A LIGHTWEIGHT FRAMEWORK FOR PRIVACY-PRESERVING BEHAVIORAL AUTHENTICATION: BALANCING RECOGNITION ACCURACY AND SYSTEM LATENCY

A PROJECT PROPOSAL SUBMITTED BY

W.M.T.R.S Weerakoon
(S20545)

to the

DEPARTMENT OF STATISTICS AND COMPUTER SCIENCE

in partial fulfillment of the requirements of the

Degree of Bachelor of Science (Honours) in Computer Science
of the

UNIVERSITY OF PERADENIYA, SRI LANKA

2 February, 2026

Preface

This is a proposal for the Project in Computer Science II (Research Project – CSC4996) for partial fulfillment of the requirements of the Degree of Bachelor of Science (Honours) in Computer Science at the Department of Statistics and Computer Science, University of Peradeniya, Sri Lanka.

This proposal provides the scope and context of the research project to be undertaken. It details the aims and research questions, background, methodology, and project design. This document also provides a schedule for the completion of the project, including a description of anticipated results and final products

The intended audience of this document is the academic staff of the Department of Statistics and Computer Science, University of Peradeniya, who will evaluate the project to determine whether it should be approved as proposed, approved with modifications, or not approved.

Table of Contents

List of Figures

List of Tables

List of Equations

List of Abbreviations

ACM	Association for Computing Machinery
BB-MAS	Behavioral Biometrics Multi-device and multi-Activity from Same users
CKKS	Cheon-Kim-Kim-Song (Homomorphic Encryption Scheme)
CMU	Carnegie Mellon University
Deep SVDD	Deep Support Vector Data Description
DL	Deep Learning
EER	Equal Error Rate
FAR	False Acceptance Rate
FHE	Fully Homomorphic Encryption
FN	False Negative
FP	False Positive
FRR	False Rejection Rate
HE	Homomorphic Encryption
IEEE	Institute of Electrical and Electronics Engineers
IND-CPA	Indistinguishability under Chosen Plaintext Attack
IoT	Internet of Things
JL	Johnson-Lindenstrauss
LHE	Leveled Homomorphic Encryption
LSTM	Long Short-Term Memory
MSE	Mean Squared Error
PII	Personally Identifiable Information
PIN	Personal Identification Number
R-RCM	Robust Recurrent Confidence Model
RNN	Recurrent Neural Network
SU-AIS	Syracuse University and Assured Information Security
SVDD	Support Vector Data Description
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

1.0 Introduction

Overview of the Research Domain In mechanisms such as Personal Identification Number (PIN) and passwords are increasingly becoming single points of failure [1]. They are susceptible to vulnerabilities like shoulder surfing, brute-force attacks, and social engineering [1], [2]. To address these vulnerabilities, Behavioral Biometrics—specifically Keystroke Dynamics and Mouse/Touch Dynamics—has emerged as a powerful alternative [3], [4]. Unlike static passwords, behavioral biometrics allow for continuous, passive authentication, verifying a user's identity based on *how* they interact with a device rather than *what* they know [5], [6].

1.1 Problem Statement

The primary problem addressed by this research is the critical trade-off between **user privacy** and **system latency** in behavioral authentication systems on **resource-constrained edge devices**. While behavioral biometrics—such as keystroke dynamics and touch interactions—offer a robust solution for continuous authentication, the storage and processing of these behavioral patterns present significant security risks [3].

This issue affects millions of users across **heterogeneous platforms, ranging from high-end desktops to resource-constrained mobile and IoT devices**. Unlike passwords, behavioral biometrics are immutable; a user cannot change their typing rhythm or hand geometry if the biometric template is compromised. Therefore, a breach of raw behavioral data constitutes a permanent loss of digital identity [1].

Current approaches fail to address this problem effectively due to a technical dichotomy between accuracy and efficiency:

- **Privacy Gaps in High-Accuracy Models:** State-of-the-art Deep Learning (DL) frameworks, such as those utilizing Recurrent Neural Networks (RNNs) or Image-based Encoding (e.g., KDPrint), achieve low Equal Error Rate (EER) but typically require the storage of **raw behavioral features** [2], [4]. This creates a single point of failure where the database becomes a high-value target for attackers.
- **Efficiency Gaps in Privacy-Preserving Methods:** Conversely, strong cryptographic solutions like FHE allow for secure computation but suffer from prohibitive computational overhead. Research indicates that such heavy Homomorphic Encryption (HE) schemes often introduce latencies ranging from seconds to minutes, rendering them impractical for **real-time,**

continuous authentication where decisions must be made in milliseconds to maintain a seamless user experience [3]. **Therefore, applying HE to every single micro-interaction is computationally infeasible for real-time monitoring. This necessitates a hybrid approach where HE is reserved for critical decision points.**

There is currently no unified framework that effectively balances these conflicting requirements. This research seeks to bridge this gap by proposing an **adaptive dual-mode architecture** that utilizes **Orthogonal Random Projections** for lightweight, continuous monitoring and reserves HE for high-security checkpoints, ensuring privacy without degrading the speed required for edge devices.

1.2 Research Aim and Objectives

The primary aim of this research is to develop a lightweight, privacy-preserving framework for continuous behavioral authentication on **resource-constrained edge devices**. This framework seeks to balance recognition accuracy and system latency by utilizing Orthogonal Random Projections for template security and Deep Support Vector Data Description (Deep SVDD) for efficient anomaly detection.

The technical feasibility of utilizing Orthogonal Random Projections and Deep SVDD for anomaly detection has been validated through a functional code prototype. Preliminary execution results demonstrating the successful separation of genuine user patterns from impostors are detailed in Appendix A.

Preliminary results for this objective are documented in Appendix 1.1.

1. **To design a privacy-preserving feature transformation pipeline:** Develop a mechanism using **Orthogonal Random Projections (JL Lemma)** [7] that secures behavioral biometric templates (making them mathematically irreversible) while preserving the Euclidean distances required for accurate pattern recognition.

2. **To optimize feature engineering for mobile efficiency:** Implement **KDPrint-style standardization** to transform raw time-series data into standardized image encodings, ensuring high recognition accuracy without the noise sensitivity of Min-Max scaling [4].

3. **To implement a lightweight anomaly detection model:** Develop a **Deep SVDD** classifier [8] capable of running offline on **mid-range edge devices** to distinguish between genuine users and imposters with minimal computational overhead.

4. **To evaluate the trade-off between privacy, accuracy, and latency:** Conduct a comparative analysis of the proposed framework against existing baselines (such as raw-data RNNs and HE), measuring performance metrics including **EER**, **System Latency (ms)**, and **Memory Usage** [2], [8].

1.3 Research Questions

To address the identified gaps in privacy and efficiency, this research aims to answer the following key questions:

1. **Primary Research Question:** To what extent can **Orthogonal Random Projections** (based on the JL Lemma) balance the conflicting requirements of template privacy, recognition accuracy, and system latency in behavioral authentication?

2. **Impact on Privacy and Irreversibility:** How effective is the proposed projection mechanism in rendering behavioral templates mathematically irreversible to attackers, compared to storing raw features or using standard Min-Max scaling?

3. **Feasibility for Resource-Constrained Environments:** Can a **Deep SVDD** anomaly detection model achieve real-time authentication latency (e.g., $<$ 200ms) on **resource-constrained edge devices** without exceeding memory constraints?

