

Continuous User Authentication Using Behavioral Biometrics for Enhanced Web Application Security

Bhagya Lekshmi B¹, Shana Yasmin², Fadwa C³, and Prof. Shruthi Susan Moncy⁴

Department of Computer Science and Engineering,
Mar Athanasius College of Engineering (Autonomous), Kothamangalam, India

Abstract—Ensuring application security beyond initial login has become essential, leading to the development of continuous authentication systems that verify users through their behavioral patterns. The system addresses major security issues such as unauthorized access after login, credential misuse, and lack of real-time identity verification in traditional authentication mechanisms. After successful login, the system silently monitors user behavior including typing speed, key press timing, and mouse movement patterns while the user interacts with the application. A behavioral profile is created for each user during normal usage, and a machine learning-based anomaly detection model compares real-time behavior with the stored profile to identify suspicious activity. When abnormal behavior is detected, the system responds by triggering re-authentication, restricting access to sensitive sections, or terminating the session. The solution is implemented as a web-based prototype using Python and FastAPI for backend development, JavaScript for capturing user behavior, HTML and CSS for frontend design, and the Isolation Forest algorithm for anomaly detection. System performance is evaluated using metrics such as detection accuracy and false positive rate, demonstrating the effectiveness of continuous authentication. This project presents a practical approach to improving security for applications such as online portals, enterprise systems, and educational platforms.

Index Terms—Continuous Authentication, Behavioral Biometrics, Anomaly Detection, Isolation Forest, User Behavior Analysis, Web Application Security, Machine Learning, Real-Time Monitoring, Keystroke Dynamics, Mouse Movement Analysis.

I. INTRODUCTION

Authentication plays a critical role in securing digital systems. Conventional authentication methods, such as passwords and one-time verification mechanisms, validate user identity only during the login phase. Once access is granted, the system assumes the user remains legitimate for the entire session. This assumption introduces significant security risks, particularly in scenarios involving session hijacking, insider threats, or credential compromise.

To overcome these limitations, continuous authentication has emerged as a promising approach. Instead of relying solely on static credentials, continuous authentication systems monitor user behavior dynamically and verify identity throughout the session. Behavioral biometrics, such as typing patterns and mouse dynamics, provide a non-intrusive and continuous source of authentication signals.

Recent advancements in machine learning have enabled real-time analysis of behavioral data for anomaly detection.

These techniques allow systems to learn normal user behavior and identify deviations that may indicate malicious activity.

This paper proposes a continuous behavioral authentication system that integrates real-time monitoring, anomaly detection, and adaptive model training. The system evaluates user behavior at regular intervals and assigns risk levels based on deviation from learned patterns. A risk-aware response mechanism ensures appropriate actions, such as re-authentication or session termination, are taken.

The main contributions of this work include:

- A real-time behavioral monitoring framework for continuous authentication
- Use of Isolation Forest for efficient anomaly detection
- Risk-based adaptive authentication with OTP verification
- A session-based feature storage and retraining strategy
- A scalable backend architecture for real-time deployment

II. RELATED WORK

Continuous authentication has gained significant attention as a solution to overcome the limitations of traditional one-time authentication mechanisms. Unlike static authentication, continuous authentication verifies user identity throughout the session using behavioral biometrics. Mondal and Bours [1] proposed a continuous authentication system based on mouse dynamics, where user behavior is monitored for every individual mouse event. Recent advancements in zero trust security models emphasize the need for continuous verification of users and devices. In this context, AI-driven anomaly detection techniques have been widely explored to enhance security in dynamic environments [2]. Furthermore, research on anomaly detection in zero trust environments highlights the importance of adaptive and intelligent security mechanisms [3].

