Reassessing CAPTCHAs in the Era of Advanced Deep Learning

Jaouhar Fattahi[1] ; Feriel Sghaier[2] ; Mohamed Mejri[1] ; Ridha Ghayoula[3] and Nadia Mesghouni[4]

***Abstract*— CAPTCHAs, used to be an important element of security, are now facing more challenges due to the great advancements in artificial intelligence. In this paper, we investigate whether CAPTCHAs are still effective in protecting websites from automated threats. A deep learning model is suggested to automatically recognize CAPTCHA embedded characters, with performance metrics achieving an accuracy of 99.46%, an AUC of 99.98%, a precision of 99.46% and a recall of 99.43%. These findings highlight the increasing susceptibility of CAPTCHAs to sophisticated AI driven attacks and seek to emphasize the pressing importance of reevaluating CAPTCHA technologies to guarantee sustainable security.**

***Index Terms*— CAPTCHA, Bot, Deep Learning, Cybersecurity.**

I. Introduction

For long years, CAPTCHAs (Completely Automated Public Turing tests to tell Computers and Humans Apart)) [1], [2] have been considered as a tool to keep websites and digital spaces secure by separating human users from automated bots at the gate of various websites and apps. CAPTCHAs serve as a barrier against entry and spamming activities while offering a to use and flexible solution to protect online platforms. CAPTCHAs serve the purpose of differentiating between users and automated bots by presenting challenges that are easy for humans but tough for machines to crack computationally. These challenges come in several forms like text-based puzzles where users need to interpret characters, image-based puzzles that ask users to pick out specific objects from a group of pictures (for instance, "Choose all the images containing traffic lights") and logical riddles, like basic math questions. Different variations involve audio CAPTCHAs that offer spoken words, for transcription, as well as behavior-based CAPTCHAs that study user actions, like mouse movements, to understand behavior patterns. Essentially, a CAPTCHA functions by creating a challenge on the server end, through algorithms showing this challenge to the user on their device and confirming the users reaction. For instance, a server could generate an image containing text, then, check the users input against the original text to confirm it. Valid input lets the user move forward. Incorrect input leads to rejection. CAPTCHAs are extensively used across scenarios such as stopping spam submissions in web forms introducing security measures and authentication setups hindering bots from launching brute force attacks or data scraping operations, enforcing fairness in surveys and thwarting automated account signups, or ticket scalping activities within e-commerce platforms. Despite being simple and flexible, in nature CAPTCHAs encounter obstacles. Over-sophisticated CAPTCHAs might annoy users. Visual or auditory CAPTCHAs could discriminate against people with disabilities affecting their access. Despite the difficulties involved in facing these obstacles, CAPTCHAs continue to serve as a tool in protecting digital platforms from harm. Table I sums up the traditional roles played by CAPTCHAs in mitigating attacks. The extraordinary advancement in deep learning [3] has completely changed fields such as image recognition and pattern analysis, and thus constitutes a huge threat to CAPTCHAs. State-of-the-art deep neural networks (DNN) can recognize the blurred texts and process the complex images; thus, they can defeat the CAPTCHAs mechanisms. Some of the strategies that bots employ include the use of CNNs or transfer learning which enables the bots to adapt to new CAPTCHA formats easily. Other techniques like reinforcement learning help the bots solve the interactive CAPTCHAs, as well, in a manner that is almost similar to how a human would proceed. This evolution shows that there is a rising dilemma for cybersecurity as it is becoming more expensive and difficult to develop secure CAPTCHAs. This paper explores the overlap between the progressing AI technologies and the role of CAPTCHAs in cybersecurity, moving forward as their effectiveness is being challenged by AI advancements. It proposes an effective deep learning model that almost recognizes all Captach scripts and could perfectly be utilized by bots. It, hence, attracts the attention to the necessity of reevaluating the significance of CAPTCHAs in today's cybersecurity systems [4]–[7]. The end goal of this paper is to animate the conversation about the limitations of CAPTCHAs when dealing with advancing AI risks in order to promote the consideration of stronger security measures to safeguard platforms.

