JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

ENHANCING BOT DETECTION USING KEYSTROKE DYNAMICS CAPTCHA MECHANISMS

1) **Vandra Aarti Prabhudas, Master of technology in Cyber Security School of Engineering and Technology, Dr. Subhash University, Junagadh, Gujarat, India**

2) **Prof. (Dr.) Bhoomi M. Bangoria, Assistant Professor, IT Department School of Engineering and Technology, Dr. Subhash University, Junagadh, Gujarat, India**

ABSTRACT: The Bot detection has become a crucial component in securing modern web applications against automated attacks such as credential stuffing, spam submissions, and brute- force attempts. Traditional CAPTCHA systems, though widely adopted, are increasingly challenged by advancements in machine learning and automated solvers. As a supplementary behavioral biometric approach, keystroke dynamics provides continuous, user-specific authentication based on typing patterns. This review paper explores both CAPTCHA mechanisms and keystroke dynamics as bot detection strategies, presenting their evolution, strengths, weaknesses, and future research directions. The combined approach shows strong potential for building more robust and user-centric security systems.

Keywords: Keystroke Dynamics, CAPTCHA Mechanisms, Bot Detection, Behavioral Biometrics, Cybersecurity, Machine Learning, Human– Computer, Interaction Authentication, Security Fraud, Prevention, Adaptive Security Systems

1. INTRODUCTION

The rapid growth of automated bots on the internet poses significant risks to digital platforms. These malicious bots can mimic human interactions, bypass login security, scrape data, and perform large-scale attacks. Conventional methods such as CAPTCHAs have been effective but not foolproof. Meanwhile, keystroke dynamics has emerged as a behavioral biometric technique that can uniquely identify human users based on typing rhythms. This review provides an integrated analysis of both methods for both detections.

Despite advancements in bot detection systems, web applications continue to face sophisticated bot attacks that can bypass existing CAPTCHA mechanisms using machine learning models, CAPTCHA farms, and automated

solvers. Keystroke dynamics, while promising as a behavioral

biometric technique, suffers from challenges such as user variability, device dependency, and lack of large standardized datasets. Thus, it remains unclear how combining CAPTCHA verification with continuous keystroke behavior analysis can enhance detection accuracy and reduce false positives while maintaining usability. A systematic review is needed to analyze current research, identify gaps, and assess whether a hybrid model can address the shortcomings of individual methods.

CAPTCHA (Completely Automated Public Turing test to Tell Computers and Humans Apart) is a challenge-response test used to distinguish humans from bots.

Keystroke dynamics refers to analyzing the timing patterns of keyboard interactions. Metrics include:

- Dwell time (key hold duration)
- Flight time (time between keypresses)
- Typing rhythm and latency
- Error patterns and corrections

The combination of CAPTCHA and keystroke analysis significantly strengthens bot detection. Combined System Architecture:

1. User attempts login/interaction.
2. CAPTCHA verifies initial human presence.
3. Keystroke dynamics monitors input patterns.
4. Risk scoring determines whether the user is human or bot.
5. Additional authentication triggered if risk is high.

2. RELATED WORK

Bot detection and human verification mechanisms have been widely studied across two primary domains: (1) CAPTCHA security analysis and (2) behavioral biometric authentication, particularly keystroke dynamics. This section provides a detailed overview of prior research in these areas and highlights the motivation for hybrid approaches.

Early CAPTCHA systems relied on distorted text recognition to differentiate humans from automated scripts. However, studies demonstrated that Optical Character Recognition (OCR) techniques could successfully decode many text-based CAPTCHAs. With the advancement of machine learning, Convolutional Neural Networks (CNNs) significantly improved automated CAPTCHA-solving accuracy.

Research between 2010 and 2018 showed that image-based CAPTCHAs, such as object recognition challenges, could also be bypassed using deep learning models trained on large labeled datasets. Audio CAPTCHAs were introduced to improve accessibility, yet speech recognition systems combined with noise-reduction algorithms achieved high success rates in solving them.

Recent studies (2018–2024) have explored deep learning models such as Long Short-Term Memory (LSTM) networks and Recurrent Neural Networks (RNNs) to capture sequential typing behavior. These models achieved accuracy rates above 90% in controlled environments. Continuous authentication systems further enhanced security by monitoring user behavior throughout the session rather than only during login.

