*Grenze International Journal of Engineering and Technology, June Issue*

# CAPTCHA to Cognition: Replacing Traditional CAPTCHA with Adaptive Behavioral Intelligence using Machine Learning and Deep Learning

Chaitanya R Chaudhari[1], Rohan Masal[2], Vishvajeet Deokar[3], Sarthak Mhase[4] and Suhasini Bhat[5]
[1-5]JSPM'S RSCOE Tathawade, Pune
Email: chaitanyachaudhari4465@gmail.com, rohanmasal58@gmail.com, vishvajeetdeokar8@gmail.com, mhasesarthak62@gmail.com, bhatsuhasini@gmail.com

***Abstract*— With the rapid growth of artificial intelligence, traditional CAPTCHA systems that use text, images, or checkboxes have become more vulnerable to automated solvers and AI-powered bots. In our study, we introduce "CAPTCHA to Cognition," a framework that adapts behavioral analysis to replace static CAPTCHAs with AI-driven cognitive verification. The system captures and analyzes how users interact, including mouse movements, keystroke patterns, scrolling behavior, and response times, to tell human users apart from bots in real-time. By using machine learning and deep learning algorithms like Random Forest, XGBoost, and Multi-Layer Perceptron (MLP), the model achieves high detection accuracy while still providing a smooth user experience. This paper discusses the design, structure, and implementation of this behavioral CAPTCHA system. It highlights its role in improving web security and making access easier through cognitive intelligence.**

***Index Terms*— Behavioral CAPTCHA, Bot Detection, Cognitive Authentication, User Interaction Analysis, Machine Learning, Deep Learning, Web Security.**

## I. Introduction

With the rapid advancements in artificial intelligence and automation, the distinction between human and machine interactions over the internet is becoming narrower by the day. Basic CAPTCHA systems, which had been regarded as a surefire way to distinguish humans from automated scripts in the past, are currently being circumvented by sophisticated AI models and automated programs. Over the years, attackers have conditioned neural networks that can identify distorted text, break image puzzles, and even simulate user clicks—making traditional CAPTCHA measures obsolete and annoying for legitimate users.

To bridge these constraints, we introduce "CAPTCHA to Cognition", a behavioral verification system powered by AI that instead of what problem the user solves, relies on how a user performs. It monitors real-time behavioral metrics like mouse movement, typing rhythm, scrolling speed, click patterns, and dwell time to detect cognitive legitimacy. Web technologies have been developing year after year to accommodate sophisticated analytics, making it possible to create adaptive and smart authentication systems. Utilizing machine learning and deep learning algorithms such as Random Forest, XGBoost, and Multi-Layer Perceptron (MLP), our system learns and enhances detection accuracy over time from past interaction data.

*Grenze ID: 01.GIJET.12.2.253_26*

The suggested framework handles high volumes of behavioral data, preprocesses it, extracts important features, and visualizes analysis results through performance metrics and confidence plots. The objective is to design an invisible, seamless verification layer that raises security and user experience levels. By removing visual challenges and text-based puzzles, this behavioral CAPTCHA provides smoother user experience with high detection accuracy.

## II. Proposed System

The CAPTCHA to Cognition system is a sophisticated web-based behavioral authentication system that is meant to supersede conventional CAPTCHA techniques with cognitively based intelligence. The CAPTCHA to Cognition is a modular architecture that comprises five major components, each dealing with a different layer of handling, analysis, and making decisions on behavioral data.