II. Experiment

A. Deep learning model

Our deep learning model consists of 10 layers involving a Dense layer with 128 units, connected to to another Dense layer with 64 units and culminating in an output Dense layer with a single unit outcome result. Subsequently, the data is transformed into a 1 vector through the use of a Flatten layer before incorporating dropout mechanisms for

[1]Jaouhar Fattahi and Mohamed Mejri are with the Department of Computer Science and Software Engineering, Laval University, Quebec, Canada. Jaouhar.Fattahi.1@ulaval.ca; Mohamed.Mejri@ift.ulaval.ca

[2]Feriel Sghaier is with the Carthage National Engineering School, Carthage University, Tunis, Tunisia. Feriel.Sghaier@enicar.ucar.tn

[3]Ridha Ghayoula is with the Faculty of Engineering, University of Moncton, New Brunswick, Canada. Ridha.Ghayoula@umoncton.ca

[4]Nadia Mesghouni is with the Ecole Nationale des Sciences de l'informatique, Université de la Manouba, Tunis, Tunisia. Nadia.Mesghouni@gmail.com

TABLE I: Roles of CAPTCHAs in mitigating security risks

Type of Attack	Role	Explanation
Brute force login attacks [8]	Prevents automated login attempts by requiring human verification.	CAPTCHAs ensure that bots cannot try combinations of usernames and passwords.
Spam in forms or comments [9], [10]	Stops automated spam submissions.	CAPTCHAs ensure that only humans can submit forms, reducing spam entries.
Ticket scalping [11]	Prevents bots from acquiring tickets in bulk.	CAPTCHAs imposes an additional verification layer to ensure fair access to limited goods.
Credential stuffing [12], [13]	Prevents automated attempts to use stolen credentials.	By imposing human verification, CAPTCHAs prevents bots from testing stolen username-password pairs.
Web scraping [14], [15]	Protects against unauthorized data extraction.	CAPTCHAs block bots from harvesting data from websites.
Fake account creation [16]–[18]	Ensures that accounts are created by humans.	CAPTCHAs help preventing bots from generating multiple fake accounts for malicious purposes.
DDoS Attacks [19]–[21]	Reduces resource exhaustion by verifying human users.	CAPTCHAs ensure that only regular human traffic consumes server resources.

mitigating overfitting issues. Moving forward entails adding a layer containing 1500 units, followed by an activation ReLU layer and another dropout layer, for additional regularization purposes. The last Dense layer produces 19 units to signify the target categories and applies sigmoid activation, for class classification purposes. Table II provides additional information about the model layers. Fig. 1 provides a visual architecture of the model layers and their connections. With respect to training, Table III provides perception on the evolution of metrics over epochs (130 epochs were used).

TABLE II: Model layer details

Layer (type)	Output Shape	Explanation
dense (Dense)	(None, 40, 20, 128)	Fully connected layer with 128 units
dense_1 (Dense)	(None, 40, 20, 64)	Fully connected layer with 64 units
dense_2 (Dense)	(None, 40, 20, 1)	Output layer with a single unit
flatten (Flatten)	(None, 800)	Flattens input into 1D vector
dropout (Dropout)	(None, 800)	Dropout layer to prevent overfitting
dense_3 (Dense)	(None, 1500)	Fully connected layer with 1500 units
activation (Activation='relu')	(None, 1500)	Activation function applied
dropout_1 (Dropout)	(None, 1500)	Dropout layer to prevent overfitting
dense_4 (Dense)	(None, 19)	Fully connected layer with 19 units
activation_1 (Activation='sigmoid')	(None, 19)	Final activation layer