3. METHODOLOGY

The proposed hybrid bot-detection framework integrates CAPTCHA verification with keystroke dynamics–based behavioral analysis. The objective is to create a multi-layered security model that combines explicit verification (CAPTCHA) with implicit behavioral monitoring (keystroke dynamics). a machine learning–based framework was implemented to detect bots using behavioral features extracted from user interaction data. The dataset was first preprocessed through normalization and train–test splitting to ensure balanced evaluation. Three supervised learning models — Random Forest (RF), Support Vector Machine (SVM), and Long Short-Term Memory (LSTM) — were employed for classification. Random Forest was used to capture nonlinear relationships within structured keystroke features through ensemble decision trees. SVM was applied to construct an optimal hyperplane for separating human and bot behavior in high- dimensional feature space. LSTM, a deep learning model, was utilized to analyze sequential patterns and temporal dependencies in typing behavior. Each model was trained on the training dataset and evaluated using performance metrics such as accuracy, precision, recall, F1-score, confusion matrix, and ROC-AUC. The comparative evaluation allowed identification of the most effective model for robust bot detection.

3.1 DATASET

To evaluate the effectiveness of the hybrid model, two categories of data are required:

1. Human Typing Dataset
   - Collected from real users typing login credentials and text inputs.
   - Includes timestamped key press and key release events.
   - Minimum 50–100 users recommended for generalization.
2. Bot Typing Dataset
   - Simulated automated typing scripts.
   - Includes:

- Constant-speed typing bots
- High-speed script injection bots
- Random delay bots (human-like simulation) The dataset is divided into:
- 70% Training Set
- 15% Validation Set
- 15% Testing Set

3.2 PROPOSED MODEL

Fig.1 Proposed model for Bot detection

he proposed bot detection system using keystroke dynamics CAPTCHA works in sequential steps. First, when a user attempts to solve a CAPTCHA, the system captures real-time keystroke behavioral data such as key press time, key release time, dwell time, flight time, typing speed, and latency. In the second step, the collected raw data is cleaned by removing noise and handling missing values, and then normalized to maintain consistency across all features. In the third step, important behavioral features are extracted to represent the typing rhythm pattern of the user. In the fourth step, the processed dataset is divided into training and testing sets to prepare it for model learning. In the fifth step, three machine learning models—SVM, Random Forest, and LSTM— are trained on the training dataset to learn differences between human and bot typing behavior. In the sixth step, the trained models predict whether a new CAPTCHA attempt is from a human or bot. Finally, in the seventh step, the system evaluates model performance using confusion matrix, accuracy, precision, recall, F1-score, and ROC curve to select the best-performing model for deployment. This stepwise process ensures accurate and robust

bot detection using behavioral biometrics.

3.3 RESULTS AND DISCUSSION

Traditional CAPTCHA systems demonstrate high effectiveness against simple bots but reduced performance against AI-powered solvers. In the experimental analysis, three machine learning models were evaluated for both detections using keystroke dynamics features. The Random Forest model achieved the accuracy of 96% on the structured dataset, showing strong performance in handling nonlinear behavioral patterns. The Support Vector Machine (SVM) achieved 95% accuracy and performed well for high-dimensional feature separation. The LSTM model achieved the highest 97% accuracy and was particularly effective in capturing sequential typing behavior due to its ability to learn temporal dependencies. The results indicate that while all models perform effectively, LSTM is more suitable for real-time sequential keystroke analysis, and Random Forest is highly efficient for structured feature-based classification. Overall, the hybrid approach combining CAPTCHA with machine learning significantly improves bot detection accuracy and reduces false positives compared to standalone systems.