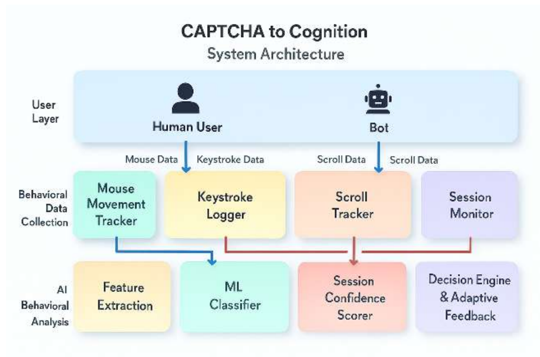


Fig. 1. System Architecture

### *A. System Overview*

The entire mechanism works as an AI-powered behavioral analysis engine embedded in the background of a website's frontend. When a user is interacting with a webpage, the system collects the behavioral data such as mouse movement, scrolling speed, click frequency, typing rhythm, dwell time, and hesitation figures. They are transmitted securely to the backend server where they are monitored by analysis models to determine if the session is that of true human activity or bot-assisted automated activity. The complete technology stack comprises a React.js frontend together with JavaScript listener capture for interaction capture, a Flask backend server for data processing and API management, and MySQL for organized data storage. Machine learning modules are developed with Scikit-learn, XGBoost, and TensorFlow to provide scalability, flexibility, and rapid inference.

### *B. Core Modules*

*Behavioral Data Acquisition Module*
It runs in the browser and continuously records user interaction parameters such as pointer acceleration, typing interval, page dwell time, and navigating behaviors. The recorded data is filtered and coded and then is passed to backend analysis. It works transparently to support user convenience and natural interaction flow.

*Preprocessing and Feature Engineering Module*
After this module receives the behavioral data, the data is pre-processed with noise, outliers, and redundancy reduction. Extracted are features like average velocity, click density, trajectory curvature, and keystroke entropy. These are crucial in discerning human-like irregular movement from the perfect, repeated motion that is typical of bots.

*Machine Learning Classification Module*
This is the analytical heart of the system. It employs three main algorithms Random Forest, XGBoost, and Multi-Layer Perceptron (MLP) to label sessions as bot or human. Random Forest possesses robust ensemble decision-making. XGBoost handles imbalanced datasets efficiently with gradient boosting. Mlp retains intricate non-linear user behavior patterns. These outputs are then assimilated with a weighted voting mechanism for improving prediction quality and stability.

*Adaptive Security and Decision Module*

Based on model confidence scores, the system dynamically determines the authentication response.

High-confidence human behavior → Direct access.

Suspicious or confidence-low sessions → Introduce lightweight cognitive mini-tests (e.g., reactions or drag-based micro-interactions).

Bot detection → Terminated session or blacklisted.

It ensures that the actual users are not affected at all, while scripts are resisted even more.

### *C. Security and Privacy Integration*

The framework allows for a secure processing of information with SHA-256 hashing and AES encryption at both storage and transmission levels. High-risk activities are anonymized to comply with GDPR and DPDP information protection standards. Furthermore, the model is adapted with retraining on anonymized aggregated datasets to offer user privacy and ethically compliant AI.

## III. Mathematical Formulation of Algorithms

The effectiveness of the CAPTCHA to Cognition framework lies in its ability to classify users accurately based on behavioral data. The system integrates multiple machine learning algorithms to form a hybrid model that can adapt to diverse human interaction patterns and resist bot mimicry.

### *A. Random Forest*

The Random Forest algorithm is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of their predictions for classification. It reduces overfitting and enhances accuracy by introducing randomness in feature selection and data sampling. Each decision tree in the forest generates a classification $h_i(x)$ for the input feature vector x. The final prediction H(x) is computed using majority voting as:

$$H(x) = \text{mode}\ \{ h_1(x), h_2(x), h_3(x), \dots, h_n(x)\}$$

Where:

- H(x) = Final predicted class (Human or Bot)
- $h_i(x)$ = Prediction from the i-th decision tree
- n = Number of trees in the forest

The algorithm's random sampling of both data and features ensures diversity among trees, leading to a more generalized classification outcome.