Furthermore, behavioral biometrics have been explored as a robust mechanism for strengthening authentication systems. Recent studies highlight the effectiveness of behavioral traits such as keystroke dynamics, mouse movements, and interaction patterns in uniquely identifying users [4]. In addition, AI-driven security frameworks have been proposed to enhance zero trust architectures through intelligent decision-making [5]. Moreover, recent research emphasizes the role of anomaly detection techniques in identifying insider threats and sophisticated attacks within secure systems [6]. Recent research has also explored the use of sensor-based behavioral biometrics

for continuous authentication in smart environments. In this context, the authors of [7] proposed a system that leverages data from multiple sensors to capture user behavior patterns for authentication purposes. Behavioral biometrics-based continuous authentication systems utilize user interaction patterns such as keystrokes and mouse movements to enable real-time identity verification and enhanced security beyond traditional login mechanisms[8]. Continuous authentication frameworks integrate machine learning with behavioral biometrics to analyze user activity patterns and detect anomalies dynamically during active sessions. Such systems enable risk-based adaptive security mechanisms, enhancing cybersecurity while ensuring seamless and non-intrusive user experiences[9]. Zero-Trust Network Access (ZTNA) frameworks implement a “never trust, always verify” model by continuously authenticating users and devices before granting access to network resources[10].

III. PROPOSED SYSTEM ARCHITECTURE

The proposed behavioral authentication system is designed as a modular intelligent framework that enables continuous user verification using interaction-based behavioral biometrics. Unlike traditional authentication mechanisms, this system provides passive, real-time identity validation without interrupting the user. Modern digital systems face challenges such as account hijacking, session theft, and impersonation, which cannot be effectively addressed by static authentication alone. To overcome these issues, the system decomposes authentication intelligence into distinct yet coordinated subsystems.

The architecture consists of three primary components:

- Continuous Behavioral Monitoring and Feature Extraction
- Adaptive Model Training and Risk Evaluation
- Security Response and Session Management

Each subsystem performs specialized processing while exchanging structured behavioral insights through a centralized decision mechanism. This modular design enhances scalability, ensures robustness, and allows independent optimization of monitoring, learning, and security modules. The overall workflow operates in two distinct phases: in-session monitoring and session-end learning.

A. Continuous Behavioral Monitoring and Feature Extraction

The monitoring subsystem captures and processes user interaction data in real time to build a behavioral profile.

User activity is continuously recorded in the form of keystroke dynamics, mouse movements, scrolling behavior, and idle patterns. These raw interaction events are collected at fixed time intervals and transmitted to the backend for processing.

Behavioral Data Acquisition: The frontend captures user interactions every fixed interval and sends structured behavioral snapshots. These include keyboard events, mouse trajectories, scroll actions, and session summaries.

Feature Extraction: From each snapshot, key behavioral features such as typing speed, keystroke intervals, mouse

velocity, movement variance, scroll frequency, and idle ratio are computed dynamically. These features represent the unique interaction patterns of the user.

Transient Processing: Extracted features are processed in memory for immediate evaluation and are not permanently stored during active sessions. Only raw behavioral logs are maintained to preserve detailed activity history.

Session-Based Aggregation: At the end of each session, behavioral data corresponding to normal interaction patterns is aggregated to form a single representative feature set. This ensures that only reliable and consistent user behavior contributes to long-term learning.

This subsystem ensures efficient and non-intrusive data collection while maintaining high-quality behavioral representation.

B. Adaptive Model Training and Risk Evaluation

The learning subsystem is responsible for modeling user behavior and detecting anomalies through machine learning techniques.

A statistical anomaly detection model is employed to distinguish between normal and abnormal behavioral patterns. The system initially operates in a data collection phase until sufficient behavioral samples are obtained.

Initial Learning Phase: During the early sessions, the system collects behavioral features without training a model. Once a predefined number of sessions is reached, the model is trained using historical behavioral data.

Anomaly Detection: During active sessions, extracted features are evaluated using the trained model to generate an anomaly score. This score reflects the degree of deviation from the user's typical behavior.

Risk Classification: Based on the anomaly score, user activity is classified into risk levels such as Low, Medium, and High. These levels indicate normal behavior, potential inconsistency, or strong deviation respectively.

Incremental Model Update: The system periodically re-trains the model using recent behavioral data to adapt to gradual changes in user behavior. A versioning mechanism ensures consistency and efficient model management.

This subsystem enables continuous learning and dynamic adaptation, ensuring accurate and personalized authentication.

C. Security Response and Session Management

The security subsystem translates risk assessments into appropriate system actions to maintain account integrity.