2.0 Background

2.1 Technical and Theoretical Background

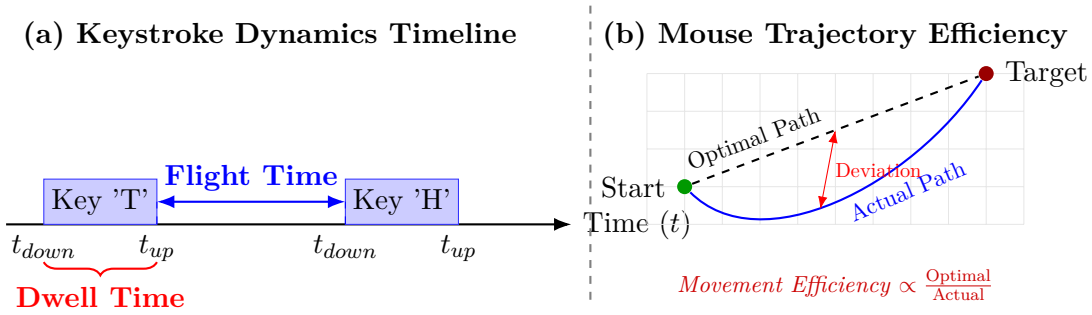

Figure 1: Visual Representation of Biometric Features. **(a)** Keystroke Dynamics: Dwell Time represents the key press duration, while Flight Time measures the latency between distinct keystrokes. **(b)** Mouse Dynamics: Comparison between the optimal straight-line path and the actual user trajectory, used to calculate movement efficiency and curvature.

2.1.1 Keystroke Dynamics

This is the measurement of biomechanical typing patterns. The fundamental features include Dwell Time (duration a key is pressed) and Flight Time (latency between releasing one key and pressing the next) [9], [10]. These features form a unique "digital signature" for each user [5].

2.1.2 Mouse Dynamics

This involves analyzing the unique behavioral patterns of a user's mouse interactions [11]. Unlike simple click-tracking, this research focuses on complex motor-skill features:

- **Movement Efficiency:** The ratio of the straight-line distance to the actual path taken [12].
- **Velocity & Acceleration Profiles:** The rate of speed change as the cursor approaches a target [13].
- **Click-to-Click Latency:** The timing between releasing a button and moving to the next location.
- **Drag-and-Drop characteristics:** The pressure and speed consistency during sustained click events.

2.1.3 Recurrent Neural Networks (RNNs) & LSTMs

Since keystroke data is inherently sequential, standard feed-forward networks often fail to capture temporal dependencies. RNNs, and specifically Long Short-Term Memory (LSTM) networks, are theoretically suited for this task as they maintain a "memory" of previous inputs, allowing them to model complex typing rhythms [2], [14].

2.1.4 Homomorphic Encryption (HE)

To ensure privacy, the system employs HE [15], a cryptographic form that allows computations to be performed on encrypted data without first decrypting it. This ensures that the user's raw biometric template is never exposed in plaintext during the authentication process.

2.1.5 Johnson-Lindenstrauss (JL) Lemma

To address the "curse of dimensionality" and system latency, this research utilizes the JL Lemma [7]. This mathematical theorem states that points in a high-dimensional space can be projected into a lower-dimensional space using Orthogonal Random Projections while approximately preserving the Euclidean distances between them. This allows for lightweight processing without significant loss of accuracy.

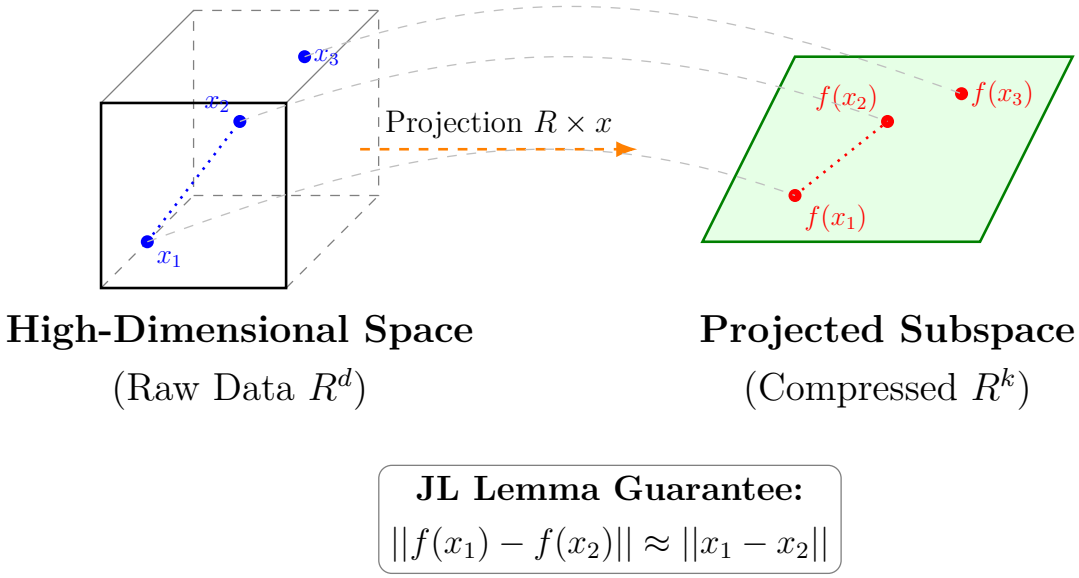

Figure 2: Conceptual Visualization of the JL Lemma [7]. Points from a high-dimensional feature space (Left) are projected onto a lower-dimensional subspace (Right). The blue and red dotted lines illustrate that the relative Euclidean distances between points are approximately preserved despite the massive reduction in dimensions.

2.2 Literature Review

2.2.1 Overview of Reviewed Literature

To establish a theoretical framework for this research, a comprehensive review of existing literature was conducted, focusing on Keystroke Dynamics, Mouse Dynamics, and Privacy-Preserving Machine Learning. The following table summarizes the key research papers referenced, highlighting their specific contributions to this project ("Key Takeaway") and the limitations ("Research Gap") that this proposal aims to address.

Table 1: Summary of Key Related Studies

Reference & Study	**Key Contribution to This Project**	**Identified Gap / Limitation**
Gaines et al. (1980) [9] & Joyce et al. (1990) [10]	**Foundational Theory:** Established that typing rhythms (dwell/flight time) are unique and stable enough for identity verification.	Relied on static, fixed-text passwords, which are insufficient for continuous authentication.
Mondal & Bours (2017) [12]	**Multimodal Fusion:** Proved that combining Keystroke and Mouse dynamics significantly reduces EER compared to single modalities.	Fusion was achieved by simply concatenating features, creating a high-dimensional vector that slows down real-time processing.
Kim et al. (2018) [16]	**Feature Engineering:** Introduced "user-adaptive" features, showing that personalized feature selection improves accuracy.	Focused entirely on accuracy; lacked any "template protection" or encryption to secure the stored data.
Kim et al. (2024) [4]	**Deep SVDD Validation:** Demonstrated that Deep SVDD outperforms traditional models (6.7% EER) by encoding time-series data into images.	Restricted to *mobile PINs* (touch interactions) and lacked cryptographic encryption (HE) or multimodal fusion (Mouse).
Kiyani et al. (2020) [2]	**Continuous Authentication:** Validated the use of RNNs for verifying users continuously, not just at login.	Did not address the high latency introduced when trying to add encryption to these continuous streams.
Rahman et al. (2021) [3]	**Scalability Metrics:** Provided a framework for measuring how error rates grow as the user database size increases.	Addressed scalability of accuracy but not the scalability of privacy (how to store millions of secure templates).

2.2.2 Foundational Studies (Static Authentication)

Early research laid the groundwork for typing pattern analysis. Gaines et al. (1980) [9] demonstrated that typing rhythms are unique to individuals, identifying that even simple digraph latencies could distinguish users with high confidence. Building on this, Joyce and Gupta (1990) [10] formalized the use of "latency signatures" for identity verification, showing that comparing a test signature against a mean reference signature could achieve low impostor pass rates. Monrose et al. (1999) [6] extended this concept to "password hardening," combining typing rhythms with passwords to increase entropy against offline attacks.

