International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 15 Issue 3, March-2026

A Survey Paper on CAPTCHA Refinement Using Machine Learning

Prathamesh Naik[1], Jayshree Mahajan[1], Yashraj Lad[1],
Misbah Bagwan[1], Farkhanda Dalal[1]

[1]Department of Computer Engineering, Pune Institute of Computer Technology, Pune, Maharashtra, India,

ABSTRACT

CAPTCHAs require active user interaction for verification purposes. UIDAI (Unique Identification Authority of India) believes that present CAPTCHAs are frustrating to users, leading to a poor user experience, and want to eliminate them with a passive machine learning based approach. Behavior will be monitored as soon as any user visits the website. Session duration, typing speed, mouse trajectory, and keystroke data are some of the parameters used to model the behavior and ensure that these are stored in a database and then sent to a machine learning model for prediction. After processing, the machine learning model will give a prediction that will be sent to the frontend, indicating the result of verification. If verification is successful, then that user will be granted access to the website otherwise, the system will ask for some minimal interaction for verification so that the security of the website is not at all compromised. This project successfully demonstrates a behavioural metrics-based approach for detecting bot interactions using user input patterns. By capturing keystroke dynamics, mouse movement patterns, and interaction delays, meaningful behavioural metrics are extracted that differentiate human users from automated bots.

Keywords: ***Behavioral biometrics, Bot, Browser fingerprinting, CAPTCHA, DDOS, DOS, Keystroke dynamics, Machine learning, Mouse dynamics, Passive verification, Random Forest.***

INTRODUCTION

The Internet has become a crucial part of our lives, offering a large number of web services, such as booking, identity management, and email. However, these services are threatened by bots. Bots are automated computer programs or scripts used for various purposes like indexing and monitoring, but there are malicious bots that can cause denial of service attacks, large scale defamation of an organization, abuse of online services, security threats etc. Thus, to protect online services, it is critical that only human requests reach the server and bot requests are blocked beforehand since the server cannot determine the legitimacy of the request, it only knows to serve the incoming request. This is where CAPTCHA comes in; it stands for "Completely Automated Public Turing test to tell Computers and Humans Apart". It is a security measure that has been very effective since its inception in distinguishing humans and bots. Conventional CAPTCHAs took advantage of the cognitive gap between humans and bots, since bots will never match the sophistication of humans. It contained some distorted text which needed to be deciphered and entered in the required input field for verification. However, the advancements in machine learning and computer vision have greatly reduced this cognitive gap. Deep learning models and optical character recognition techniques existing today are capable of solving many traditional CAPTCHA challenges with high accuracy, which challenges the assumption that states "CAPTCHAs are bot hard and human friendly"

These advancements in bots demand a more robust CAPTCHA; as a result, existing CAPTCHAs are getting more and more difficult. Not only do they frustrate the users, but also cause accessibility and usability issues. They lead to poor resident engagement and impact people with disabilities or low digital literacy. This compromise between security and user experience is not at all acceptable for large-scale platforms, particularly government and identity-based systems. Because of these drawbacks, passive verification systems have become a necessity.

A passive verification system will cause minimal friction to human users and block automated bots by continuously monitoring user behavior right when the user visits the website, it will also provide real time prediction results to the

frontend when the processing is done. This improves user experience, removes the long-standing CAPTCHA barrier, and improves security because of a machine learning model deployed in the backend, which will provide the prediction. If the model fails to provide the prediction, then minimal interaction (like scroll mouse up or down) will be required to prove legitimacy. This passive system ensures that the user experience is improved without compromising security.

A machine learning model is employed for differentiation because bots will evolve and their behavior will keep changing, hence the system must be able to discover new behavior and patterns and also become robust with time. Every request's data that is stored in the database can be used for model training in the future and this will improve the system's overall accuracy and efficiency.