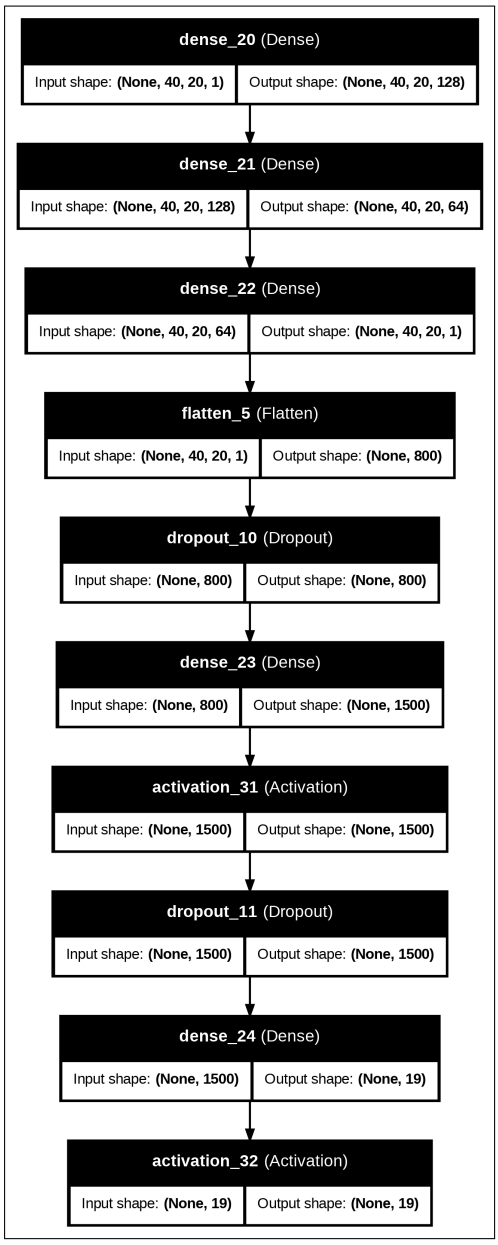


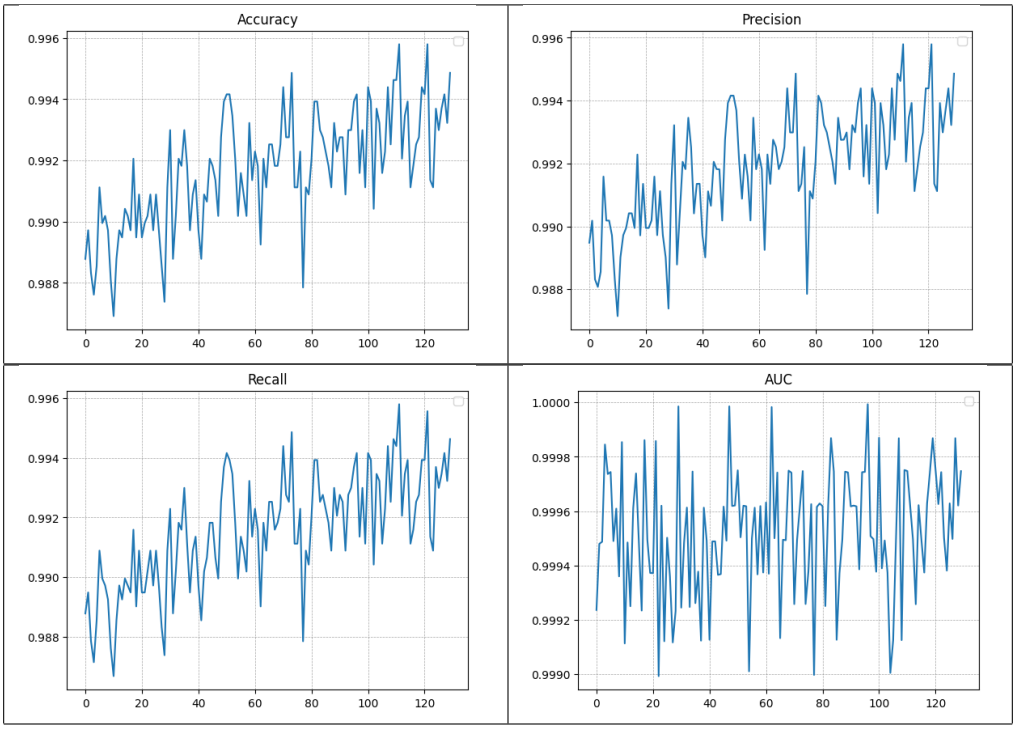
Fig. 1: Model architecture

B. Dataset and data preprocessing

In our experiment, we used the CAPTCHA ImagesV2 dataset [22]. The dataset initially contains 1070 CAPTCHA images, each is of size 200x50 pixels and stored in PNG format. Each CAPTCHA contains 5 alphanumerical characters. The file name is the same as the characters contained in the image. We applied the nine following preprocessing steps to enhance the CAPTCHA images:

1) Adaptive thresholding to convert all images to binary and remove uneven lighting;
2) Morphological closing to fill small holes and remove noise;
3) Dilation to expand white regions and make characters more distinguishable;

TABLE III: Model training and metric evolution over epochs



4) Gaussian blur to reduce noise and smooth edges;
5) Splitting each CAPTCHA image into five character segments using slicing. This results in a dataset of 5350 items, overall;
6) Extracting labels for each character from the image filenames;
7) Normalizing the images by scaling pixel values to the range [0, 1];
8) Label encoding to convert characters to integers;
9) One-Hot encoding to convert integers into categorical format (19 categories overall corresponding to the 19 characters used in all CAPTCHA images : ['d', 'w', '3', 'n', 'c', '7', '5', 'e', 'p', 'y', 'b', '8', 'g', 'f', 'm', '2', '4', 'x', '6']).