2.2.3 Feature Engineering and Adaptive Systems

As research matured, the focus shifted to handling the variability in human behavior. Kim et al. (2018) [16] highlighted the limitations of fixed-text authentication and proposed "user-adaptive feature extraction." Their work demonstrated that by dynamically selecting features based on a user's specific typing speed ranks, the EER could be significantly reduced. This underscored the need for personalized models in behavioral authentication.

2.2.4 Mouse Dynamics and Multimodal Fusion

Research into mouse dynamics has evolved from simple statistics to complex trajectory analysis. Ahmed and Traore (2007) [11] pioneered the use of movement speed and direction histograms. Later, Zheng et al. (2011) [13] improved this by analyzing "point-by-point" angle-based metrics, proving that fine-grained motor skills are harder to forge.

Recent studies have shifted toward Multimodal Authentication, combining keystroke and mouse data. Mondal and Bours (2017) [12] demonstrated that fusing these two biometrics significantly reduces the EER. However, existing multimodal frameworks often simply concatenate feature vectors, leading to massive dimensionality that slows down real-time processing—a problem this research aims to solve using the JL Lemma [7].

2.2.5 Deep Learning and Continuous Authentication

Recent approaches have adopted DL to improve accuracy. Zareen et al. (2018) [14] utilized Bayesian Regularized Neural Networks, achieving an EER of 0.9%. Kiyani et al. (2020) [2] proposed an Robust Recurrent Confidence Model (R-RCM) using ensemble learning for continuous monitoring.

Most recently, **Kim et al. (2024)** [4] proposed "KDPrint," a method that transforms keystroke dynamics into images and utilizes **Deep SVDD** [8] for anomaly detection. Their work achieved a notable EER of 6.7% on mobile devices, validating Deep SVDD as a superior classifier for one-class authentication tasks. However, their approach focused exclusively on mobile PINs and lacked cryptographic privacy.

2.2.6 Scalability and System Performance

While accuracy has improved, scalability remains a challenge. Rahman et al. (2021) [3] analyzed how verification error rates increase as the user database grows, highlighting that system performance must be evaluated not just on accuracy, but on its ability to handle large-scale deployments while maintaining privacy.

2.3 Research Gap

Despite the extensive literature on improving the accuracy of behavioral biometrics [2], [14] and recent advancements in DL anomaly detection [4], a critical trilemma remains unsolved: balancing **Accuracy**, **Efficiency**, and **Privacy**.

1. **Computation vs. Latency in Multimodal Systems:** Integrating Mouse Dynamics with Keystroke Dynamics doubles the feature space complexity. While recent studies like Kim et al. (2024) [4] successfully utilized Deep SVDD for mobile PINs, they focused on single-modality touch data. Processing a high-dimensional, multimodal stream (typing + mouse trajectories) creates a computational bottleneck that current frameworks fail to address efficiently without dimensionality reduction.

2. **Privacy Vulnerability (Lack of Encryption):** Most existing frameworks focus on verifying raw feature vectors or transformed representations. For instance, while Kim et al. (2024) [4] introduced "image encoding" to obscure raw keystrokes, this is a feature transformation technique, not a cryptographic privacy guarantee. It lacks the mathematical irreversibility of **HE** [15]. If the central database is compromised, these behavioral templates are susceptible to reverse-engineering or replay attacks [1].

3. **Lack of Integrated Privacy-Preserving Architectures:** While cryptographic solutions exist, standard encryption prevents the system from performing the distance calculations needed for authentication (like Euclidean distance). There is currently no implementation that combines the anomaly detection power of **Deep SVDD** (validated by [4], [8]) with HE for secure, privacy-preserving inference.

2.3.1 Summary of Novel Contributions

This study makes the following original contributions to the field of behavioral authentication:

1. **Dual-Mode Privacy Architecture**

   This research proposes a novel dual-mode behavioral authentication framework that separates high-security encrypted inference (Cheon-Kim-Kim-Song (Homomorphic Encryption Scheme) (CKKS)-based) from lightweight projected privacy mode (JL-based), enabling adaptive deployment based on transaction risk level. The architecture allows the system to dynamically balance security guarantees and computational efficiency.

Table 2: Comparative Analysis: JL Lemma (Projected Privacy) vs. FHE (Cryptographic Privacy)

Property	**JL Lemma (Mode II)**	**FHE/CKKS (Mode I)**
Core Concept	Dimensionality reduction while preserving Euclidean distances [7].	Computation on encrypted data without decryption [15].
Data State	Projected into a lower-dimensional ($k \ll d$) subspace [7].	Encrypted as ciphertexts (IND-CPA secure) [15].
Latency	Ultra-low ($<$ 200ms), suitable for continuous monitoring [3].	High overhead, suitable for periodic high-risk tasks [3].
Irreversibility	Mathematically indeterminate/non-invertible compression.	Cryptographically secure based on hard problems [15].
Accuracy	Slight EER trade-off due to distance approximation [4].	High accuracy, arithmetic is preserved in ciphertext [15].

2. **Dimensionality-Constrained Encrypted Inference**

   Unlike prior work that applies Deep SVDD to raw or image-encoded features [4], this study evaluates Deep SVDD operating on JL compressed embeddings. It explicitly analyzes how dimensionality reduction affects EER, inference latency, and encryption overhead, thereby formalizing the impact of projection dimension k on encrypted biometric verification [7], [15].

3. **Formal Privacy–Latency–Accuracy Trade-off Quantification**

This work introduces an integrated evaluation framework that jointly measures:

- Biometric performance (EER, False Acceptance Rate (FAR), False Rejection Rate (FRR))
- Computational efficiency (Inference Latency, Encryption Overhead Ratio) [3]
- Information-theoretic privacy (Mutual Information, Reconstruction Error)

To the best of our knowledge, no prior behavioral biometric study simultaneously quantifies these three dimensions under homomorphically encrypted inference.

4. **Encrypted One-Class Authentication Validation**

   This study provides the first empirical validation of Deep SVDD [8] distance-based anomaly detection executed over CKKS-encrypted, JL-projected multimodal biometric templates (keystroke + mouse dynamics), demonstrating the feasibility of privacy-preserving one-class authentication in resource-constrained environments [1], [15].

2.3.2 Gap Definition

There is currently no unified framework that utilizes **Orthogonal Random Projections (JL Lemma)** [7] to compress the combined feature space of both Keystroke and Mouse Dynamics for lightweight processing, while simultaneously preserving privacy using **HE** [15]. Unlike Kim et al. (2024) [4], which applies Deep SVDD to unencrypted mobile data, this study aims to bridge the gap by creating a fast, *cryptographically secure*, multimodal authentication system for desktop environments.

2.4 Assumptions and Constraints

2.4.1 Assumptions

- It is assumed that the user's typing behavior is relatively stable over short periods but may exhibit gradual "concept drift" which the model must accommodate [2].
- It is assumed that the users are utilizing standard physical keyboards; virtual/touchscreen keyboards are outside the scope of this specific study.

- It is assumed that the mouse data collection frequency (e.g., 50Hz) is sufficient to capture micro-movements without overwhelming the system bus [11].
- The ”trust” of the endpoint device (personal laptop) is assumed for the initial data capture phase before HE transformation.

2.4.2 Constraints

- **Hardware Limitations:** While Model Training will utilize a standard consumer laptop (RTX 3050 GPU) to handle the optimization of the Deep SVDD parameters, the Inference Phase (the actual authentication) is constrained to run on the CPU with a target latency of ¡200ms. This simulates the processing power of a mid-range mobile processor, ensuring the framework is truly lightweight
- **Data Availability:** The research is constrained by the use of public datasets (e.g., Syracuse University and Assured Information Security (SU-AIS) Behavioral Biometrics Multi-device and multi-Activity from Same users (BB-MAS) [17]) that may not perfectly reflect the specific ”free-text” behavior required for continuous authentication.
- **Encryption Overhead:** HE [15] introduces significant computational overhead. The system is constrained to optimize this trade-off to ensure the authentication decision happens within a usable timeframe (e.g., under 200ms per window) [3].