LITERATURE REVIEW

2.1 Traditional CAPTCHA Evolution and Vulnerabilities

The original purpose of CAPTCHA mechanisms was to take advantage of cognitive differences between humans and machines using simple interaction-based puzzles, distorted text, or image recognition. Although early implementations were successful against simple automation, their robustness has been severely undermined by the quick development of machine learning and computer vision techniques. Research shows that many conventional designs are no longer effective because deep neural networks and optical character recognition systems can now solve text-based and image-based CAPTCHAs with high accuracy.

Additionally, recent studies demonstrate that without task-specific training, generalized vision-language models can achieve high success rates across multiple CAPTCHA categories. The long-held belief that CAPTCHA challenges are intrinsically bot-hard is challenged by this development, which exposes a basic weakness in challenge-response security models [13]. Measurement-based studies also show that automated tools and human-assisted CAPTCHA farms enable widespread, real-world CAPTCHA abuse, highlighting the fact that conventional CAPTCHA systems no longer offer dependable defense against sophisticated attacks [14].

2.2 Usability and Accessibility Challenges

CAPTCHA systems, through their security-focused design, face problems that make it difficult for genuine users to access their services. Multiple studies report increased user frustration, task abandonment, and reduced engagement due to repeated or complex CAPTCHA challenges. The accessibility problems create difficulties for users who have disabilities or restricted digital skills or experience low-bandwidth internet access. CAPTCHA providers work to solve these problems by decreasing challenge frequency or creating solutions that do not need users to solve puzzles. Research shows that these solutions still depend on behavioral data collection because they remain vulnerable to both human and artificial intelligence-based attacks [14]. The transition to passive and invisible bot detection systems has occurred because usability problems force organizations to implement these systems as their main solution.

2.3 Machine Learning for Bot Detection

Supervised Machine learning is also mostly used for bot detection since it has been proven that supervised Machine learning is able to learn complex human behavior patterns [1]. In passive CAPTCHA systems, ensemble learning is mostly used for modeling the learning of users in real-time [5], and research has shown that Machine learning is effective against advanced CAPTCHA attacks [6]. The supervised learning methods Random Forests, Support Vector Machines, and ensemble classifiers have shown their ability to identify human users and bots through analysis of their behavioral and environmental characteristics [1], [5]. Recent research demonstrates that machine learning models which develop their training from current interactive data can deliver effective web platform detection at high speed, which enables their use across extensive web platforms [4]. The combination of various feature sources through hybrid frameworks which include user interaction patterns and web log data establishes a better system to handle bots that try to hide their presence and those who use browser fingerprinting techniques [12].

2.4 Behavioral Biometrics in Authentication

Users exhibit natural and distinct interaction patterns to create behavioral biometrics that prevent bots from replicating their unique human behavior. Behavioral biometrics provide both security and usability benefits because they authenticate users through automatic and continuous monitoring without requiring user input.

Mouse Dynamics: Mouse dynamics research studies how people move their computer mouse because it examines different aspects of cursor movement including speed and acceleration and curvature and hesitation and trajectory patterns [10]. Research consistently shows that human mouse movements exhibit natural variability and non-linear behavior whereas bots tend to produce smoother, more predictable trajectories. Machine learning models trained on mouse dynamics features

have proven effective in detecting both basic and advanced bots which include those that attempt to mimic human behavior.

Keystroke Dynamics: Keystroke dynamics analysis centers on three timing-based elements which include dwell time, inter-key delay, and typing rhythm. Studies show that human typing patterns depend on three factors which include cognitive load and skill level and context, which produce inconsistent results that bots find difficult to duplicate. The research examined two methods of keystroke analysis which included fixed-text and free-text approaches, while the new deep learning models and large-scale datasets improved detection accuracy for bot identification tests [11].

2.5 Environmental and Device Fingerprinting

Environmental and device fingerprinting techniques collect contextual information through browser configuration and operating system details and screen resolution and language settings and hardware capabilities. The features create high-entropy identifiers which work together with behavioral biometrics to improve detection performance [7]. Research shows that combining browser fingerprinting with behavioral signals enables more robust bot detection research which specifically targets bots using actual browsers or automation systems to bypass basic fingerprint checks. The studies demonstrate that privacy-aware design needs to show privacy protection because excessive fingerprinting creates both ethical issues and regulatory challenges. The latest methods now support data collection which requires only essential information to power machine learning-based inference systems.

