International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 05 | May 2026 www.irjet.net p-ISSN: 2395-0072

Continuous User Authentication Using AI-Driven Keystroke Dynamics

Kishor Kumar M P[1], Charan H B[2], Harshavardhan N[3], Lisha C R[4]

[1234]*Student, Dept. of Computer Science & Engineering, SSIT, Tumakuru, Karnataka, India*
Guide: Sindhu T N, Asst. Professor, Dept. of CSE, SSIT, Tumakuru

---

Abstract - *Traditional online banking systems authenticate users only at login using static credentials. Once a session begins, there is no mechanism to verify the authenticated user continues to operate the account, leaving it vulnerable to session hijacking. This paper presents a Continuous User Authentication system based on AI-driven keystroke dynamics. The system captures Hold Time (Dwell Time) and Flight Time through a JavaScript event listener embedded in the banking application. These features are sent to a Python Flask backend where a One-Class Support Vector Machine (OC-SVM) compares the typing pattern against the enrolled user's profile. Any significant deviation triggers immediate session termination. The system operates in the backgro und without specialized hardware, providing a non-intrusive security layer for banking operations.*

Key Words **- Keystroke Dynamics, Behavioral Biometrics, Continuous Authentication, One-Class SVM, Anomaly Detection, Banking Security, Machine Learning**

1. INTRODUCTION

The rapid growth of internet banking has exposed users to evolving cybersecurity threats. Conventional username-password authentication protects accounts only at login. Once a session is established, the system implicitly trusts the authenticated entity, creating a critical vulnerability window. An attacker who obtains valid credentials can exploit an ongoing session with no challenge mechanism in place. Behavioral biometrics addresses this gap by continuously verifying identity based on observable interaction patterns. Keystroke dynamics is particularly practical it requires no additional hardware, works with any standard keyboard, and operates passively without disrupting the user experience. This paper proposes a Continuous User Authentication framework integrated into a web-based banking application using keystroke dynamics and One-Class SVM.

2. PROBLEM STATEMENT

Modern online banking platforms rely on static, one-time authentication at login. Once authenticated, the session is fully trusted for its entire duration. This is fundamentally flawed as credential theft through phishing and session hijacking attacks allow adversaries to impersonate authenticated users. Unauthorized fund transfers and sensitive data exposure can occur within seconds of a compromised session. There is a pressing need for a mechanism that continuously monitors session authenticity and responds to anomalies in real time.

3. OBJECTIVES

- Develop a continuous authentication framework that monitors the user throughout their banking session.
- Capture keystroke timing data using a JavaScript-based event listener with high temporal resolution.
- Train a personalized One-Class SVM model on the enrolled user's typing behavior.
- Trigger immediate security responses when typing patterns deviate from the established baseline.
- Ensure the system operates without specialized hardware or disruption to the user experience.

4. LITERATURE SURVEY

A growing body of research supports the viability of keystroke dynamics as a continuous authentication mechanism. The following table presents key works that informed the design of the proposed system. Table 1: Summary of Related Works Author(s) & Year Focus Area Methodology Sun et al. (2022) Key Findings Keystroke biometrics for user authentication LSTM/RNN-based deep learning Deep learning outperformed traditional methods; high computational complexity limits real-time deployment. Kumar & Bansal (2023) Killourhy & Maxion (2009) Web-based continuous authentication for banking Benchmarking anomaly detection for keystroke dynamics SVM on keystroke timing features Comparative analysis of 14 algorithms Unauthorized session access successfully detected; validates SVM for banking continuous authentication. Established CMU benchmark dataset; foundational reference for algorithm selection. The surveyed literature demonstrates that classical machine learning approaches, particularly SVM-based classifiers, strike a better balance between performance and efficiency in browser-server environments compared to deep learning methods. The OC-SVM variant adopted in this project

removes the requirement for imposter data during training a practical necessity when only genuine user samples are available at enrollment.

5. METHODOLOGY

The proposed system is implemented as a full-stack web banking application with an integrated machine learning-powered biometric layer. 5.1 System Modules Module 1 Data Acquisition: A JavaScript event listener captures key down and key up timestamps using the performance. Now () API. Each keystroke event is stored with its key identifier and timestamps, then serialized to JSON and sent to the backend via an asynchronous POST request. Module 2 Feature Extraction: The Flask server derives two biometric features from the raw keystroke log. Hold Time (Dwell Time) is the duration a key is held down. Flight Time is the interval between consecutive keystrokes. These features are assembled into a feature vector and normalized using Standard Scalar. Module 3 Classification: The normalized feature vector is evaluated by a pre-trained One-Class SVM model. A result of +1 indicates genuine user typing; -1 identifies an anomaly and triggers session termination. 5.2 Algorithm: One-Class SVM One-Class SVM is specifically designed for scenarios where only positive-class (genuine user) data is available for training. It learns a compact decision boundary around enrollment data and treats any point outside that boundary as an anomaly. The RBF kernel accommodates natural variability in human typing, and the model trains effectively on as few as 20 enrollment samples, making it practical for real-world deployment.

6. RESULTS AND DISCUSSION

The prototype was evaluated with five users, each completing enrollment by typing a reference sentence 20 times. The OC-SVM models were tested against genuine and simulated impostor inputs from other enrolled users. Genuine users passed authentication consistently across normal typing conditions with no interruption to their banking transactions. When impostor inputs were introduced, the system correctly blocked transactions and triggered re-authentication. The false rejection rate remained low, indicating the model was well-calibrated to accommodate natural within-user typing variability. The biometric check completed server-side in under 50 milliseconds on average, confirming viability for real-time deployment.

7. CONCLUSIONS

This paper presented a Continuous User Authentication system for web-based banking applications using AI-driven keystroke dynamics. By moving beyond static login verification to persistent behavioral biometric checking, the system addresses a critical gap in conventional banking security. The One-Class SVM model requires no imposter data for training, integrates seamlessly into existing web infrastructure, and imposes no perceptible latency on end users. Future work will focus on expanding to free-text analysis, incorporating additional behavioral signals such as mouse dynamics, and evaluating performance across larger user populations.

REFERENCES

[1] Sun, L., et al., "Keystroke Biometrics for User Authentication using Deep Learning," IEEE Access, vol. 10, pp. 123-134, 2022.

[2] Kumar, R. & Bansal, S., "Web-based Continuous Authentication for Banking Systems using SVM," International Journal of Information Security, 22(4), pp. 45-58, 2023.

[3] Killourhy, K. S. & Maxion, R. A., "Comparing Anomaly-Detection Algorithms for Keystroke Dynamics," IEEE/IFIP DSN, 2009.

[4] Pedregosa, F., et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[5] Bours, P., "Continuous Keystroke Dynamics: A Different Perspective," IEEE Security & Privacy, 2012.

[6] MongoDB Inc., "MongoDB Documentation for Python Developers," 2024. [Online]. Available: https://www.mongodb.com/docs/

Project Team Lead

Kishor Kumar M P
8th Semester, CSE Branch
SSIT, Tumakuru - 572105
Karnataka, India.