Low-Risk Handling: When user behavior aligns with expected patterns, the session continues uninterrupted, ensuring a seamless user experience.

Medium-Risk Handling: For moderate deviations, the system triggers an additional verification step using one-time password (OTP) authentication. This ensures that legitimate users can continue while preventing unauthorized access.

High-Risk Handling: In cases of strong behavioral anomalies, the system immediately terminates the session to prevent potential security breaches.

Session-Based Feature Persistence: At the end of each session, validated behavioral data is stored for future learning. This selective persistence ensures that only trustworthy data contributes to model improvement.

Data Management and Integrity: A sliding window mechanism maintains a bounded dataset size by removing outdated records while preserving consistency between behavioral logs and feature storage.

This subsystem ensures proactive security enforcement while maintaining a balance between usability and protection.

IV. IMPLEMENTATION DETAILS

The proposed continuous authentication framework is implemented as a modular multi-component system integrating real-time behavioral monitoring, feature extraction, anomaly detection, adaptive model training, and risk-based response handling. The implementation closely follows the architectural design described in Section III, consisting of continuous monitoring, model evaluation, and security response subsystems. The system operates continuously during user sessions, enabling real-time behavioral analysis and dynamic identity verification without interrupting normal user activity.

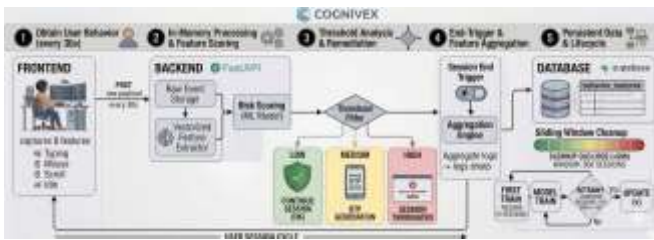


Fig. 1. Architecture diagram

A. Continuous Monitoring and Feature Extraction Implementation

The monitoring subsystem captures raw user interaction data and transforms it into structured behavioral features required for authentication. User activity is collected through the frontend using event listeners that track keyboard inputs, mouse movements, scroll behavior, and idle time. These events are aggregated over fixed intervals of 30 seconds and transmitted to the backend as structured JSON payloads.

The backend processes each incoming snapshot by storing raw behavioral data into a logging database. Feature extraction is performed dynamically on the received data, computing key behavioral metrics such as typing speed, backspace usage ratio, average keystroke interval, keystroke variance, mouse speed, mouse movement variance, scroll frequency, and idle ratio. These features are computed in-memory for immediate analysis and are not persisted during active sessions.

To ensure efficient data handling, only raw behavioral logs are stored continuously, while feature vectors are generated temporarily for risk evaluation. At the end of each session, behavioral data corresponding to low-risk activity is aggregated to form a single representative feature vector. This aggregation

process ensures that only consistent and reliable user behavior contributes to long-term learning and model training.

B. Model Training and Risk Evaluation Implementation

The model subsystem is responsible for learning user behavior patterns and identifying anomalies using machine learning techniques. The system employs the Isolation Forest algorithm for unsupervised anomaly detection due to its efficiency and suitability for high-dimensional behavioral data.

Initially, the system operates in a data collection phase where behavioral features from multiple sessions are stored without model training. Once a sufficient number of sessions is collected, the model is trained using the stored feature vectors. The trained model is serialized using joblib and stored in the database along with metadata such as model version and training count.

During active sessions, extracted feature vectors are evaluated using the trained model to compute an anomaly score. The score is obtained using the decision function of the Isolation Forest model,

$$s = f(x), \quad (1)$$

where x represents the feature vector and s denotes the anomaly score. Based on predefined thresholds, the score is mapped to risk levels such as Low, Medium, and High.

To support adaptability, the system incorporates periodic retraining using recent behavioral data. A versioning mechanism ensures that updated models replace outdated ones, while an in-memory cache is maintained for efficient real-time inference.

C. Security Response and OTP Handling Implementation

The security subsystem converts risk evaluations into appropriate system actions to maintain session integrity. For low-risk behavior, the system allows uninterrupted session continuation while logging the risk level for monitoring purposes.

