHA**	**Dark Web CAPTCHA**		
	WordPress CAPTCHA	**Rescator 1**	**Rescator 2**	**Yellow Brick**
Image-level CNN only [14]	7.72%	63.57%	35.05%	5.88%
Image-level CNN + Preprocessing [12]	8.12%	70.5%	36.04%	7.84%
Character-level CNN + Segmentation [13]	84.16%	88.12%**	77.23%**	93.72%*
DW-GAN (Ours)	**85.15%**	**92.08%***	**97.50%****	**95.98%****

VI. CONCLUSION AND FUTURE DIRECTIONS

Text-based CAPTCHA breaking has been a major bottleneck for large-scale dark web CTI monitoring due to the particularly noisy background and the variable character length of CAPTCHA patterns. Leveraging GAN and CNN, we propose a novel framework for breaking text-based CAPTCHA in the dark web. The proposed framework utilizes GAN to counteract background security measures for dark web-specific CAPTCHA and leverages an enhanced border tracing algorithm to represent CAPTCHA segments as single characters. Our DW-GAN was evaluated on both benchmark and dark web CAPTCHA datasets. DW-GAN outperformed baseline methods on all datasets, particularly on the dark web datasets, where there is a lack of labeled CAPTCHA data. Our proposed research could be further extended by addressing other anti-crawling measures. For example, some emerging dark web platforms complement the text-based CAPTCHA with a 'question-answering' challenge to devise a more complex security measure. For future research, our DW-GAN can be enhanced with automated question-answering to be able to counteract these types of dark web CAPTHCA as well.

REFERENCES

[1] H. Chen, *Dark web: Exploring and data mining the dark side of the web*. New York: Springer, 2012.

[2] M. Ebrahimi, J. F. Nunamaker Jr, and C. Hsinchun, "Semi-Supervised Cyber Threat Identification in Dark Net Markets: A Transductive and Deep Learning Approach," *Journal of Management Information Systems*, vol. 37, no. 3, 2020.

[3] C. Tsosie, "Discover launches social security number alert features," Available at http:// https://www.nerdwallet.com/article/credit-cards/discover-feature-alerts-social-security-number-risky-sites-dark-web (2020/07/21).

[4] A. Sapienza, S. K. Ernala, A. Bessi, K. Lerman, and E. Ferrara, "Discover: Mining online chatter for emerging cyber threats," in *Companion Proceedings of the The Web Conference 2018*, 2018, pp. 983–990.

[5] M. Ebrahimi, M. Surdeanu, S. Samtani, and H. Chen, "Detecting cyber threats in non-english dark net markets: A cross-lingual transfer learning approach," in *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2018, pp. 85–90.

[6] H. Weng, B. Zhao, S. Ji, J. Chen, T. Wang, Q. He, and R. Beyah, "Towards understanding the security of modern image captchas and underground captcha-solving services," *Big Data Mining and Analytics*, vol. 2, no. 2, pp. 118–144, 2019.

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

[8] Y. Zhang, H. Gao, G. Pei, S. Luo, G. Chang, and N. Cheng, "A survey of research on captcha designing and breaking techniques," in *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 2019, pp. 75–84.

[9] X. Wu, S. Dai, Y. Guo, and H. Fujita, "A machine learning attack against variable-length chinese character captchas," *Applied Intelligence*, vol. 49, no. 4, pp. 1548–1565, 2019.

[10] J. Chen, X. Luo, Y. Guo, Y. Zhang, and D. Gong, "A survey on breaking technique of text-based captcha," *Security and Communication Networks*, vol. 2017, 2017.

[11] X. Ling-Zi and Z. Yi-Chun, "A case study of text-based captcha attacks," in *2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*. IEEE, 2012, pp. 121–124.

[12] Z. Nouri and M. Rezaei, "Deep-captcha: a deep learning based captcha solver for vulnerability assessment," *Available at SSRN 3633354*, 2020.

[13] M. Tang, H. Gao, Y. Zhang, Y. Liu, P. Zhang, and P. Wang, "Research on deep learning techniques in breaking text-based captchas and designing image-based captcha," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2522–2537, 2018.

[14] T. A. Le, A. G. Baydin, R. Zinkov, and F. Wood, "Using synthetic data to train neural networks is model-based reasoning," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 3514–3521.

[15] R. Hussain, H. Gao, and R. A. Shaikh, "Segmentation of connected characters in text-based captchas for intelligent character recognition," *Multimedia Tools and Applications*, vol. 76, no. 24, pp. 25 547–25 561, 2017.

[16] R. Shanmugamani, *Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras*. Packt Publishing Ltd, 2018.

[17] H. Gao, W. Wang, J. Qi, X. Wang, X. Liu, and J. Yan, "The robustness of hollow captchas," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 1075–1086.

[18] D. D. Ferreira, L. Leira, P. Mihaylova, and P. Georgieva, "Breaking text-based captcha with sparse convolutional neural networks," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2019, pp. 404–415.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[20] M. Yadav and A. Kumar, "Feature extraction techniques for handwritten character recognition," *International Journal of Advanced Research in Computer Science*, vol. 9, no. 2, p. 521, 2018.

[21] A. H. Pratomo, A. F. Nugraha, J. Siswantoro, and M. F. Nasruddin, "Algorithm border tracing vs scanline in blob detection for robot soccer vision system." *International Journal of Advances in Soft Computing & Its Applications*, vol. 11, no. 3, 2019.

[22] E. Bursztein, M. Martin, and J. Mitchell, "Text-based captcha strengths and weaknesses," in *Proceedings of the 18th ACM conference on Computer and communications security*, 2011, pp. 125–138.