Finally, we split the dataset into training and testing subsets with a 75-25 split as shown in Fig. 2.

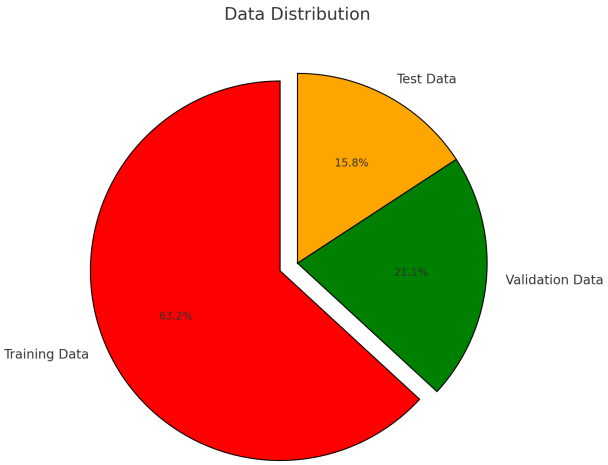


Fig. 2: Dataset repartition

C. Experiment environment setup

During our experiment, we used the Google Colab environment. We used Python 3 along with Keras and TensorFlow, as well as CV2, to create and evaluate our deep learning model and process our data. The cloud based system of Google Colab, along with its free GPU, allowed us to train and test quickly. Python 3 gave us an adaptable programming environment that worked well with the recent libraries.

D. Used metrics

In our experiment, we used the following metrics to assess our model: Accuracy, Precision, Recall, and AUC (Area Under the Curve). Below are the formulas and purposes of each metric, where *TP* are the true positives, *TN* the true negatives, *FP* the false positives and *FN* the false positives :

- Accuracy : it measures the overall correctness of the model by calculating the ratio of correctly predicted instances (true positives and true negatives) to the total number of predictions. It is defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

- Precision : it evaluates the model's ability to correctly identify positive instances, focusing on minimizing false positives. It is defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

- Recall : it measures the capability of the model to recognize all real positive instances, emphasizing reducing false negatives. It is defined as follows:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

- AUC : it represents the area under the Receiver Operating Characteristic (ROC) curve, which plots the true positive rate against the false positive rate. It calculates the capability of the model to make the difference between classes.

III. Results

As we can see it in Fig. 3, our ten-layered cracker model achieves an accuracy of 99.46%, an AUC of 99.98%, a precision of 99.46% and a recall of 99.43% for previously unseen CAPTCHA images. The outcomes pinpoint how the model can handle CAPTCHA effectively and reliably decodes distorted and noisy CAPTCHA images, as shown in Table IV. The impressive results demonstrate the usability of the model for automated CAPTCHA recognition purposes within bots.

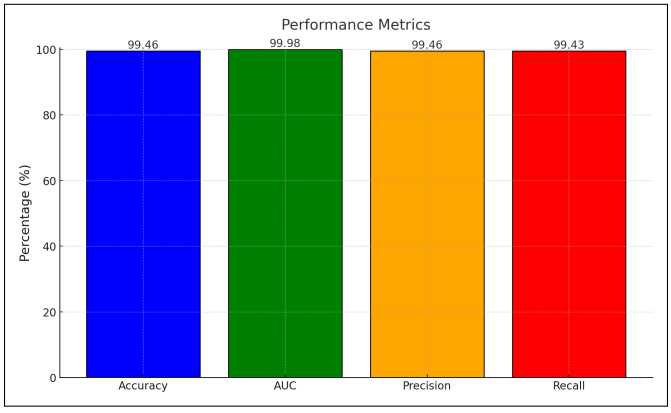


Fig. 3: Model performance

IV. Discussion and related work

Our finding outlined in this paper highlights how CAPTCHA mechanisms can be defeated by bots using advanced deep learning techniques. This serves as an additional warning that echoes the worries expressed in related studies [23]–[25].