In cases of medium-risk behavior, the system triggers an additional authentication step using a one-time password (OTP). The OTP is generated and stored with an expiration timestamp, and the user is prompted to verify their identity through the frontend interface. Upon successful verification, the session continues normally.

If the OTP verification fails, expires, or is ignored, the system escalates the risk level and terminates the session. For high-risk scenarios, where strong behavioral anomalies are detected, the system immediately logs out the user without initiating OTP verification, ensuring rapid response to potential threats.

This risk-aware response mechanism balances usability and security by applying additional verification only when necessary while preventing unauthorized access in critical situations.

D. System Integration and Runtime Flow

The integrated runtime pipeline operates continuously during user sessions. The frontend periodically captures behavioral data and sends snapshots to the backend through REST

API endpoints. The backend processes each snapshot by storing raw data, extracting features, and evaluating risk levels using the trained model.

If a model is not yet available, the system continues collecting data until sufficient samples are obtained. When a model exists, risk evaluation is performed in real time, and the corresponding response is sent back to the frontend. The frontend then handles user interaction, including OTP prompts or session termination actions.

At session termination, the backend aggregates low-risk behavioral data and stores the resulting feature vector for future learning. The system also applies a sliding window mechanism to maintain a bounded dataset size by removing outdated records while preserving data consistency.

E. Performance Considerations

Performance evaluation indicates that feature extraction and anomaly scoring operations are lightweight and suitable for real-time execution. The Isolation Forest model provides efficient inference with minimal computational overhead, allowing the system to process behavioral snapshots within milliseconds.

The use of in-memory processing for feature extraction reduces database load, while model caching minimizes repeated loading operations. Periodic retraining is performed asynchronously to avoid impacting real-time performance.

Overall, the system achieves efficient real-time continuous authentication with low latency, making it suitable for scalable deployment in web applications. Optimization strategies include lightweight feature computation, efficient database management, and modular system design for independent scaling of components.

V. EXPERIMENTAL RESULTS AND PERFORMANCE EVALUATION

This section presents the experimental validation of the proposed continuous behavioral authentication system in real-world web application scenarios. The system was evaluated based on behavioral feature consistency, anomaly detection accuracy, risk classification reliability, OTP verification handling, and overall system performance under dynamic user interactions.

A. Experimental Setup

Experiments were conducted using real user interaction sessions in web-based environments including:

- Normal browsing sessions
- Form input and typing-intensive tasks
- Mixed interaction scenarios with mouse and keyboard usage

System configuration:

- Web-based frontend for capturing behavioral events
- Backend server using FastAPI framework
- Database storage using Supabase
- Machine learning model implemented using scikit-learn