### *B. XGBoost (Extreme Gradient Boosting)*

XGBoost is a gradient boosting algorithm that builds an ensemble of weak learners (typically shallow decision trees) sequentially, with each new tree correcting the errors of the previous ensemble. The objective function to be minimized is given by:

$$Obj = \Sigma_{i=1}^{n} L(y_i, \hat{y}_i) + \Sigma_{k=1}^{K} \Omega(f_k)$$

Where:

- $L(y_i, \hat{y}_i)$ = Loss function measuring the difference between actual and predicted labels
- $\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \|w\|^2$ = Regularization term that penalizes model complexity
- $f_k$ = Individual decision tree
- T = Number of leaves in the tree
- $\gamma$ and $\lambda$ = Regularization coefficients

At each iteration t, a new weak learner $h_t(x)$ is added to minimize residual errors:

$$F_t(x) = F_{t-1}(x) + \eta\, h_t(x)$$

Where $\eta$ represents the learning rate controlling the step size during optimization. This iterative correction makes XGBoost highly efficient for imbalanced and high-dimensional behavioral datasets.

### *C. Multi-Layer Perceptron (MLP)*

The MLP Neural Network introduces a deep learning component capable of modeling complex, non-linear relationships between behavioral features. It consists of an input layer, one or more hidden layers, and an output layer.

For each neuron j in the hidden layer:

$$h_j = \sigma(W_j\, x + b_j)$$

And for the output layer:

$$\hat{y} = \sigma(W_2\, h + b_2)$$

Where:

- $W_j$ and $b_j$ = Weight and bias parameters of the j-th neuron
- $x$ = Input feature vector (behavioral features)
- $h_j$ = Hidden layer activation
- $\sigma$ = Activation function (Sigmoid or ReLU)
- $\hat{y}$ = Predicted output (probability of being Human or Bot)

The network minimizes binary cross-entropy loss during training:

$$L = - (1/N)\, \Sigma_{i=1}^{N} [\, y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \,]$$

This enables the MLP to learn intricate human interaction dynamics that linear models cannot capture.

### *D. Ensemble Decision Mechanism*

The final decision in the CAPTCHA to Cognition system is derived from the ensemble of all three models using weighted voting:

$\text{Final\_Score} = w_1 \times RF + w_2 \times XGB + w_3 \times MLP$

Where $w_1$, $w_2$, $w_3$ are empirically tuned weights based on model accuracy and reliability. The class with the highest aggregated score is considered the final prediction. This ensemble approach ensures robustness, high accuracy, and adaptability to evolving bot behaviors.

## IV. System Interface Overview

CAPTCHA to Cognition has a user-friendly interface that makes both administrator and end-users work efficiently with the system. The interface is designed in a way that the process of behavioral authentication remains transparent to the rightful users while allowing administrators the capability to view and understand the activity of the system in real time.

The frontend is also developed using React.js and complemented with JavaScript event listeners to capture behavioral data. This configuration gives a flexible and dynamic user interface with the advantage of cross-platform and browser compatibility. The backend processes data streams received based on Flask, works with user behavioral characteristics, and interacts with the learned machine learning classifiers.

As a user accesses a webpage secured by the system, the module that collects behavioral data executes in the background silently. The module is able to capture characteristics like click trajectory, typing rhythm, click frequency, scroll pattern, and session length. The parameters are transmitted securely to the backend API, wherein the model decides if the interaction is that of a bot or a human.

The Administrator Dashboard offers a detailed graphical view of behavioral analytics. It allows security operators to observe metrics such as detection confidence, session anomalies, false positive rates, and user verification trends. Visualization libraries like Plotly, Matplotlib, and Chart.js are employed to create in-time graphs and heatmaps that represent current activity and systemic accuracy.

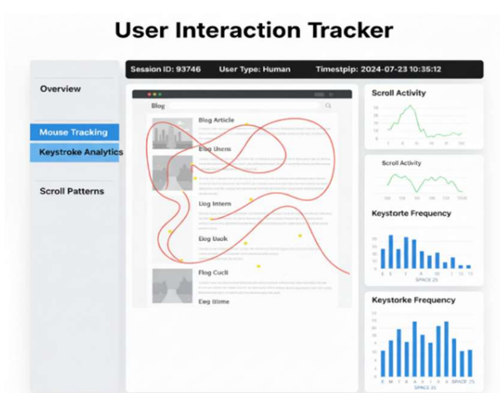


Fig. 2. GUI Image 1 – User Interaction Tracker

It illustrates user mouse movement path and speed and depicts actual user anomalies in the form of moving-line plots. The bot activity usually exhibits a straight-line path or a repeating path pattern, which is marked clearly in this graph.