To reinforce CAPTCHA security, key players in the field have put forward remarkable efforts. A prominent illustration is reCAPTCHA v.2 [26] from Google which aims to differentiate between users and automated bots. It is heavily based on cookie and browser history data. It has been widely embraced on the internet to safeguard websites and servers from unwanted spam, cloud-based service attacks [27] and DDoS attacks [28]. Introduced as a better version of reCAPTCHA v.1, deemed vulnerable [29], [30], it provides users with a smoother experience while successfully identifying bots and reducing inconvenience for genuine users. Despite all that, Plesner et al. [31], as well as other related work [32], have recently declared that reCAPTCHA v.2 has already been broken by leveraging advanced YOLO models for image segmentation and classification.

V. Conclusion

This research reveals that CAPTCHAs are no longer as effective in thwarting automated threats as they once were believed to be. A learning model showcased performance in deciphering CAPTCHAs with almost perfect effecacy and highlights how artificial intelligence advancements are chipping away at the reliability of this long standing security measure. Although CAPTCHAs have been a security tool for years the found susceptibility raises doubts about their durability, in a landscape shaped by ever evolving AI technologies. As AI technology advances further and further ahead, the challenges of creating CAPTCHAs that can effectively ward off intrusions are becoming increasingly complex and costly. These expenses might soon surpass the benefits they offer. It is crucial for cybersecurity experts to acknowledge these constraints and seek out approaches to combat automated bots. This research is aiming to draw the attention of experts in cybersecurity to think outside the box and come up with other defense mitigation mechanisms.

References

[1] Y. Chow, W. Susilo, and P. Thorncharoensri, "CAPTCHA Design and Security Issues," in *Advances in Cyber Security: Principles, Techniques, and Applications* (K. Li, X. Chen, and W. Susilo, eds.), pp. 69–92, Springer, 2019.

[2] D. Brodic and A. Amelio, *The CAPTCHA: Perspectives and Challenges - Perspectives and Challenges in Artificial Intelligence*, vol. 162 of *Smart Innovation, Systems and Technologies*. Springer, 2020.

[3] J. Fattahi, "Machine Learning and Deep Learning Techniques used in Cybersecurity and Digital Forensics: a Review," *arXiv e-prints*, p. arXiv:2501.03250, Dec. 2024.

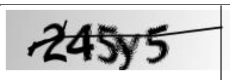
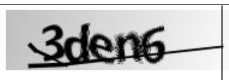
[4] J. Fattahi, B. E. Lakdher, M. Mejri, R. Ghayoula, E. Manai, and M. Ziadia, "Fingfor: a deep learning tool for biometric forensics," in *10th International Conference on Control, Decision and Information Technologies, CoDIT 2024, Vallette, Malta, July 1-4, 2024*, pp. 1667–1672, IEEE, 2024.

[5] J. Fattahi, F. Sghaier, M. Mejri, R. Ghayoula, E. Pricop, and B. E. Lakdher, "Handwritten signature recognition using parallel cnns and transfer learning for forensics," in *10th International Conference on Control, Decision and Information Technologies, CoDIT 2024, Vallette, Malta, July 1-4, 2024*, pp. 1697–1702, IEEE, 2024.

[6] J. Fattahi, O. Fkiri, M. Mejri, and R. Ghayoula, "Hands and palms recognition by transfer learning for forensics: A comparative study," in *New Trends in Intelligent Software Methodologies, Tools and Techniques - Proceedings of the 23rd International Conference on New Trends in Intelligent Software Methodologies, Tools and Techniques (SoMeT_24), Cancun, Mexico, September 24-26, 2024* (H. Fujita, H. M. P. Meana, and A. Hernandez-Matamoros, eds.), vol. 389 of *Frontiers in Artificial Intelligence and Applications*, pp. 213–225, IOS Press, 2024.

[7] E. Manai, M. Mejri, and J. Fattahi, "Fingerprint fraud explainability using grad-cam for forensic procedures," in *New Trends in Intelligent Software Methodologies, Tools and Techniques - Proceedings of the 23rd International Conference on New Trends in Intelligent Software Methodologies, Tools and Techniques (SoMeT_24), Cancun, Mexico, September 24-26, 2024* (H. Fujita, H. M. P. Meana, and A. Hernandez-Matamoros, eds.), vol. 389 of *Frontiers in Artificial Intelligence and Applications*, pp. 457–470, IOS Press, 2024.