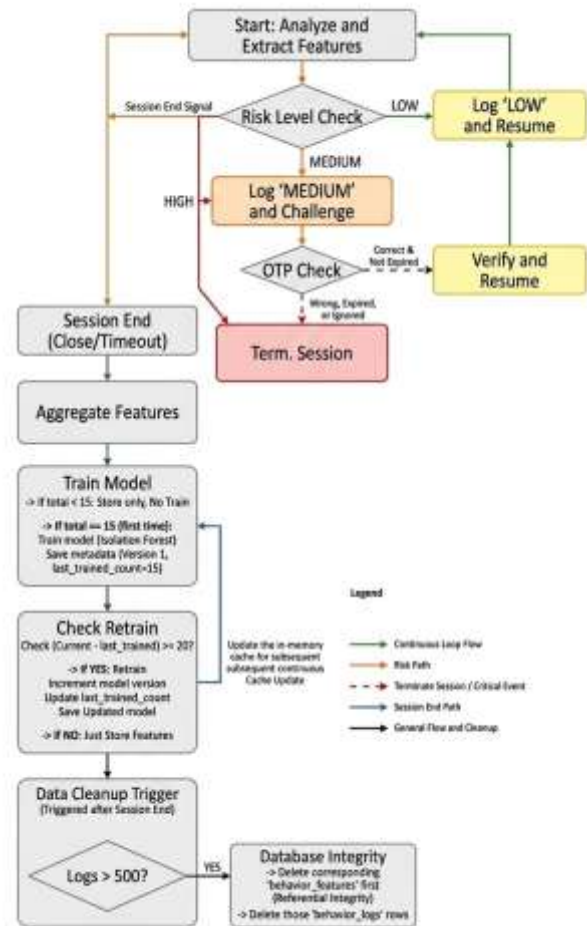


Fig. 2. Flow Diagram

Behavioral data was collected in the form of keystroke events, mouse movements, scroll interactions, and idle activity. The Isolation Forest model was used for anomaly detection, and OTP verification was integrated for adaptive authentication. Experiments were conducted across multiple sessions per user to evaluate model learning and adaptation.

B. Behavioral Feature Analysis

The extracted behavioral features demonstrated consistent patterns for individual users across multiple sessions. Features such as typing speed, keystroke intervals, mouse movement patterns, and idle ratios showed distinguishable trends unique to each user.

Observations include:

- Stable typing speed and keystroke interval distribution for the same user
- Consistent mouse movement patterns during regular interaction
- Noticeable variation in idle behavior across different usage scenarios

These consistent behavioral characteristics enabled effective user profiling and reliable feature representation for anomaly detection.

C. Anomaly Detection Performance

The Isolation Forest model effectively distinguished between normal and anomalous user behavior. During testing, deviations from learned behavioral patterns resulted in lower anomaly scores, indicating potential risk.

The anomaly scoring function is given by:

$$s = f(x) \quad (2)$$

where x represents the behavioral feature vector and s denotes the anomaly score.

Performance observations:

- Accurate identification of consistent user behavior as low risk
- Detection of irregular interaction patterns as medium or high risk
- Robust performance across different user interaction scenarios

The model demonstrated reliable classification capability even with moderate variations in user behavior.

D. Risk Classification and Response Evaluation

The system classified user activity into three risk levels based on anomaly scores: Low, Medium, and High. Each risk level triggered an appropriate system response.

Evaluation results include:

- Low-risk sessions proceeded without interruption
- Medium-risk sessions successfully triggered OTP verification
- High-risk sessions resulted in immediate session termination

OTP verification was validated by testing both correct and incorrect inputs. Successful verification allowed session continuation, while failed or expired OTP attempts resulted in session termination.

E. Session-Based Learning and Adaptation

The system demonstrated effective learning behavior through session-based feature aggregation and periodic model retraining.

Observations:

- Initial sessions were used for data collection without model training
- Model training after sufficient sessions improved detection accuracy
- Periodic retraining enabled adaptation to gradual behavioral changes

The use of a sliding window mechanism ensured that outdated data was removed while maintaining recent behavioral patterns for training.

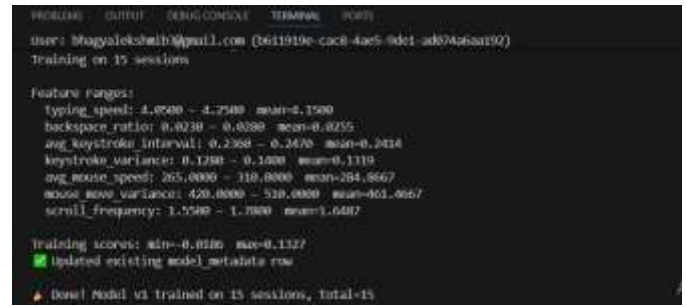


Fig. 3. Behavioral feature analysis and Isolation Forest model training based on collected user sessions



Fig. 4. Risk-based authentication mechanism showing OTP challenge triggered for anomalous user behavior

F. System Performance Evaluation

The overall system demonstrated efficient real-time performance during continuous authentication.

Key performance metrics include:

- Low-latency processing of behavioral snapshots (within milliseconds)
- Efficient anomaly detection using lightweight Isolation Forest model
- Minimal impact on user experience due to passive monitoring
- Scalable backend performance for multiple concurrent sessions

The use of in-memory feature processing and model caching further improved system efficiency.