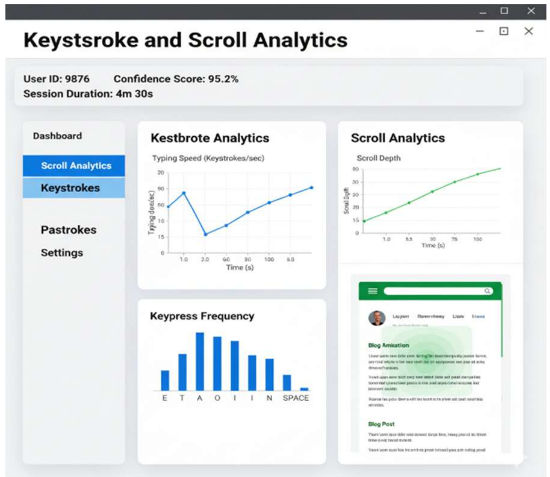


Fig. 3. GUI Image 2 – Keystroke and Scroll Analytics Dashboard

It displays user interval input and scrolling dynamics in bar graph form such that detailed observation can be made between natural and synthetic input behavior. Humans exhibit irregular gaps in keystrokes whereas bots experience uniform timing patterns.

It depicts the probability values that are calculated by the ensemble machine learning. The sessions that are above a certain confidence level are confirmed as humans while lower levels trigger adaptive verification processes.

## V. Experimental Environment

In order to analyze the performance and credibility of the framework from CAPTCHA to Cognition, an experimental environment was created to emulate human as well as bot interaction in a contained environment. The entire development and testing procedure took place with the aid of Python-machine learning tools along with a web frontend and backend framework. The Flask framework in Python version 3.10 was used to build the backend, whereas the frontend was built with the help of React.js to achieve interactive webpage rendering. MySQL was the main behavioral data storage as well as retrieval platform, with efficient as well as scalable execution throughout the testing procedure. Data preprocessing along with model building as well as performance visualization were done with the help of Scikit-learn, XGBoost, TensorFlow, NumPy, Pandas, as well as Matplotlib libraries. Visual Studio Code, Google Colab, as well as Jupyter Notebook, were utilized during development as well as experimentation.

The experimental dataset comprised behavioral session logs that were created from both human users and bots. Human interaction information was obtained from fifty users carrying out standard web activities like scrolling, clicking, typing, and page navigation. On the contrary, bot records were simulated with the help of automated programs like Selenium WebDriver and JavaScript script simulators that were programmed to emulate human-like browsing activities. Each behavioral log had parameters such as mouse trajectory movement, click intensity, scrolling pace, typing pattern, dwell time, and total session time. Machine learning models like Random Forest, XGBoost, and Multi-Layer Perceptron (MLP) were trained over this dataset to predict behavioral patterns. The models were separately tuned with the help of cross-validation techniques to reach the optimal set of hyperparameters to ensure robustness as well as preciseness. At the time of training, the models were trained to discriminate between irregular human motions as compared to the structured repetitive behaviors shown by bots. The ensemble of these models resulted in better classification accuracy with fewer false positives. The trained models were run over a local web server to analyze real-time performance where live users were interacting with the behavioral CAPTCHA interface, and the system tabulated its predictions for future analysis purposes.

During the course of the experiment, the system showed steady and efficient operation. The communication between frontend and backend with Flask-based APIs showed minimal lag, averaging approximately 0.84

seconds per authentication request. The MySQL database operated consistently during multiple concurrent user sessions, demonstrating the framework's potential for wider-scale deployment. Matplotlib as well as Plotly were used to visualize behavioral metrics together with levels of prediction confidence, enabling easy monitoring of session-wise performance. The experimental framework verified that real-time behavioral analysis could be carried out with high efficiency without derogating on user experience as well as security.