METHODOLOGY

3.1 System Architecture

The proposed system architecture consists of three primary components which create a secure authentication method that scales with user needs while remaining easy for users to operate. The architecture uses background operations to detect human users while using minimal methods to interact with users which distinguishes it from standard CAPTCHA systems. The system consists of four primary parts which include the Frontend Module, Backend Module, Machine Learning Engine, and the Module Interaction and Data Flow layer.

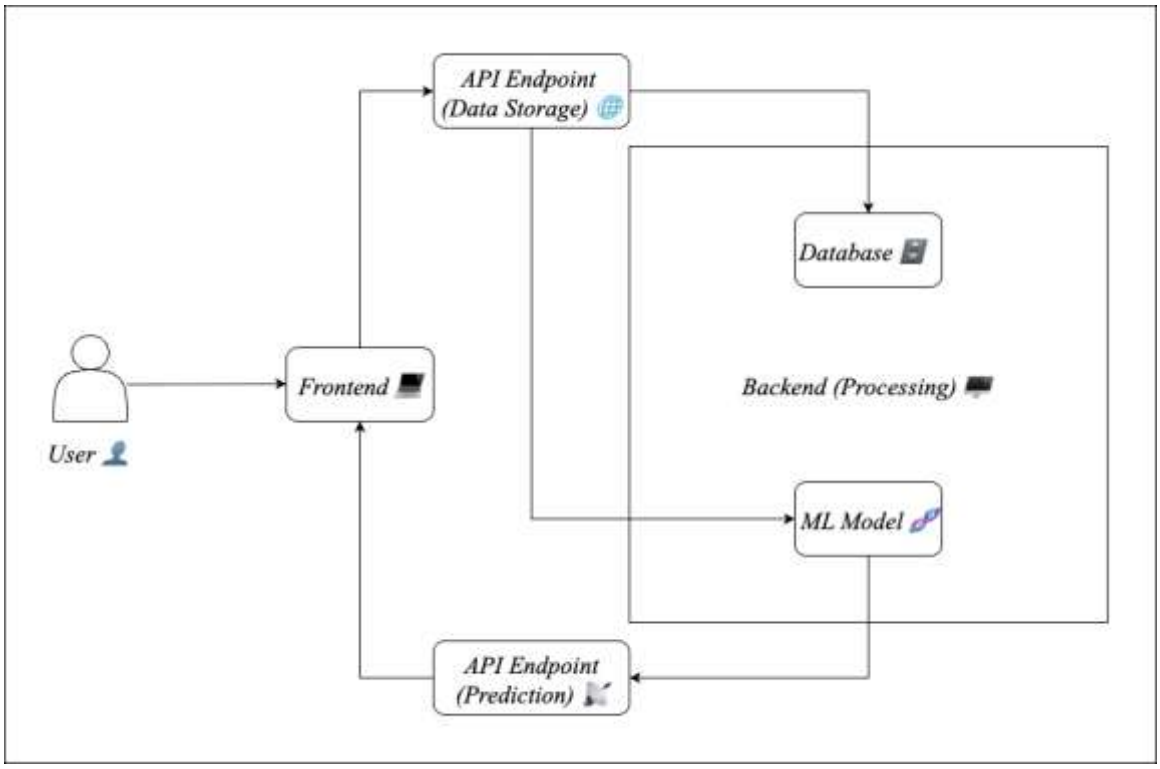

Fig. 1 Proposed System Architecture

3.1.1 Frontend Module

The frontend module collects data from users who interact with a web application through their normal activities. This module operates in the background to collect various user and environmental data through mouse movement patterns,

cursor velocity and acceleration, keystroke timing, browser specifications, screen resolution, and session-level interaction data. The system uses lightweight JavaScript listeners which developers embed in webpages to collect data while maintaining compatibility with standard browsers and delivering minimal disruption to users. Several studies have highlighted that the acquisition of data on the frontend should be unobtrusive, having minimal intrusion or distraction, so that usability and accessibility are not degraded [3], [5]. The system tracks mouse movements together with keystroke patterns as users fill out forms and use website elements, which enables the system to create detailed user behavior profiles that users will not realize. The system collects environmental data which includes browser settings and device specifications to enable fingerprint analysis while maintaining privacy requirements [7].