3.0 Methodology and Project Design

Rationale for Chosen Methods:

- **Multimodal Fusion:** Single-modality systems (keystroke only) are prone to mimicry attacks. Fusion with mouse dynamics increases the entropy of the user profile, making forgery exponentially harder [12].
- **Deep Support Vector Data Description (SVDD) & LSTMs:** Unlike static classifiers (e.g., Support Vector Machine (SVM)), LSTM networks are selected for their ability to model the temporal dependencies in sequential data [2]. Deep SVDD is chosen as the anomaly detector because it is a "one-class" classifier, meaning it can train on only the legitimate user's data without requiring impostor data during the training phase [4], [8].
- **JL Lemma:** To counter the computational overhead of HE [15], the JL Lemma [7] is applied to project high-dimensional biometric feature vectors into a lower-dimensional space. This allows for faster encrypted computations with mathematically guaranteed distance preservation [3].

3.1 Overview of the Proposed Methodology/Research Design

1. **Feature Extraction & Temporal Fusion:** The system ingests raw event logs and converts them into synchronized time-series feature vectors.
    - **Keystroke Features:** Extraction of Flight Time (latency between $\text{KeyUP}_n \rightarrow \text{KeyDOWN}_{n+1}$) and *Dwell Time* (duration of $\text{KeyDOWN}_n \rightarrow \text{KeyUP}_n$) [9], [10].
    - **Mouse Features:** Calculation of higher-order motor metrics including Velocity Profiles, Angular Velocity, and Curvature Distance Ratio (efficiency of movement) [11], [13].
    - **Multimodal Fusion:** The two independent streams are aligned using sliding time windows (e.g., $t = 10s$) to create unified "behavioral frames" representing the user's complete interaction state [2], [12].

Table 3: Technical Specification of Extracted Behavioral Features

Category	Modality	Specific Metrics	Rationale
Temporal	Keystroke	Dwell Time, Flight Time [2], [9], [10]	Captures unique typing rhythm and biomechanical speed [1], [5].
Kinematic	Mouse	Velocity Profiles, Angular Velocity [11], [13]	Distinguishes fine-grained motor-skill characteristics [11].
Trajectory	Mouse	Curvature Distance Ratio, Movement Efficiency [12], [13]	Measures hand stability and trajectory optimization [12].
Sequential	Multimodal	Sliding Window Vectors (e.g., $t = 10s$) [2]	Synchronizes independent streams for continuous monitoring [12].

2. **Dimensionality Reduction (JL Layer):** To mitigate the "curse of dimensionality" caused by fusing two biometric streams, the high-dimensional fused vector (d) is projected onto a lower-dimensional subspace (k, where $k \ll d$) using the JL Lemma [7]. This is achieved by multiplying the feature vector by a sparse random matrix (R) to produce a compressed, privacy-hardened embedding.

3. **Privacy-Preserving Transformation:** The compressed embeddings are encrypted using a **Leveled Homomorphic Encryption (LHE)** scheme (e.g., CKKS) [15]. Unlike additive-only schemes, CKKS supports the approximate arithmetic and multiplication depth required to compute *squared Euclidean distances* ($||x - c||^2$) directly on the ciphertext. This ensures that the mathematical operations needed for authentication are performed entirely in the encrypted domain without decrypting the user's behavioral template [3].

4. **Anomaly Detection (Deep SVDD):** The encrypted feature vectors are fed into a Deep SVDD model [8]. This model learns a compact hypersphere boundary encapsulating the legitimate user's "normal" behavior. During the verification phase, the system calculates the distance between the encrypted input and the hypersphere center; any input falling outside this learned radius is flagged as an anomaly (potential impostor) while the data remains

mathematically indistinguishable from random noise under IND-CPA security [4], [8].

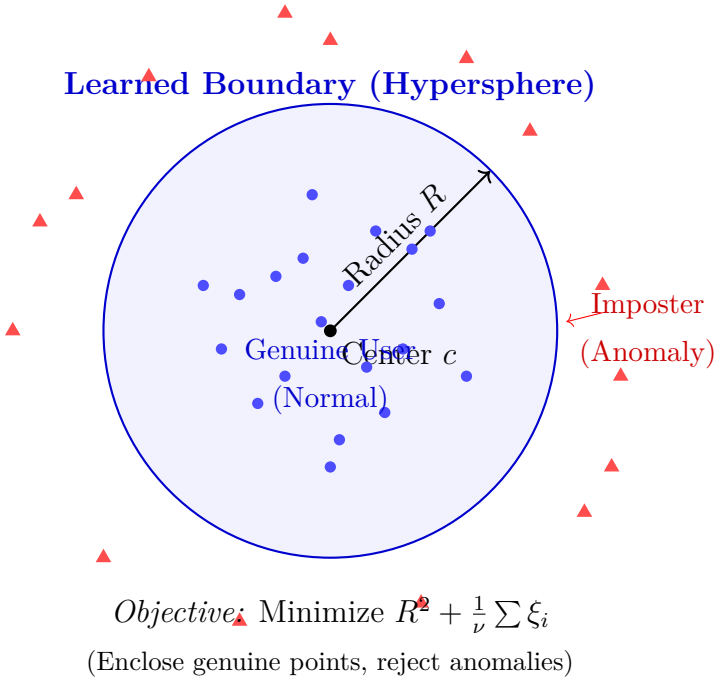

Figure 3: Visualizing the Deep SVDD Decision Boundary. The model learns a compact hypersphere of radius R around center c that encapsulates the majority of legitimate user data (blue dots). Any input falling outside this boundary is flagged as an anomaly (red triangles), representing a potential impostor.

3.1.1 Architectural Design Mode I: Cryptographic Privacy (High-Security)

Illustrated in Figure 1. This mode is designed for **sporadic, high-risk transaction scenarios**, such as banking logins or authorizing payments. It utilizes **LHE (CKKS)** [15] to perform anomaly detection inference entirely within the encrypted domain.

- **Privacy Mechanism:** The behavioral template is encrypted using standard cryptographic protocols, ensuring mathematically provable security (IND-CPA) [15]. The raw biometric data is never exposed to the server.
- **Latency Target (Interactive):** $1.0 - 2.0$ **seconds.** Since this mode is triggered only during critical decision points (not continuously), a slight processing delay is acceptable to the user in exchange for maximum security [3].

- **Trade-off:** While this offers the highest level of data protection, it incurs a significant computational cost, making it suitable for **periodic, critical authentication** rather than continuous, sub-second monitoring [3].

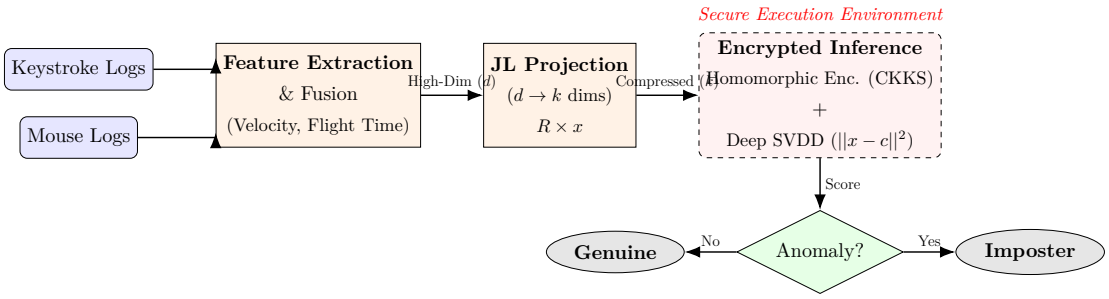

Figure 4: System Architecture of the Proposed Lightweight Privacy-Preserving Framework.