TABLE IV: Model prediction (sample)

				
Ground truth : 245y5 Predicted : 245y5	Ground truth : 3den6 Predicted : 3den6	Ground truth : 2nf26 Predicted : 2nf26	Ground truth : p2dw7 Predicted : p2dw7	Ground truth : ypw3d Predicted : ypw3d

[8] X. Liu and X. Hu, "Research on techniques for detecting brute-force attacks on corporate email," *J. Comput. Methods Sci. Eng.*, vol. 24, no. 3, pp. 1379–1393, 2024.

[9] V. Alves and J. Ribeiro, "Detection and Classification of Spam in Social Media Comments Using Artificial Intelligence - A Case Study," in *Progress in Artificial Intelligence - 23rd EPIA Conference on Artificial Intelligence, EPIA 2024, Viana do Castelo, Portugal, September 3-6, 2024, Proceedings, Part II* (M. F. Santos, J. Machado, P. Novais, P. Cortez, and P. M. Moreira, eds.), vol. 14968 of *Lecture Notes in Computer Science*, pp. 311–323, Springer, 2024.

[10] H. Oh, "Corrections to A YouTube Spam Comments Detection Scheme Using Cascaded Ensemble Machine Learning Model," *IEEE Access*, vol. 10, p. 40860, 2022.

[11] H. Yang, H. Lee, and H. Hsiao, "Poster: Challenges in Stopping Ticket Scalping Bots, booktitle = ASIA CCS '20: The 15th ACM Asia Conference on Computer and Communications Security, Taipei, Taiwan, October 5-9, 2020," pp. 931–933, ACM, 2020.

[12] Q. Zhang, "Detecting Credential Stuffing Between Servers," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage - SpaCCS 2020 International Workshops, Nanjing, China, December 18-20, 2020, Proceedings* (G. Wang, B. Chen, W. Li, R. D. Pietro, X. Yan, and H. Han, eds.), vol. 12383 of *Lecture Notes in Computer Science*, pp. 454–464, Springer, 2020.

[13] C. Tankard, "Credential stuffing - the new hack," *Netw. Secur.*, vol. 2021, no. 2, p. 20, 2021.

[14] E. Chiapponi, *Detecting and Mitigating the New Generation of Scraping Bots. (Détecter et neutraliser la nouvelle génération de robots de grattage web)*. PhD thesis, Sorbonne University, Paris, France, 2023.

[15] Y. M. Abrenica, L. Bautista, J. M. Caparas, L. Donato, and G. L. Intal, "ScrapifyHomes: A Study on the Assessment of a Real Estate Web Scraping Data for Lead Generation using SWOT Analysis," in *Proceedings of the 2024 7th International Conference on Information Management and Management Science, IMMS 2024, Beijing, China, August 23-25, 2024*, pp. 150–155, ACM, 2024.

[16] S. G. Fahmy, S. AbdelGaber, O. H. Karam, and D. S. Elzanfaly, "Modeling the Influence of Fake Accounts on User Behavior and Information Diffusion in Online Social Networks," *Informatics*, vol. 10, no. 1, p. 27, 2023.

[17] M. M. Swe and N. N. Myo, "Blacklist Creation for Detecting Fake Accounts on Twitter," *Int. J. Networked Distributed Comput.*, vol. 7, no. 1, pp. 43–50, 2018.

[18] S. P and Shankaraiah, "Social Behavioral Biometric Multimodal Union to Evade Fake Account Creation in Facebook," *Multim. Tools Appl.*, vol. 81, no. 27, pp. 39715–39751, 2022.

[19] W. Alhalabi, A. Gaurav, V. Arya, I. F. Zamzami, and R. A. Aboalela, "Machine Learning-Based Distributed Denial of Services (DDoS) Attack Detection in Intelligent Information Systems," *Int. J. Semantic Web Inf. Syst.*, vol. 19, no. 1, pp. 1–17, 2023.

[20] S. Onyshchenko, O. Haitan, A. Yanko, Y. Zdorenko, and O. Rudenko, "Method for detection of the modified DDoS cyber attacks on a web resource of an Information and Telecommunication Network based on the use of intelligent systems," in *Proceedings of the Modern Data Science Technologies Workshop (MoDaST-2024), Lviv, Ukraine, May 31 - June 1, 2024* (M. T. M. Emmerich, V. Lytvyn, and V. Vysotska, eds.), vol. 3723 of *CEUR Workshop Proceedings*, pp. 219–235, CEUR-WS.org, 2024.