3.1.2 Backend Module

The backend module functions as an intermediate processing layer that converts raw frontend interaction data into machine learning analysis ready data. The module performs multiple functions which include validating data, removing unwanted data, standardizing data, combining session data, and extracting features from the data. The system transforms raw mouse coordinates and keystroke timestamps into statistical and temporal features, which include speed variance, trajectory curvature, dwell time, and inter-key delay. The backend module manages data storage which is used for model training and auditing and system evaluation. Multiple studies demonstrate that backend systems require low-latency optimization to support real-time decision-making in high-traffic environments [4]. The system implements security and privacy controls at this stage to verify compliance with data protection laws which apply to behavioral and fingerprinting data processing [3].

3.1.3 Machine Learning Engine

The essential decision-making function of the system depends on its machine learning engine which serves as the primary operational component. The system establishes a human-bot classification system through its analysis of processed behavioral and environmental data to assess incoming sessions. The surveyed literature reports the use of supervised learning models such as Random Forests, Support Vector Machines, and ensemble classifiers because these models provide strong performance and clear understanding of their results when they process tabular behavioral data. The detection accuracy against complex evasive bots improves through advanced methods which use deep learning architectures to model intricate temporal patterns present in mouse movement and keystroke data [11]. Hybrid models that combine behavioral biometrics with browser or device fingerprinting have been shown to enhance resilience against bots that mimic human interaction patterns using real browsers or automation frameworks. The machine learning model after its training process becomes an operational system which performs real-time classification through its lightweight inference service with fast response times. Some studies also support adaptive learning, where the model is periodically retrained using newly collected data to counter evolving attack strategies [14].

3.1.4 Module Interactions and Data Flow

The interaction between modules follows a sequential yet tightly integrated data flow. The frontend module first captures user interactive behavioral signals together with environmental signals which it then transmits to the backend through secure channels. The backend module processes incoming data through preprocessing steps to extract essential features which it sends to the machine learning engine for classification tasks. The application layer receives the prediction result which determines whether the entity is human or bot and it enables the system to implement access approval, risk scoring, and secondary verification processes. Several studies show that using multiple data sources together with different system modules results in better detection performance since bots that manage to bypass one detection system will still be caught by the remaining detection systems. The system uses a modular design which enables organizations to expand their operations through separate updates of each component when new security threats and detection methods emerge.

3.2 Working Methodology

3.2.1 Data Collection and Storage

The proposed bot detection framework starts with passive behavioral data collection. The system operates without test-based user authentication by monitoring user behavior throughout their regular web usage. Previous studies show that collecting background data in a transparent way is better for the user experience than the traditional challenge response CAPTCHAs [3]. Passive behavioral CAPTCHAs are more accurate for increasing the detection rate without disturbing the normal interaction of the users [5], [6].

The system uses basic JavaScript event listeners which include mousemove and mousedown and keydown and scroll to gather data on users who visit the website. The system records mouse movements through its tracking system which generates two-dimensional screen position data that includes time stamps to enable researchers to study the movement patterns and speed and acceleration and small hand movements of the mouse cursor. Automated scripts have shown difficulty in replicating these particular behavioral patterns which researchers identified through studies [10], [15].

The system records keystroke dynamics by saving all key press and key release activities together with their exact moment of occurrence. The system extracts three metrics from these raw events which include dwell time and inter-key delay and typing rhythm and these metrics help identify human users from automated bot systems [11]. The system detects two types of user activities which include interface event delays and session-level interaction timing to create a system that captures contextual behavioral cues.

The system gathers environmental data which includes browser type and operating system and screen resolution and device characteristics to enable fingerprint analysis while maintaining the privacy protection methods developed in previous studies [7], [8], [9]. The system collects all behavioral and environmental information which gets combined into a structured JSON object that gets sent securely to the backend for preprocessing and machine learning analysis which enables scalable and non-intrusive detection of bot activity.