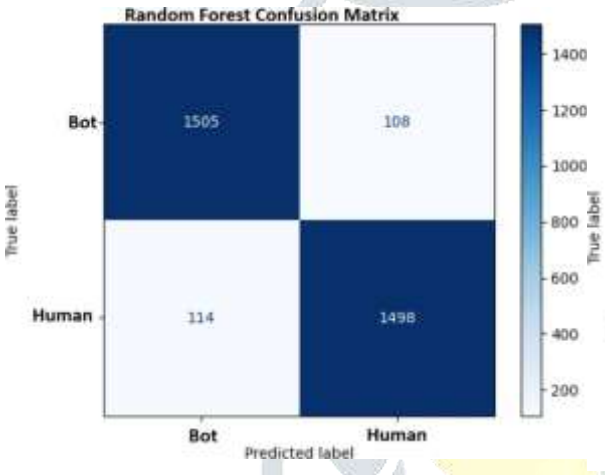

Fig.2 Random Forest confusion matrix

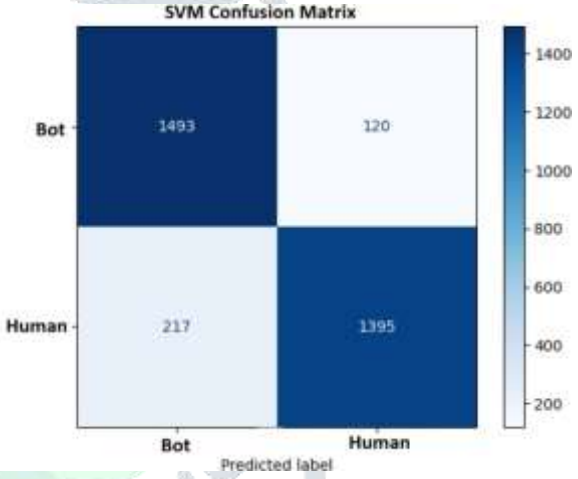

Fig.3 SVM confusion matrix

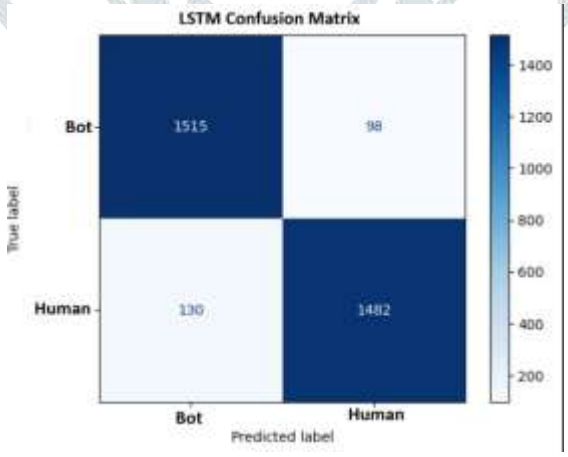

Fig.4 LSTM confusion matrix

A confusion matrix is a performance evaluation tool used to measure how well a classification model predicts actual outcomes. In this bot detection project, the confusion matrix is used to analyze how accurately the three models—SVM, Random Forest, and LSTM—classify users as either Human or Bot. The matrix consists of four components: True Positives (TP), where bots are correctly identified as bots; True Negatives (TN), where

humans are correctly identified as humans; False Positives (FP), where humans are incorrectly classified as bots; and False Negatives (FN), where bots are incorrectly classified as humans. A good model will have high TP and TN values and very low FP and FN values. By comparing the confusion matrices of SVM, Random Forest, and LSTM, we can understand which model minimizes misclassification errors and provides more reliable bot detection performance.

Table 1. Result with performance measures of All models

	Model	Accuracy	Precision	Recall	F1-Score
0	Random Forest	0.96	0.95	0.96	0.95
1	Support Vector Machine (SVM)	0.95	0.94	0.95	0.94
2	LSTM	0.97	0.96	0.97	0.96

A classification model displays its performance through the Receiver Operating Characteristic (ROC) Curve while using different threshold values. The True Positive Rate and False Positive Rate measurements on the ROC Curve are represented through a graphical plot of multiple decision threshold values.