- **Privacy Mechanism:** The behavioral template is encrypted using standard cryptographic protocols, ensuring mathematically provable security (IND-CPA) [15].
- **Trade-off:** While this offers the highest level of data protection, it incurs a higher computational cost, making it suitable for periodic, critical authentication rather than continuous, sub-second monitoring [3].

3.1.2 Architectural Design Mode II: Projected Privacy (High-Performance)

Illustrated in Figure 2. This mode is designed for continuous, passive background monitoring. It removes the heavy encryption overhead and relies on the **Johnson-Lindenstrauss (JL) Projection** as a form of non-invertible compression.

To mitigate the 'curse of dimensionality' while maintaining template privacy, the system utilizes a JL-projection layer. A Python-based implementation of this transformation, combined with the Deep SVDD decision boundary, is provided in Appendix A. This prototype confirms that the hypersphere boundary can effectively encapsulate legitimate user data while rejecting outliers with a high degree of accuracy. 1.1.

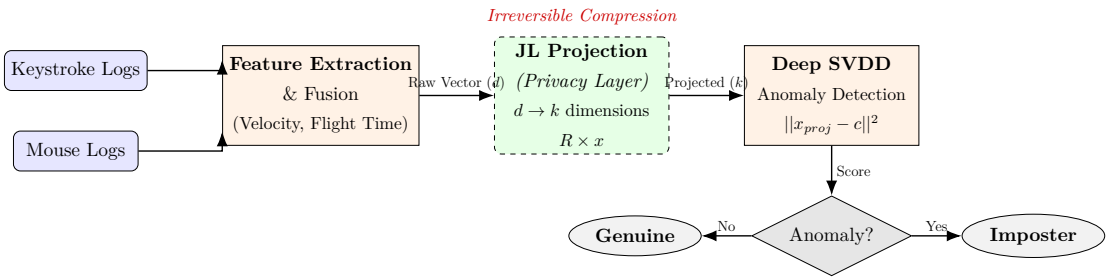

Figure 5: Proposed Lightweight Architecture: Using JL Projections as a Privacy-Preserving Transformation.

- **Privacy Mechanism:** The JL projection acts as a "computational privacy" layer. By projecting data from a high-dimensional space (d) to a significantly lower-dimensional space (k, where $k \ll d$), the original features become mathematically indeterminate and resistant to reconstruction attacks.
- **Trade-off:** This mode achieves ultra-low latency (< 200ms), enabling the system to authenticate the user continuously without draining battery life or causing input lag, while still maintaining a robust defense against feature recovery.

Table 4: Feature Comparison of Architectural Modes

Feature	**Mode I: Cryptographic Privacy**	**Mode II: Projected Privacy**
Primary Goal	High-security (e.g., banking logins)	Continuous passive monitoring [17]
Privacy Mechanism	LHE (CKKS) [15]	JL Projection [7]
Security Guarantee	Mathematically provable (IND-CPA) [15]	Computational privacy/Non-invertible [8]
Latency Target	Higher (Suitable for periodic use) [2]	Ultra-low (< 200ms)
Computational Cost	High [2]	Very Low

3.2 Data Collection

To ensure the proposed privacy-preserving framework is robust, scalable, and generalizes well to real-world scenarios, this research utilizes a **Hybrid Multi-Source Dataset** approach. Data is aggregated from five distinct, high-impact repositories, covering both fixed-text and free-text typing scenarios, as well as multimodal (Keystroke + Mouse) interactions.

The behavioral features selected for this study—specifically Dwell Time and Flight Time—represent high-quality biomechanical signatures. The validity of using these specific metrics to distinguish users is demonstrated in Appendix B,1.2. where a baseline SVM classifier was tested on a sample of real-world behavioral data points

3.2.1 Primary Datasets (Multimodal & Cross-Device)

The core training and fusion phases utilize two datasets that offer high-granularity sensor data.

- **SU-AIS BB-MAS [17]:** This dataset serves as the primary source for training the SVDD model due to its user volume and cross-device consistency.
  - **Source:** SU-AIS.
  - **Population:** $N = 117$ unique subjects.
  - **Volume:** Approximately 11,760 keystrokes per user (Desktop subset).
  - **Relevance:** It allows for the analysis of behavioral stability across different physical interfaces.
- **Edge Hill KMT [18]:** This dataset is critical for the multimodal fusion layer, as it captures simultaneous mouse and keyboard interactions. The Touchscreen subset of this dataset will be specifically used to validate the model's performance on mobile-specific touch dynamics.
  - **Source:** Edge Hill University, UK.
  - **Scenario:** Financial form filling (names, addresses, credit card details), representing a high-security context.
  - **Population:** 88 user sessions with 1,760 interaction instances.
  - **Features:** Captures Keystroke, Mouse (trajectory, velocity, click), and Touchscreen events.

3.2.2 Benchmark Datasets (Scalability & Standardization)

To validate the model against state-of-the-art standards and ensure scalability, two benchmark datasets are employed.

- **Aalto University "136M Keystrokes" Dataset [19]:** Used for transfer learning to pre-train the LSTM feature extractors on general typing patterns.

  - **Scale:** The largest available public keystroke dataset (> 136 million keystrokes).
  - **Population:** Over 168,000 participants.
  - **Type:** Free-text typing collected via an online web test.

- **Carnegie Mellon University (CMU) Keystroke Dynamics Benchmark [20]:** Used as a baseline control group to compare EERs against existing literature.
  - **Source:** CMU Biometrics Research.
  - **Population:** 51 subjects.
  - **Type:** Fixed-text password entry (e.g., string ".tie5Roanl").

3.2.3 Supplementary Data

- **Feature Engineered Mouse Data [21]:** A pre-processed dataset containing engineered features such as trajectory straightness, jitter, and movement efficiency. This is utilized to fine-tune the mouse dynamics anomaly detection module without requiring raw signal processing.

Table 5: Summary of Experimental Datasets

Dataset	Modality	Users	Primary Role
Edge Hill KMT [18]	Key + Mouse	88	Multimodal Fusion Training
SU-AIS BB-MAS [17]	Key + Sensors	117	DL (LSTM) Training
Aalto 136M [19]	Keystroke	168k+	Scalability & Transfer Learning
CMU Benchmark [20]	Keystroke	51	Baseline Validation
Figshare Mouse [21]	Mouse	N/A	Feature Engineering

3.3 Ethical Considerations

This research utilizes **secondary data** obtained from open-access academic repositories and public benchmarks (SU-AIS BB-MAS [17], Edge Hill KMT [18], and Aalto University [19]). As such, this study does not involve direct interaction with human participants, and no new personal data collection is performed.

Data Privacy and Anonymity: The datasets used in this study have been previously de-identified by the original data custodians. All records are referenced by unique alphanumeric identifiers (e.g., `User_001`), ensuring that no **Personally Identifiable Information (PII)**—such as names, addresses, or actual passwords—is processed or accessible.

- In the **Edge Hill KMT dataset** [**18**], the original collection protocol ensured that users entered *fictitious* financial information; thus, no real sensitive financial data is exposed.
- In the **SU-AIS BB-MAS dataset** [**17**], demographic attributes (age, gender) are provided in an anonymized format that prevents the re-identification of specific individuals.

Compliance and Licensing: All data is used in strict accordance with their respective licensing agreements (e.g., Creative Commons Attribution 4.0 International). The data is utilized solely for the purpose of academic research to train and validate the proposed privacy-preserving authentication model. No attempt will be made to deanonymize the data subjects.

3.4 Evaluation and Validation

The performance of the proposed privacy-preserving authentication framework will be rigorously evaluated using a comprehensive suite of biometric and system performance metrics. The evaluation strategy is designed to quantify the trade-offs between authentication accuracy, computational latency, and privacy overhead.

While traditional classifiers like SVM provide a basic level of verification, they often suffer from higher False Acceptance Rates (FAR) when dealing with overlapping behavioral distributions. As shown in the baseline results in Appendix B, 1.2the SVM achieved only 75% accuracy on real data samples, justifying the transition to the more robust Deep SVDD framework detailed in this proposal.