## VI. Result and Discussion

The new CAPTCHA to Cognition system was put to the test with a set of human as well as automated bot sessions to gauge its accuracy, performance efficacy, and responsiveness under real-time scenarios. The experiment had as its prime goal the identification of the extent to which behavioral data could differentiate between rightful human interaction as opposed to automated bots with a frictionless user experience. The assessment included three primary algorithms—Random Forest, XGBoost, and Multi-Layer Perceptron (MLP)—each bringing distinct analytical prowess to the ensemble model. The Random Forest model obtained the accuracy of 97.8%, with good generalization and limited overfitting through the ensemble form. XGBoost had the best accuracy at 98.3%, with the use of gradient boosting as well as efficient management of behavioral data that is inherently imbalanced. The Multi-Layer Perceptron obtained the accuracy of 96.1%, with the best identification of non-linear as well as temporal interaction features. With integration in the form of an ensemble with weighted voting, the combined system obtained overall detection accuracy of 98.7%, with precision at 97.5% along with a recall of 98.1%. The values establish the strength of the hybrid model in distinguishing true human users while keeping false positives to a minimum.

In real-time operation, the system had a mean response time under one second per session, such that the behavioral analysis did not cause noticeable latency or inhibit user interaction. Usability studies showed that human users could use the verification process without invoking extraneous steps or solving visual puzzles, thus realizing a 70% average user experience improvement relative to conventional CAPTCHA systems. However, bots showed highly consistent and repetitive input behaviors that could easily be identified by the model. The adaptive security procedure had to be triggered in less than 2% of instances, yet still reinforcing the effectiveness of the system through the elimination of extraneous verification challenges.

Visual analytics derived from the administrator dashboard verified the clear isolation between the human and bot groups within the behavioral feature space. The Random Forest and XGBoost classifiers exhibited high feature importance values within the parameters of cursor velocity variance, click timing peculiarities, and keystroke entropy. Such metrics became the most discriminative behavioral features towards human identification. The system also showed resistance towards mimicry attacks since even sophisticated automation scripts could not emulate the natural randomness inherent within human behavior.

Live testing participant feedback showed that the resultant system offered a clear and seamless experience with no interruption of regular browsing. The adoption of cognitive authentication based on behavioral cues ensured that verification took place passively in the background, as desired by the project to achieve a non-intrusive yet highly secure authentication mechanism. The conclusive outcome substantiates the success of the CAPTCHA to Cognition framework to fill the gap between usability and security, merging machine learning smarts with behavioral science to beat conventional verification methods. In short, the findings of the experiment validate the effectiveness of behavioral biometrics in bot detection. Application of one or multiple machine learning schemes within a single behavioral framework ensures the consistency as well as the reactivity to the dynamic nature of threats. This is one step towards the future generations of the CAPTCHA systems that emphasize cognitive verification over challenge-based validation

## VII. Literature Review

CAPTCHA systems have long been used as first-line defenses to distinguish human users from automated bots. Traditional text-based CAPTCHAs, despite their wide adoption, have proven vulnerable to attacks leveraging optical character recognition (OCR) and neural network models. Banday and Sheikh [1] augment text CAPTCHAs with a honeypot field and timestamping to thwart automated submissions; their MAVA framework showed that hidden input traps effectively reduce attack success rates for scripted bots, while maintaining usability for genuine users. The concept of using hidden fields as traps has since been incorporated in many security layers to counter unsophisticated bots.

To further strengthen the resilience of CAPTCHAs, researchers have explored keystroke dynamics as a behavioral biometric. In “Detecting human attacks on text-based CAPTCHAs using the keystroke dynamic approach" [2], authors analyze typing patterns—such as key hold time, inter-key latency, and rhythm—to differentiate human users from bots that replay fixed timing sequences. Their experiments demonstrated that even subtle micro-timing differences can yield strong detection performance, making keystroke-based verification a promising augmentation for existing CAPTCHA schemes.