G. Overall System Performance

The integrated system achieved:

- Real-time continuous authentication during active sessions
- Accurate anomaly detection and risk classification
- Effective adaptive authentication using OTP verification
- Reliable session management and security enforcement

The modular architecture enabled independent evaluation and optimization of each subsystem, ensuring flexibility and scalability for real-world deployment.

VI. DISCUSSION

The proposed continuous authentication framework demonstrates that real-time user verification can be effectively achieved using behavioral biometrics combined with

lightweight machine learning techniques. Unlike traditional static authentication methods, the system provides continuous and passive identity validation without interrupting the user, thereby enhancing both security and usability.

The integration of behavioral feature extraction with Isolation Forest-based anomaly detection enables the system to learn user-specific interaction patterns and detect deviations in real time. The use of session-based learning and adaptive retraining further improves system robustness by allowing gradual adaptation to changes in user behavior over time.

The risk-based response mechanism, incorporating OTP verification for moderate anomalies and immediate session termination for critical deviations, ensures a balanced trade-off between security and user convenience. This adaptive approach minimizes unnecessary interruptions while maintaining strong protection against unauthorized access.

However, behavioral patterns can vary due to contextual factors such as user fatigue, device changes, or environmental conditions. While the system accounts for gradual variations through retraining, sudden behavioral shifts may still affect detection accuracy. Future improvements could involve incorporating contextual awareness and multi-modal behavioral signals to enhance reliability.

VII. LIMITATIONS

Despite promising results, certain limitations were observed:

- Behavioral patterns may vary due to external factors such as device type, user mood, or environmental conditions.
- Initial data collection phase requires multiple sessions before effective model training can be performed.
- Isolation Forest may produce false positives in cases of sudden but legitimate behavioral changes.
- OTP-based verification introduces minor interruption in medium-risk scenarios.
- The system relies on continuous frontend event capture, which may be affected by browser restrictions or network latency.

VIII. CONCLUSION

This paper presented a real-time continuous authentication framework designed to enhance system security through behavioral monitoring and intelligent anomaly detection. The proposed system integrates behavioral feature extraction, session-based monitoring, Isolation Forest-based anomaly detection, and risk-based adaptive authentication within a unified modular architecture. By combining behavioral biometrics with machine learning techniques, the system achieves robust user verification and anomaly detection in dynamic usage environments.

The integration of Isolation Forest for unsupervised anomaly detection enhances reliability in identifying deviations from normal user behavior, while the adaptive authentication module enables secure and efficient responses through risk-based OTP verification.

Experimental evaluations demonstrate that the proposed framework operates efficiently in real time with minimal

computational overhead, highlighting its practical feasibility for scalable and lightweight deployment. Furthermore, the inclusion of session-based data storage and periodic retraining enables the system to adapt to evolving user behavior over time, improving long-term accuracy and robustness.

Overall, the proposed architecture establishes a scalable foundation for continuous authentication systems that combine cybersecurity, machine learning, and behavioral biometrics, contributing toward enhanced security and reduced reliance on traditional static authentication mechanisms in modern digital environments.

IX. FUTURE WORK

Although the proposed system demonstrates effective real-time continuous authentication and anomaly detection, several directions exist for further improvement and research exploration. Future work may investigate the integration of advanced behavioral biometrics such as keystroke dynamics, touch patterns, and multi-device interaction data to obtain more robust and comprehensive user profiling. While the current approach utilizes selected behavioral features, incorporating multimodal inputs could improve detection accuracy in complex usage scenarios.

An important direction involves the integration of adaptive and context-aware learning mechanisms to enable dynamic user modeling and risk assessment. Although the current system performs session-based monitoring and retraining, future enhancements could incorporate online learning techniques to continuously update user profiles with minimal latency and improved responsiveness.

Deployment on edge computing platforms represents an essential step toward practical real-world implementation. Optimizing models for lightweight execution on user devices or distributed environments would reduce latency, enhance privacy by minimizing data transmission, and enable seamless continuous authentication without reliance on centralized systems.

Future research may also explore the use of advanced anomaly detection techniques, including deep learning-based models and hybrid approaches, which could improve robustness against sophisticated attacks and reduce false positives in dynamic environments.