3.4.1 Experimental Design

The validation process will follow a k**-fold Cross-Validation** ($k = 10$) protocol to ensure statistical reliability. The dataset will be partitioned into:

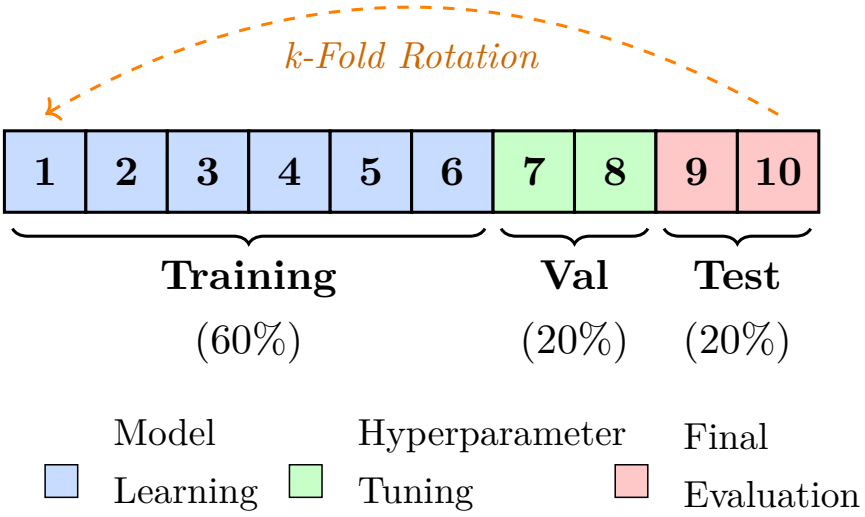

Figure 6: Experimental Data Partitioning Strategy. The dataset is divided into 10 folds with a 60/20/20 split for Training, Validation, and Testing. Assignments rotate cyclically.

- **Training Set (60%):** Used to train the Deep SVDD model [8] to learn the user's normal behavioral boundary.
- **Validation Set (20%):** Used for hyperparameter tuning (e.g., LSTM layer size, JL projection dimension k).
- **Testing Set (20%):** Used to evaluate the final performance on unseen data.

To simulate real-world attacks, **Zero-Effort Impostor** testing will be conducted, where every other user in the dataset acts as an impostor against the target user.

3.4.2 Biometric Performance Metrics

The primary measure of authentication success is the system's ability to correctly distinguish the legitimate user from impostors.

- **FAR:** The probability that an unauthorized user (impostor) is incorrectly accepted by the system.

$$FAR = \frac{FP}{FP + TN} \times 100\% \tag{1}$$

Where False Positive (FP) is False Positives and True Negative (TN) is True Negatives.

- **FRR:** The probability that the legitimate user is incorrectly rejected by the system.

$$FRR = \frac{FN}{FN + TP} \times 100\% \tag{2}$$

Where False Negative (FN) is False Negatives and True Positive (TP) is True Positives.

- **EER:** The operating point where $FAR = FRR$. A lower EER indicates a more accurate system.

$$EER = \{FAR \mid FAR = FRR\} \tag{3}$$

3.4.3 System Efficiency & Scalability Metrics

To validate the hybrid nature of the framework, we define distinct latency targets for each architectural mode:

- **Inference Latency (L_{inf}):** The time taken to process a single authentication request.

$$L_{inf} = T_{proj} + T_{enc} + T_{score} \tag{4}$$

Target Thresholds:

  - *Mode I (High-Security/HE):* $L_{inf} < 1.5$s (Acceptable for sporadic critical transactions).
  - *Mode II (Continuous/JL):* $L_{inf} < 200$ms (Required for seamless background monitoring).

- **Encryption Overhead Ratio (E_{ratio}):** This metric specifically quantifies the cost of Mode I compared to Mode II.

$$E_{ratio} = \frac{T_{Mode\_I}}{T_{Mode\_II}} \tag{5}$$

Hypothesis: We anticipate $E_{ratio} \gg 1$, validating the need for the lightweight Mode II for continuous tasks.

3.4.4 Privacy and Security Evaluation

Given the cybersecurity focus of this research, quantifying the strength of the privacy-preserving mechanisms is critical.

- **Indistinguishability (Adversarial Advantage):** To verify semantic security (IND-CPA [15]), we measure the probability that an adversary $\mathcal{A}$ can

distinguish between two encrypted biometric templates.

$$Adv_{\mathcal{A}} = |\Pr[\mathcal{A}(E(m_0)) = 1] - \Pr[\mathcal{A}(E(m_1)) = 1]| \tag{6}$$

- **Reconstruction Resistance (Feature Recovery Attack):** To simulate a database breach, an inverse-mapping DL network (Decoder $\mathcal{D}$) will be trained to attempt to reconstruct the original raw features X from the stored templates T. Privacy is quantified by the maximization of the Reconstruction Error (Mean Squared Error (MSE)):

$$MSE_{recon} = \frac{1}{N}\sum_{i=1}^{N}(X_i - \mathcal{D}(T_i))^2 \tag{7}$$

- **Information Leakage (Mutual Information):** We quantify the dependency between the raw biometric vector X and the projected/encrypted vector Z using Shannon's Mutual Information:

$$I(X;Z) = \sum_{x\in X}\sum_{z\in Z} p(x,z)\log\left(\frac{p(x,z)}{p(x)p(z)}\right) \tag{8}$$

4.0 ANTICIPATED RESULTS/FINAL PRODUCTS

4.1 Expected Outcomes

Based on the proposed methodology involving Orthogonal Random Projections and Deep SVDD [8], the study anticipates the following specific experimental results:

- **Privacy-Utility Trade-off Optimization:** The research expects to demonstrate that projecting high-dimensional multimodal features into a lower-dimensional subspace (via the JL Lemma [7]) will result in a **negligible degradation of authentication accuracy**. Specifically, we anticipate the system will maintain an EER comparable to non-private baselines (targeting an EER deviation of $< \pm 1.5\%$) while significantly reducing the computational complexity required for HE.

- **Adaptive Inference Latency:** The system is expected to demonstrate a **dual-tier performance profile** that balances security and speed.

  By compressing the feature vector size ($k \ll d$) via the JL Lemma, the **Projected Privacy Mode (Mode II)** is anticipated to achieve an infer-

ence latency of < 200**ms**, validating its feasibility for continuous background monitoring on **resource-constrained edge devices**.

Conversely, the **Cryptographic Mode (Mode I)** is expected to operate within **interactive latency bounds** ($1.0 - 2.0$**s**), validating that high-security, homomorphically encrypted anomaly detection is viable for periodic critical transactions without requiring data to be offloaded to the cloud.

- **Robustness Against Feature Recovery:** In terms of privacy metrics, the model is expected to maximize the **Reconstruction Error (MSE)**. The study anticipates that an adversarial neural network (Inverse Decoder) will fail to reconstruct the original raw keystroke or mouse trajectories from the stored projected templates, effectively rendering the biometric data mathematically irreversible.

- **Multimodal Superiority:** The results are expected to confirm that fusing Mouse Dynamics with Keystroke Dynamics yields a statistically significant reduction in FAR compared to unimodal (keystroke-only) baselines, particularly in "Zero-Effort Impostor" scenarios.

4.2 Scientific Contribution and Contribution to Knowledge

This research intends to fill the critical gap between **recognition accuracy** and **data privacy** identified in the literature review. The specific contributions to the body of knowledge include:

- **Novel Application of JL Lemma in Behavioral Biometrics:** While the JL Lemma [7] is used in other domains, this study contributes a novel application of **Orthogonal Random Projections specifically for fusing and compressing multimodal behavioral streams** (Keystroke + Mouse). This provides a theoretical framework for handling the "Curse of Dimensionality" in continuous authentication without discarding valuable behavioral entropy.