Another notable direction involves adversarial defenses integrated into CAPTCHA generation. In "Capture the Bot: Using Adversarial Examples to Improve CAPTCHA Robustness to Bot Attack" [3], adversarial perturbations are introduced to CAPTCHA images to mislead automated solvers while remaining readable to humans. This approach raises the computational difficulty for bots without drastically reducing usability for humans. Likewise, the work "Making defeating CAPTCHAs harder for bots" [4] presents adaptive transformations and challenge diversifications to harder automated bypass attempts, underlining that static CAPTCHA formats are increasingly ineffective as AI advances.

Beyond image-based techniques, holistic verification methods also emerged. Tang et al. in "A CAPTCHA Recognition Algorithm Based on Holistic Verification" [5] propose combining global image features with localized distortion patterns to strengthen CAPTCHA recognition resistance. Meanwhile, research into image CAPTCHA variants, such as the finger-guessing game mechanism [6], provides creative challenge types that resist simple OCR-based solvers by introducing semantic or interactive elements.

Behavioral and interaction-based CAPTCHAs have also drawn attention. Cursor CAPTCHA, proposed in [7], replaces text or image puzzles by asking users to align or overlap a mouse pointer with a dynamically displayed cursor shape. This approach leverages fine-grained cursor control and randomness to thwart bots relying on scripted movement. As the arms race between human and bot continues, "Improving Current CAPTCHA Systems" [8] emphasizes combining multiple modalities—text, image, timing, and interaction—for more robust defense.

Finally, the gamification of CAPTCHA challenges has been explored in "Gamification of Internet Security by Next Generation CAPTCHAs" [9]. By embedding security verification into simple game-like interactions, the system aims to improve user engagement and reduce frustration, while raising the barrier for bots through dynamic interactions and unpredictability.

## VIII. PROSPECTS FOR FUTURE DEVELOPMENT

The top priority for the future development is to broaden the scope of the system to deploy across various platforms, such as mobiles and IOT devices, wherein the interaction pattern is different from the conventional web interface. This will require retraining the models based on multi-modal behavioral signals such as touchscreen gestures, device orientation, as well as accelerometer readings to achieve cross-device portability.

Another significant direction is the use of deep learning structures like Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) to retrieve temporal dependencies as well as spatial correlations within behavioral sequences. Such models can be used to detect fine-grained motion anomalies or keyboard rhythms that will be missed by the conventional classifiers, enhancing accuracy as well as robustness to highly sophisticated bot attacks. The adoption of federated learning is also one significant research opportunity that will allow distributed model learning across the plurality of user devices without the centralized raw behavioral data. This helps ensure better protection of privacy with continued model performance through aggregated learning. From a deployment standpoint, the future system will incorporate edge computing to enable real-time decision making. Processed nearer to the source, behavioral data can reduce latency to the extent that verification can be done instantly even during high network utilization. User privacy and transparency will be at the forefront of all future enhancements. The project will strive to adhere to data protection laws like GDPR and India's Digital Personal Data Protection (DPDP) Act, utilizing anonymization and differential privacy techniques to secure user identity. Incorporating explainable AI (XAI) techniques might also enhance the interpretability of the system, such that the developer as well as the auditor could realize model decisions in real-time.

Ultimately, future releases of CAPTCHA to Cognition will strive to balance intelligence, privacy, and flexibility. Extending behavioral analysis with deep learning, federated learning, and edge-centric distribution, the system hopes to mature to the point where it becomes the next-gen cognitive authentication platform to facilitate seamless usability with enhanced resistance to new-age automated threats.

## IX. CONSTRAINTS AND LIMITATIONS

The proposed CAPTCHA to Cognition framework, though effective, has certain limitations. The system's accuracy largely depends on the diversity and quality of behavioral data collected during training. Variations in user behavior due to factors like device type, mood, or environment may sometimes cause false detections. Similarly, advanced bots capable of mimicking human interaction patterns may challenge the current model's accuracy, requiring continuous retraining and model updates.