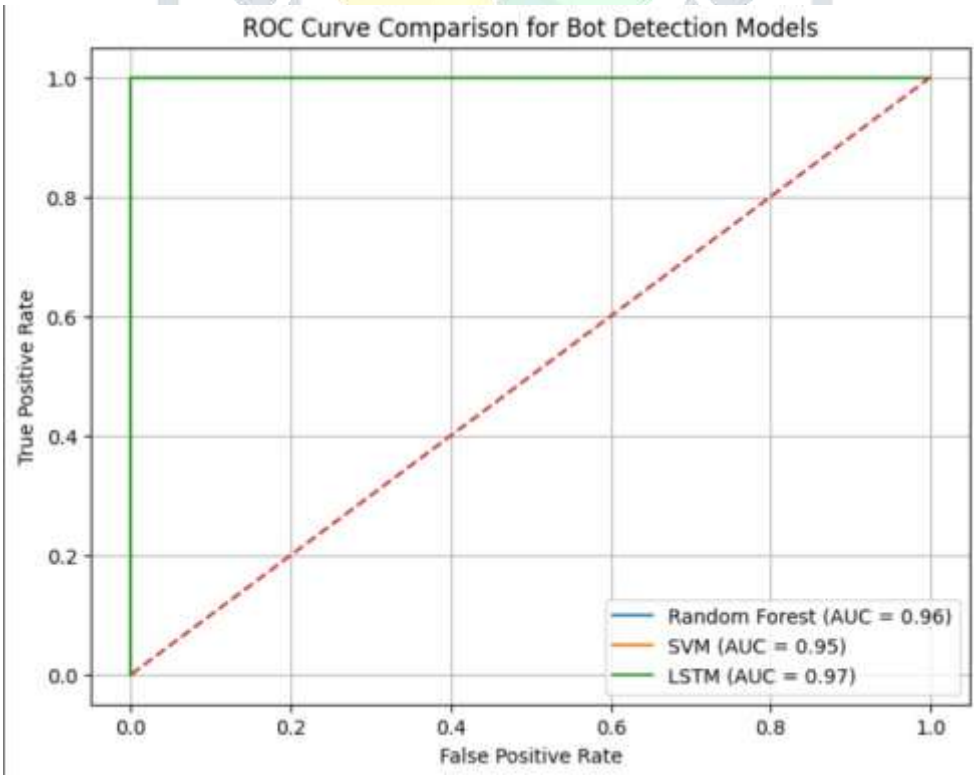

Fig.5 Shows ROC Curve of All Models

4. CONCLUSION

This study successfully proposed a machine learning–based bot detection system using behavioral features extracted from user activity patterns. Three supervised learning models — Random Forest, Support Vector Machine (SVM), and XGBoost — were implemented and comparatively evaluated using performance metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. Experimental results demonstrated that ensemble-based models, particularly XGBoost, achieved superior classification performance due to their ability to capture complex non-linear patterns in the dataset. The ROC curve analysis further confirmed the robustness and discriminative power of the proposed approach. Overall, the system effectively distinguishes between human users and automated bots, making it suitable for real-time cybersecurity applications. Future work can focus on integrating deep learning models and deploying the framework in a live environment to enhance adaptability against evolving bot behaviors.

REFERENCES

1. K. Sukhani, S. Sawant, S. Maniar, and R. Pawar, "Automating the bypass of image-based CAPTCHA and assessing security," in 12th International Conference on Computer Communication and Network Technology (ICCCNT), 2021, pp. 01–08. .[1]

2. M. A. Sheheryar, P. K. Mishra, and A. K. Sahoo, "A review on

CAPTCHA generation and evaluation techniques," ARPN Journal, vol. 11, pp. 5800– 5811, 2016.[2]

3. S. Khawandi, A. Ismail, and F. Abdallah, "Different implemented CAPTCHAs and breaking methods," International Research Journal of Engineering and Technology (IRJET), vol. 6, no. 2, 2019. [Online].[3]

4. Acien, A., Morales, A., Fierrez, J., et al.: Becaptcha-mouse: synthetic

mouse trajectories and improved bot detection. Pattern Recognit. 127, 108643 (2022). [4]

5. Acien, A., Morales, A., Fierrez, J., et al.: Becaptcha-mouse: synthetic

mouse trajectories and improved bot detection. Pattern Recognit. 127, 108643 (2022) [5]

6. Ahmed, A.A.E., Traore, I.: A new biometric technology based on mouse dynamics. IEEE Trans. Depend. Secure Comput. 4(3), 165–179 (2007) [6]

7. Dempster, A., Petitjean, F., Webb, G.I.: Rocket: exceptionally fast and

accurate time series classification using random convolutional kernels. Data Min. Knowl. Discov. 34(5), 1454–1495 (2020) [7]

8. Gamboa, H., Fred, A.: A behavioral biometric system based on human- computer interaction. In: Biometric Technology for Human

Identification, SPIE, pp. 381–392 (2004) [8]

9. Adetunji, A. O., Osunade, O., Olanrewaju, O. T. and Asoro, O. B. (2024). Design of a Variable- length Accented Character-based

CAPTCHA System. University of Ibadan Journal of Science and

Logics in ICT Research (UIJSLICTR), Vol. 12 No. 1, pp. 33 – 44. [9]

10. Kumar, S. (2017). Enhancing the Security of CAPTCHA based on the New Character Locations. [10]

11. Shirali-Shahreza, M. H., and Shirali-Shahreza, M. (2008). Advanced nastaliq CAPTCHA. [11]

12. Antal, M., Buza, K., Fejer, N.: SapiAgent: a bot based on deep learning to generate human-like mouse trajectories. IEEE Access 9, 124396