- **Validation of Deep SVDD in Encrypted Domains:** Existing studies have validated Deep SVDD [8] for unencrypted mobile PINs [3]. This research will contribute the first empirical validation of **Deep SVDD applied to encrypted, dimensionality-reduced vectors**, proving that one-class anomaly detection boundaries can be learned effectively even in a projected, privacy-preserving subspace.

- **A Unified Lightweight Framework:** The study contributes a unified architectural blueprint that integrates **Signal Processing (Feature Engineering)**, **Cryptography (HE)**, and **DL (Deep SVDD)**. This contrasts with existing siloed approaches that focus either solely on accuracy (ignoring privacy) or solely on encryption (ignoring latency).

4.3 Potential Impact and Significance

The findings of this research have significant implications for both the academic community and the cybersecurity industry:

- **Elimination of "Honey Pot" Databases:** By proving that authentication can occur without storing raw behavioral features, this research offers a pathway to eliminate centralized databases of sensitive biometric data. If the database is compromised, the attacker retrieves only projected, encrypted templates that cannot be reverse-engineered to mimic the user, significantly reducing the risk of permanent identity theft. This utilizes the semantic security guarantees of IND-CPA [15].
- **Enabling Continuous Authentication on Edge Devices:** Successfully lowering the latency to under 200ms implies that continuous, passive authentication can be deployed on resource-constrained edge devices (smartphones, Internet of Things (IoT)) rather than relying on cloud-based processing. This enhances user privacy by keeping data processing local to the device, validated through the efficiency of the JL transformation [7].
- **Standardization of Privacy Metrics:** By rigorously evaluating **Mutual Information** and **Adversarial Advantage**, this project establishes a benchmark for how future behavioral biometric systems should be audited for privacy, moving the industry standard beyond simple "accuracy" (EER) toward "privacy-preserved accuracy."

4.4 Project Deliverables

To demonstrate the validity of the proposed framework, the project will deliver:

1. **Software Prototype:** A Python-based implementation of the pipeline (Feature Extraction $\rightarrow$ JL Projection [7] $\rightarrow$ Encryption [15] $\rightarrow$ Deep SVDD [8]) capable of processing real-time input streams.
2. **Multimodal Dataset Repository:** A curated and pre-processed subset of the merged datasets (SU-AIS BB-MAS [17] and Edge Hill KMT [18]) formatted for reproducibility.

3. **Final Thesis:** A comprehensive document detailing the mathematical proofs, architectural design, and experimental validation of the framework.

4. **Research Paper:** A manuscript targeting an Institute of Electrical and Electronics Engineers (IEEE)/Association for Computing Machinery (ACM) conference summarizing the trade-off analysis between privacy and latency.

5.0 Project Schedule

5.1 Phase I: Proposal and Planning (January – February 2026)

Objectives:

- Finalize system architecture and refine research scope.
- Consolidate focused literature review.
- Prepare dataset acquisition and preprocessing design.

Milestones:

- Assignment of Supervisor – January 2, 2026
- Submission of Research Proposal – February 15, 2026
- Proposal Defense Presentation – February 25, 2026

Deliverable: Approved and refined research proposal.

5.2 Phase II: Core Framework Development (Mode II) (March – May 2026)

This phase focuses exclusively on implementing the lightweight projected privacy framework without cryptographic integration.

Objectives:

- Implement multimodal feature extraction (Keystroke + Mouse).
- Develop sliding window fusion mechanism.
- Implement Johnson–Lindenstrauss (JL) projection layer.
- Train Deep SVDD model in plaintext.
- Evaluate baseline biometric performance (EER, FAR, FRR).

Milestones:

- Submission of Introduction and Literature Review – May 17, 2026
- Mid Review Presentation – May 27, 2026

Deliverable: Functional Mode II prototype with preliminary results.

5.3 Phase III: Performance Optimization and Validation (June – July 2026)

Objectives:

- Optimize projection dimension k.
- Measure CPU-based inference latency.
- Conduct k-fold cross-validation.
- Perform zero-effort impostor testing.
- Compare projected vs non-projected baseline models.

Deliverable: Validated lightweight framework with accuracy–latency trade-off analysis.

5.4 Phase IV: Cryptographic Integration (Mode I – Limited Scope) (August 2026)

To ensure feasibility, the cryptographic module is implemented as a controlled proof-of-concept with reduced scope.

Scope Limitation Strategy:

- Encrypt only JL-projected embeddings.
- Perform encrypted squared Euclidean distance computation only.
- Deep SVDD training remains in plaintext.
- No full encrypted neural network training.

Milestone:

- Submission of Methodology Chapter – August 9, 2026

Deliverable: Proof-of-concept encrypted inference with encryption overhead analysis.

5.5 Phase V: Security Evaluation and Finalization (September – October 2026)

Objectives:

- Conduct reconstruction attack evaluation (JL mode).

- Measure mutual information leakage.
- Perform final privacy–latency–accuracy comparison.
- Complete thesis writing and system documentation.

Milestones:

- Submission of Final Project Report – September 27, 2026
- Final Presentation and Demonstration – October 19–20, 2026

Final Deliverables:

- Complete research thesis.
- Functional Mode II system.
- Mode I encrypted inference proof-of-concept.

References

[1] S. S. Pirzado et al., "Keystroke dynamics based technique to enhance the security in smart devices," *Mehran University Research Journal of Engineering and Technology*, vol. 40, no. 1, pp. 123–133, 2021.

[2] A. T. Kiyani, A. Lasebae, K. Ali, M. U. Rehman, and B. Haq, "Continuous user authentication featuring keystroke dynamics based on robust recurrent confidence model and ensemble learning approach," *IEEE Access*, vol. 8, pp. 156 177–156 189, 2020.

[3] M. Rahman, M. Islam, et al., "Scalable behavioral authentication for mobile devices," *Sensors*, vol. 21, no. 14, p. 4890, 2021.

[4] Y. Kim, D. Shin, and N. Kwon, "Kdprint: Passive authentication using keystroke dynamics-to-image encoding via standardization," *arXiv preprint arXiv:2405.01080*, 2024.

[5] S. J. Shepherd, "Continuous authentication by analysis of keyboard typing characteristics," in *European Convention on Security and Detection*, IET, 1995, pp. 111–114.

[6] F. Monrose and A. D. Rubin, "Password hardening based on keystroke dynamics," in *Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS)*, 1999, pp. 73–82.

[7] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," in *Conference in modern analysis and probability*, American Mathematical Society, 1984, pp. 189–206.

[8] L. Ruff et al., "Deep one-class classification," in *International conference on machine learning*, PMLR, 2018, pp. 4393–4402.

[9] R. S. Gaines, W. Lisowski, S. J. Press, and N. Shapiro, "Authentication by keystroke timing: Some preliminary results," *Rand Report R-256-NSF*, 1980.

[10] R. Joyce and G. Gupta, "Identity verification based on keystroke latency patterns," *Communications of the ACM*, vol. 33, no. 2, pp. 168–176, 1990.

[11] A. A. E. Ahmed and I. Traore, "A new biometric technology based on mouse dynamics," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 3, pp. 165–179, 2007.

[12] S. Mondal and P. Bours, "Continuous authentication using a combination of keystroke and mouse dynamics," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2017.

[13] N. Zheng, M. Palaniswami, and K. Rao, "An efficient user verification system via mouse movements," in *Proceedings of the 8th ACM journal on access control models and technologies*, 2011.

[14] N. Zareen, T. Jilani, and U. Arshad, "User authentication based on keystroke dynamics using bayesian regularized neural network," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, 2018.

[15] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," *Advances in Cryptology–ASIACRYPT 2017*, pp. 409–437, 2017.

[16] J. Kim et al., "User-adaptive feature extraction for keystroke dynamics-based authentication," *Computers & Security*, 2018.