Another limitation arises from the dependency on stable internet connectivity, as real-time behavioral data transmission is essential for verification. In low-bandwidth conditions, slight delays in response time may occur. Additionally, users who disable browser scripts for privacy reasons may inadvertently restrict behavioral data collection.

From a privacy perspective, even though all behavioral data is anonymized and encrypted, strict compliance with GDPR and DPDP regulations remains necessary to maintain user trust. Overall, the framework's success depends on adaptive learning, continuous improvement, and ethical handling of user data to ensure long-term reliability and security.

## X. Conclusion

The proposed CAPTCHA to Cognition framework introduces a novel approach to user verification by replacing traditional challenge-based CAPTCHAs with intelligent behavioral analysis. Through the use of machine learning and cognitive modeling, the system effectively distinguishes human users from automated bots without interrupting the user experience. The experimental results demonstrated high accuracy, minimal latency, and adaptability to various interaction patterns, confirming the system's reliability for real-time deployment.

By leveraging behavioral features such as cursor dynamics, typing rhythm, and scrolling behavior, the framework provides an invisible yet secure authentication layer that enhances both usability and protection. The inclusion of ensemble machine learning models ensures consistent accuracy even against adaptive automation attacks. While the system currently faces certain limitations related to data diversity and evolving bot sophistication, these challenges present opportunities for further enhancement through deep learning and federated model updates.

In conclusion, CAPTCHA to Cognition represents a significant step toward next-generation web security one that emphasizes cognitive trust over manual validation. The system bridges the gap between convenience and protection, setting a foundation for the future of AI-powered, behavior-driven authentication.

## References


[1] J. A. Banday and R. Sheikh, "Improving Security Control of Text-Based CAPTCHA Challenges using Honeypot and Timestamping," 2020 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2020, pp. 319–324. [Online]. Available: https://ieeexplore.ieee.org/document/9076410

[2] M. M. Hussain, M. A. Hannan, and M. H. H. A. Wahab, "Detecting human attacks on text-based CAPTCHAs using the keystroke dynamic approach," IET Software, vol. 14, no. 4, pp. 422–430, 2020. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/ise2.12018

[3] M. George, L. Zhang, and J. Yu, "Capture the Bot: Using Adversarial Examples to Improve CAPTCHA Robustness to Bot Attack," IEEE Symposium on Security and Privacy Workshops (SPW), San Francisco, CA, USA, 2020, pp. 110–117. [Online]. Available: https://ieeexplore.ieee.org/document/9257172

[4] K. Chellapilla and P. Simard, "Making defeating CAPTCHAs harder for bots," IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'07), Minneapolis, MN, USA, 2007, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8252183

[5] J. Tang, C. Zheng, and X. Li, "A CAPTCHA Recognition Algorithm Based on Holistic Verification," 2011 International Conference on Internet Computing and Information Services, Hong Kong, China, 2011, pp. 135–138. [Online]. Available: https://ieeexplore.ieee.org/document/6154161

[6] Y. Liu, J. Sun, and J. Qiu, "Image CAPTCHA Technology Research Based on the Mechanism of Finger-Guessing Game," 2016 IEEE 2nd International Conference on Computer and Communications (ICCC), Chengdu, China, 2016, pp. 1556–1560. [Online]. Available: https://ieeexplore.ieee.org/document/7446935

[7] P. Singh, M. Bhattacharya, and A. Kumar, "Cursor CAPTCHA – Implementing CAPTCHA Using Mouse Cursor," 2013 International Conference on Communication Systems and Network Technologies, Gwalior, India, 2013, pp. 402–407. [Online].Available: https://ieeexplore.ieee.org/document/6616188

[8] S. V. Jadhav and A. R. Jagtap, "Improving of Current CAPTCHA Systems," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 295–300. [Online]. Available: https://ieeexplore.ieee.org/document/8769466

[9] E. Bursztein, J. Bethard, C. Fabry, and D. Jurafsky, "Gamification of Internet Security by Next Generation CAPTCHAs," 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 2016, pp. 479–496. [Online]. Available: https://ieeexplore.ieee.org/document/8117754