3.2.2 Data Preprocessing

The collected raw behavioral data from user interactions cannot be used for machine learning classification because it exhibits two main issues: session length variability and high dimensionality and lack of interpretability. Prior studies show that inserting raw cursor coordinate data into classifiers results in poor generalization performance and higher operational expenses and makes the system vulnerable to overfitting issues [10], [15].

The system transforms raw mouse trajectory data to create statistical representations that solve existing challenges in the system. The Pearson correlation coefficient functions as an effective transformation method which calculates the correlation between X and Y coordinate sequences of mouse movements. This metric measures how horizontal and vertical movements relate to each other because it combines thousands of position samples into a single feature which can be easily understood [15], [16].

Human operators display irregular cursor movement patterns because their natural motor control creates small angular variations and motion pauses and corrective movements, which differ from the precise linear pathing used by bots [10]. The correlation coefficient functions as a behavioral feature that distinguishes between different patterns while remaining constant across different screen resolutions and input device variations.

The Pearson correlation coefficient r can be expressed using the formula given below:

$$r_{XY} = \frac{\sum_{i=1}^{n} (x_i - \underline{x}) (y_i - \underline{y})}{\square}$$

where:

- n: The dataset's total number of observations.
- ΣXY: The sum of the element-wise products of x and x values.
- Σx, Σy: Total of all x-values and x-values.
- Σx^2, Σy^2: Sum of squared x-values and x-values.

Table 1 Interpretation of values for the Pearson Correlation Coefficient

Correlation Range	Interpretation
$0.00 \leq \lvert r \rvert < 0.20$	Negligible/ No correlation
$0.20 \leq \lvert r \rvert < 0.40$	Weak correlation
$0.40 \leq \lvert r \rvert < 0.60$	Moderate correlation
$0.60 \leq \lvert r \rvert < 0.80$	Strong correlation
$0.80 \leq \lvert r \rvert < 1.00$	Very strong correlation

The correlation-based representation provides improved model interpretability while enabling the system to operate with a reduced feature dimensionality. The present study observes that bot-generated trajectories often produce correlation values exceeding 0.80 due to deterministic motion patterns, whereas human interactions typically fall within the 0.20–0.60 range. These observations align with prior research describing the predictable behavior of automated cursor movements and the natural variability present in human interaction dynamics [10], [15]. The distinct behavioral patterns between humans and bots therefore establish a dependable basis for identifying genuine user presence.

The effectiveness of the correlation metric further improves when combined with complementary spatial and temporal behavioral features, including velocity variance and interaction dynamics, forming a multidimensional profile capable of capturing both linear and nonlinear interaction characteristics [16].

3.2.3 Training and Making Predictions with a Model

The sessions get classified into human and bot categories through supervised machine learning methods which require labeled behavioral data for training. The lack of publicly accessible datasets which record detailed behavioral interaction data leads multiple studies to create their own data collection methods for generating accurate training data [1], [2], [4]. The researchers developed a labeled dataset which contains human interaction sessions and bot automated session records according to their methodology.

Researchers in behavioral authentication and CAPTCHA improvement studies prefer Random Forest classifiers because these classifiers can manage noisy data while they effectively process numerous features and maintain accurate results across different situations [1], [5], [6]. The Random Forest model uses decision trees which it trains through bootstrap aggregation to create an ensemble where each tree learns from its individual data and feature space samples.

The ensemble strategy decreases overfitting while increasing system resilience to different user patterns and device operations and various interaction environments. The model delivers feature importance scores which show how different behavioral traits like keystroke timing variability and mouse movement irregularity affect classification results [16].

The trained model processes incoming session features through real-time inference to generate probability scores which determine whether a human or automated agent conducts the interaction. Adaptive retraining mechanisms exist to counter evolving bot strategies based on recent studies about dynamic CAPTCHA and bot detection systems [14].