Finally, large-scale user evaluations across diverse user groups are necessary to assess system performance, usability, and real-world reliability. User-centered studies will provide valuable insights for improving system design, enhancing user experience, and ensuring the practical adoption of continuous authentication frameworks in real-world applications.

ACKNOWLEDGMENT

We express our sincere gratitude to all individuals and organizations that have contributed to the successful completion of this research work. We extend our heartfelt appreciation to our academic institution, Mar Athanasius College of Engineering, for providing the necessary infrastructure, research facilities,

and technical environment that enabled the development and evaluation of the proposed continuous authentication system. We are deeply grateful to our project advisors and faculty mentors for their invaluable guidance, continuous support, and constructive feedback throughout this research. Their expertise in cybersecurity, machine learning, and intelligent systems significantly influenced the design decisions and helped us overcome several technical challenges encountered during system development and integration.

We extend our appreciation to the open-source research community and developers whose contributions have been fundamental to this project. Frameworks and tools such as FastAPI, Python-based machine learning libraries, and data processing frameworks enabled efficient implementation of behavioral monitoring, anomaly detection using Isolation Forest, and real-time authentication mechanisms that form the backbone of the proposed system.

We further acknowledge the contributions of researchers in the fields of behavioral biometrics, anomaly detection, and zero trust security models, whose prior work has provided the scientific foundation and inspiration for this study. Lastly, we express our heartfelt gratitude to our families and friends for their constant encouragement, patience, and motivation throughout this research journey.

REFERENCES

- [1] "AI-powered anomaly detection in zero trust environments: A comprehensive review of methods and evaluation," K. Sandhu, S. G. R. Bojja, S. Venkataramanan, V. K. R. Vangoor, and S. Thota, *Nanotechnology Perceptions*, vol. 20, no. S16, pp. 2634–2654, 2024.
- [2] "A generic privacy-preserving protocol for keystroke dynamics-based continuous authentication", A. F. Baig and S. Eskeland, in *Proc. 19th Int. Conf. Security and Cryptography (SECRYPT)*, 2022, pp. 491–498, doi: 10.5220/0011141400003283.
- [3] "Continuous authentication using mouse dynamics," S. Mondal and P. Bours, in *Proc. Int. Conf. on Biometrics and Kansei Engineering (ICBAKE)*, Gjøvik, Norway, pp. 1–8.
- [4] "Behavioral biometrics: A powerful defense against social engineering attacks," K. V. K. Pasumarthi, *Journal of Computer Science and Technology Studies*, vol. 7, no. 8, pp. 1166–1173, Aug. 2025, doi: 10.32996/jcsts.2025.7.8.132.
- [5] "AI-powered zero trust access evaluation using behavioral fingerprinting," H. Yadav, G. Subramanian Ramachandran, and K. Sharma, *International Journal of Computer Applications*, vol. 187, no. 16, June 2025.
- [6] "Privacy-preserving continuous authentication using behavioral biometrics," A. F. Baig, S. Eskeland, and B. Yang, *International Journal of Information Security*, vol. 22, no. 6, pp. 1833–1847, Jul. 2023, doi: 10.1007/s10207-023-00721-y.
- [7] "User authentication method based on keystroke dynamics and mouse dynamics with scene-irrelated features in hybrid scenes," X. Wang, Y. Shi, K. Zheng, Y. Zhang, W. Hong, and S. Cao, *Sensors*, vol. 22, no. 17, p. 6627, Sep. 2022, doi: 10.3390/s22176627.
- [8] "Behavioral biometrics for continuous authentication", A. Gunuganti, *Journal of Biosensors and Bioelectronics Research*, vol. 1, no. 3, pp. 1–5, 2023, doi: 10.47363/JBBER/2023(1)122.
- [9] "Continuous Authentication and Behavioral Biometrics: Enhancing Cybersecurity in the Digital Era," S. Oduri, *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)*, vol. 13, no. 7, July 2024, pp. 13632–13640, doi: 10.15680/IJIRSET.2024.1307140.
- [10] "Zero-Trust Network Access (ZTNA)", V. Mavroudis, *Alan Turing Institute*, 2024.