[21] D. Ameyed, F. Jaafar, and J. Fattahi, "A Slow Read Attack using Cloud," in *2015 7th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Bucharest, Romania, June 25-27, 2015*, IEEE, 2015.

[22] P. Fournier, R. Wilhelmy, and H. Rosas, "CAPTCHA Images - Version 2." (Online), https://www.kaggle.com/datasets/fournierp/captcha-version-2-images. Last accessed on 1/1/2025.

[23] M. Moradi, M. Moradi, S. Palazzo, F. Rundo, and C. Spampinato, "Image CAPTCHAs: When Deep Learning Breaks the Mold," *IEEE Access*, vol. 12, pp. 112211–112231, 2024.

[24] B. Zhang, Y. Xiong, C. Xia, and Y. Gao, "Transformer-based end-to-end attack on text CAPTCHAs with triplet deep attention," *Comput. Secur.*, vol. 146, p. 104058, 2024.

[25] H. M. Kanoosh, A. F. Abbas, N. N. Kamal, Z. M. Khadim, D. A. Majeed, and S. Algburi, "Image-Based CAPTCHA Recognition Using Deep Learning Models," in *Proceedings of the Cognitive Models and Artificial Intelligence Conference, AICCONF 2024, İstanbul, Turkiye, May 25-26, 2024*, pp. 273–278, ACM, 2024.

[26] O. Gaggi, "A study on Accessibility of Google ReCAPTCHA Systems," in *OASIS@HT 2022: Open Challenges in Online Social Networks, Barcelona, Spain, 28 June 2022* (B. Guidi, A. Michienzi, and L. Ricci, eds.), pp. 25–30, ACM, 2022.

[27] H. B. Abubaker, K. Salah, H. Al-Muhairi, and A. Bentiba, "Architectural Design of a Cloud-based reCAPTCHA Service," in *2016 12th International Conference on Innovations in Information Technology (IIT), Al-Ain, United Arab Emirates, November 28-30, 2016*, pp. 1–6, IEEE Computer Society, 2016.

[28] M. Poongodi, V. Vijayakumar, F. Al-Turjman, M. Hamdi, and M. Ma, "Intrusion Prevention System for DDoS Attack on VANET With reCAPTCHA Controller Using Information Based Metrics," *IEEE Access*, vol. 7, pp. 158481–158491, 2019.

[29] P. Baecher, N. Büscher, M. Fischlin, and B. Milde, "Breaking reCAPTCHA: A Holistic Approach via Shape Recognition," in *Future Challenges in Security and Privacy for Academia and Industry - 26th IFIP TC 11 International Information Security Conference, SEC 2011, Lucerne, Switzerland, June 7-9, 2011. Proceedings* (J. Camenisch, S. Fischer-Hübner, Y. Murayama, A. Portmann, and C. Rieder, eds.), vol. 354 of *IFIP Advances in Information and Communication Technology*, pp. 56–67, Springer, 2011.

[30] S. Sano, T. Otsuka, K. Itoyama, and H. G. Okuno, "HMM-based Attacks on Google's ReCAPTCHA with Continuous Visual and Audio Symbols," *J. Inf. Process.*, vol. 23, no. 6, pp. 814–826, 2015.

[31] A. Plesner, T. Vontobel, and R. Wattenhofer, "Breaking reCAPTCHAv2," in *48th IEEE Annual Computers, Software, and Applications Conference, COMPSAC 2024, Osaka, Japan, July 2-4, 2024* (H. Shahriar, H. Ohsaki, M. Sharmin, D. Towey, A. K. M. J. A. Majumder, Y. Hori, J. Yang, M. Takemoto, N. Sakib, R. Banno, and S. I. Ahamed, eds.), pp. 1047–1056, IEEE, 2024.

[32] D. Wang, M. Moh, and T. Moh, "Using Deep Learning to Solve Google reCAPTCHA v2's Image Challenges," in *14th International Conference on Ubiquitous Information Management and Communication, IMCOM 2020, Taichung, Taiwan, January 3-5, 2020*, pp. 1–5, IEEE, 2020.