RESULTS AND DISCUSSION

The existing CAPTCHA improvement methods and bot detection systems undergo the examination process which shows their development from direct challenge response methods toward their present implementation of behavior analytical and machine learning methods. Early CAPTCHA systems used text distortion together with image recognition tasks as their method to distinguish between human users and automated bots. The traditional CAPTCHAs used in security systems now face increased vulnerability to automated attacks because multiple studies prove that deep learning and vision-language model advancements have reduced their effectiveness [6], [13].

The Behavioral CAPTCHA techniques use mouse movement patterns and typing speed patterns as their passive user interaction signals. Research consistently shows that these behavioral biometrics capture natural human irregularities which bots fail to replicate with perfect accuracy. The performance of the Random Forest model and Support Vector Machine model is satisfactory for the behavioral CAPTCHA system [5]. The mouse movement pattern exhibits human behavior that is difficult for bots to replicate or mimic [10], [15]. The keystroke model also helps differentiate between humans and bots [11]. The adaptive behavioral authentication model also improves detection against bot programs [16]. The behavioral approaches experience operational challenges because user device differences and user skill differences and contextual factor differences create data variability.

Browser and device fingerprinting techniques use environmental data which includes the operating system and browser settings and screen resolution and device specifications. Methods of Browser and Device Fingerprinting are highly unique identification techniques that are used to identify bots running automation tools or headless browsers [7]. Studies, however, show that performing fingerprinting in large scale raises privacy issues and problems relating to user consent and legality [8], [9].

The hybrid approach combines behavioral biometrics with environmental fingerprinting and web log analysis to create a system that protects against advanced bot attacks which use stealthy techniques. The hybrid system uses multiple detection methods to overcome the limitations of its single detection method and achieve better results in actual operational

environments [12], [14]. The systems provide effective results but they create problems with architectural complexity and computational demands and difficulties in system upkeep and system expansion.

Overall, recent studies indicate that hybrid behavioral frameworks offer improved security and usability compared to traditional CAPTCHA systems, supporting the transition toward passive, multi-modal machine learning-based authentication methods.

Table 2 Comparative Analysis of CAPTCHA Refinement and Bot Detection Techniques

Approach	Features	Models Used
Traditional CAPTCHA	Text, images, puzzle-based challenges	None
Behavioral CAPTCHA	Mouse dynamics, keystroke timing patterns	RF, SVM, ML classifiers
Fingerprinting	Browser, OS, and device attributes	Statistical and ML-based models
Hybrid Models	Behavioral and environmental features	Ensemble and hybrid ML models

CONCLUSION

The research proposes an improved CAPTCHA anti-spam approach in terms of its behavior-based CAPTCHA enhancement framework, which utilizes user interactions and machine learning to address the decreased effectiveness of traditional challenge-response anti-spam tools. Considering the recent advances in deep learning and automation technologies, the ability of machines in solving CAPTCHA has improved in recent times, especially with the usage of automated tools [6]. Besides, measurement studies reveal the increasing trends of CAPTCHA abuse in the real world, particularly in spamming [14]. In general, vision-language models show high efficiency in addressing different CAPTCHA challenges [13].

The proposed approach uses behavioral signals such as the dynamics of the mouse movement and patterns in the keystroke timing to develop interpretable features of a lower dimension [10], [11], [15]. Environmental features and passive CAPTCHA methods are used to boost the strength of the proposed human-bot classification system and increase its usability from different devices and interaction situations [5], [16]. Ensemble methods and the usage of a Random Forest classifier are found to be more effective in human-bot classification problems while ensuring transparency in the model and resisting overfitting [1], [2], [4].

Behavioral and hybrid approaches, on one hand, provide better scalability, usability, and security features compared to conventional CAPTCHA systems [3], [5]. However, browser and device fingerprinting methods need to be used cautiously because of their potential risks to user privacy, as highlighted by previous studies [7], [8], [9]. On the other hand, passive machine learning-based behavioral authentication remains a promising model for the development of web security systems in the future. Further studies should be carried out regarding its deployments, better data collection mechanisms, and fairness evaluation in diverse populations [16], [17].

REFERENCES

1. R. Kamal Raj, S. Ramesh, and A. Kumar, “Developing an ML-Based Solution to Refine CAPTCHA for UIDAI,” International Journal of Computer Applications, 2025.
2. A. V. Preeya, S. Aslam, S. Reddy, and V. Reddy, “Develop a Machine Learning Model Based Solution to Refine CAPTCHA,” International Journal of Creative Research Thoughts (IJCRT), vol. 13, no. 5, pp. 750–757, 2025.
3. G. C. M., R. K. Nayak, and P. S. Rao, “Develop a Machine Learning Based Solution to Refine CAPTCHA,” International Journal of Emerging Technologies, 2024.
4. T. L. Phanindra, S. Kiran, and M. R. Prasad, “ML Based Solution to Refine CAPTCHA Systems,” International Journal of Advanced Research in Computer Science, 2025.

5. C. Mohitkar, A. Joshi, and R. Kulkarni, “Passive CAPTCHA: AI Driven Bot Detection Using Behavioral Analysis,” International Journal of Information Security, 2025.
6. Dayanand, W. Jeberson, and K. Jeberson, “Machine Learning Defenses for CAPTCHA Systems,” International Journal of Scholarly Research in Multidisciplinary Studies, vol. 4, no. 2, pp. 1–7, 2024.
7. D. Zhang, A. Acar, and F. Li, “A Survey of Browser Fingerprinting Techniques,” ACM Computing Surveys, 2022.
8. [8] U. Iqbal, S. Englehardt, and A. Narayanan, “Detecting and Measuring Browser Fingerprinting,” in Proc. NDSS, 2020.
9. M. Laštovička, T. Jirsík, and P. Čeleda, “Passive Operating System Fingerprinting,” IEEE Communications Surveys & Tutorials, 2023.
10. S. Khan, C. Devlen, M. Manno, and D. Hou, “Mouse Dynamics Behavioral Biometrics: A Survey,” ACM Computing Surveys, 2024.
11. D. DeAlcala, A. Morales, R. Tolosana, A. Acien, J. Fierrez, S. Hernandez, M. A. Ferrer, and M. Diaz, “BeCAPTCHA-Type: Biometric Keystroke Data Generation for Improved Bot Detection,” arXiv preprint arXiv:2207.13394, 2023.
12. C. Iliou, T. Kostoulas, T. Tsikrika, V. Katos, S. Vrochidis, and I. Kompatsiaris, “Detection of Advanced Web Bots by Combining Web Logs with Mouse Behavioural Biometrics,” Digital Threats: Research and Practice, vol. 2, no. 3, Article 24, 2021, doi:10.1145/3447815.
13. X. Teoh, Y. Lin, S. Li, R. Liu, A. Sollomoni, Y. Harel, and J. S. Dong, “Are CAPTCHAs Still Bot-hard? Generalized Visual CAPTCHA Solving with Agentic Vision Language Models,” in Proc. 34th USENIX Security Symposium, 2025.
14. H. D. Nguyen, K. Subramani, B. Acharya, R. Perdisci, and P. Vadrevu, “C-FRAME: Characterizing and Measuring In-the-Wild CAPTCHA Attacks,” in Proc. IEEE Symposium on Security and Privacy, 2024, doi:10.1109/SP54263.2024.00200.
15. N. S. Afanaseva and P. S. Lozhnikov, “Bot Detection Using Mouse Movements,” in Proc. XVII Int. Conf. Dynamics of Systems, Mechanisms and Machines, IEEE, 2023, doi:10.1109/Dynamics60586.2023.10349640.
16. M. Amshavalli, S. Chandiran, R. Dharshini, and A. M. Edwin Arul Solomon, “Adaptive Behavioral Authentication for Bot Detection Using Machine Learning,” International Research Journal on Advanced Science Hub, vol. 7, no. 5, pp. 507–512, 2025, doi:10.47392/IRJASH.2025.057.
17. X. Teoh et al., “Generalized Visual CAPTCHA Solving in the AIGC Era,” Supplementary Extended Version, USENIX Security, 2025.