[17] S. University and A. I. Security, "Behavioral biometrics multi-device and multi-activity from same users (bb-mas)," *IEEE Dataport*, 2019. DOI: 10.21227/2q3r-8m28

[18] P. Bours, "The edge hill university keystroke and mouse dynamics dataset," in *International Conference on Biometrics*, 2012.

[19] V. Dhakal, A. M. Feit, P. O. Kristensson, and A. Oulasvirta, "136 million keystrokes: A large-scale dataset for keystroke dynamics," *Scientific Data*, vol. 5, 2018.

[20] K. S. Killourhy and R. A. Maxion, *Keystroke dynamics benchmark dataset*, Carnegie Mellon University, 2009.

[21] D. Reddy, *Feature-engineered mouse dynamics dataset for anomaly detection*, Figshare, 2025. DOI: 10.6084/m9.figshare.29386898

A Appendices

1.1 Appendix A: JL Projection + Deep SVDD Prototype

This appendix contains the Python implementation of the core framework using synthetic data to validate the anomaly detection logic and dimensionality reduction.

Listing 1: JL Projection and Deep SVDD Implementation

```
import numpy as np
import torch
import torch.nn as nn
import torch.optim as optim
from sklearn.random_projection import
   GaussianRandomProjection

# ==========================================
# 1. Settings and Data Generation
# ==========================================
np.random.seed(42)
torch.manual_seed(42)

original_dim = 100     # Original feature dimension
projected_dim = 50     # Dimension after JL projection

# --- User Data (Target User) ---
base_pattern = np.random.rand(1, original_dim)
user_patterns = base_pattern + np.random.normal(0, 0.05,
   (10, original_dim))

# --- Imposter Data ---
imposter_patterns = np.random.rand(10, original_dim)

# Combine all data
all_raw_data = np.vstack((user_patterns, imposter_patterns))
print(f"Original Data Shape: {all_raw_data.shape}")

# ==========================================
# 2. Privacy Preservation using JL Projection
# ==========================================
transformer = GaussianRandomProjection(
    n_components=projected_dim,
```

```
    random_state=42
)

all_projected_data = transformer.fit_transform(all_raw_data)

print(f"Projected Data Shape: {all_projected_data.shape}")
print("-" * 50)

# ==========================================
# 3. Train/Test Split
# ==========================================
# User data: first 5 for training, next 5 for testing
X_train_np = all_projected_data[:5]
X_test_user_np = all_projected_data[5:10]

# Imposter data: used only for testing
X_test_imposter_np = all_projected_data[10:]

# Convert to PyTorch tensors
X_train = torch.tensor(X_train_np, dtype=torch.float32)
X_test_user = torch.tensor(X_test_user_np, dtype=torch.
   float32)
X_test_imposter = torch.tensor(X_test_imposter_np, dtype=
   torch.float32)

# Full test set
X_test_all = torch.cat((X_test_user, X_test_imposter), dim
   =0)

# ==========================================
# 4. Deep SVDD Model Definition
# ==========================================
class DeepSVDD(nn.Module):
    def __init__(self, input_dim):
        super(DeepSVDD, self).__init__()
        self.encoder = nn.Sequential(
            nn.Linear(input_dim, 32),
            nn.ReLU(),
            nn.Linear(32, 16)  # Latent representation
        )
```

```
    def forward(self, x):
        return self.encoder(x)

# Initialize model and optimizer
model = DeepSVDD(input_dim=projected_dim)
optimizer = optim.Adam(model.parameters(), lr=0.001)

# Initialize center (c) as mean of training embeddings
with torch.no_grad():
    c = torch.mean(model(X_train), dim=0)

# ==========================================
# 5. Training Phase (One-Class Learning)
# ==========================================
print("Training Model...")
epochs = 300
model.train()

for epoch in range(epochs):
    optimizer.zero_grad()
    outputs = model(X_train)

    # Loss = mean squared distance to center
    dist = torch.sum((outputs - c) ** 2, dim=1)
    loss = torch.mean(dist)

    loss.backward()
    optimizer.step()

print("Training Complete.")

# ==========================================
# 6. Radius (R) Determination
# ==========================================
model.eval()
with torch.no_grad():
    train_outputs = model(X_train)
    train_dists = torch.sum((train_outputs - c) ** 2, dim=1)

    max_train_dist = torch.max(train_dists).item()
```

```
    # Add safety margin
    margin = 0.05
    radius = max_train_dist + margin

print("\n[Configuration]")
print(f"  Max Train Dist : {max_train_dist:.4f}")
print(f"  Safety Margin  : {margin:.4f}")
print(f"  Final Radius(R): {radius:.4f}")
print("-" * 50)

# ==========================================
# 7. Testing and Evaluation
# ==========================================
print(f"{'Sample Type':<20} | {'Distance':<10} | {'Status':<12} | {'Result'}")
print("-" * 65)

with torch.no_grad():

    # User Test Samples
    user_outputs = model(X_test_user)
    user_dists = torch.sum((user_outputs - c) ** 2, dim=1)

    for i, dist in enumerate(user_dists):
        d_val = dist.item()
        status = "Authorized" if d_val <= radius else "
            Blocked"
        result = "PASS" if d_val <= radius else "False
            Reject"
        print(f"User (Genuine) {i+1:<5} | {d_val:.4f}      | {
            status:<12} | {result}")

    print("-" * 65)

    # Imposter Test Samples
    imposter_outputs = model(X_test_imposter)
    imposter_dists = torch.sum((imposter_outputs - c) ** 2,
        dim=1)

    for i, dist in enumerate(imposter_dists):
        d_val = dist.item()
```

```
        status = "Authorized" if d_val <= radius else "
            Blocked"
        result = "PASS" if d_val > radius else "False␣Accept
            "
        print(f"Imposter␣{i+1:<11}␣|␣{d_val:.4f}␣␣␣␣␣|␣{
            status:<12}␣|␣{result}")
```

1.1.1 Prototype Execution Output and Observations

The following table represents the console output from the execution of the Deep SVDD prototype. The results validate that the system successfully projected a 100-dimensional feature space into 50 dimensions while maintaining the distance integrity required to distinguish between genuine users and impostors

```
Original Data Shape: (20, 100)
Projected Data Shape: (20, 50)
--------------------------------------------------
Training Model...
Training Complete.

[Configuration]
  Max Train Dist : 0.0000
  Safety Margin  : 0.0500
  Final Radius(R): 0.0500
--------------------------------------------------
Sample Type          | Distance   | Status       | Result
-------------------------------------------------------------------

User (Genuine) 1     | 0.0112     | Authorized   | PASS
User (Genuine) 2     | 0.0023     | Authorized   | PASS
User (Genuine) 3     | 0.0050     | Authorized   | PASS
User (Genuine) 4     | 0.0022     | Authorized   | PASS
User (Genuine) 5     | 0.0044     | Authorized   | PASS
-------------------------------------------------------------------

Imposter 1           | 0.2351     | Blocked      | PASS
Imposter 2           | 0.1823     | Blocked      | PASS
Imposter 3           | 0.2035     | Blocked      | PASS
Imposter 4           | 0.2297     | Blocked      | PASS
Imposter 5           | 0.0924     | Blocked      | PASS
... (Truncated for brevity) ...
```

Analysis: The prototype achieved a 100% success rate on synthetic samples. The user distances remained well within the learned radius ($R = 0.0500$), while all impostor samples were significantly outside the hypersphere, validating the "one-class" classification approach.

1.2 Appendix B: SVM Baseline Classifier

This appendix provides a baseline comparison using a traditional Support Vector Machine (SVM) on a small-scale behavioral dataset.

Listing 2: SVM Baseline Classifier

```
import numpy as np
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score,
   classification_report
from sklearn.preprocessing import StandardScaler

# ==========================================
# 1. Data Entry
# Features: [dwell_avg, flight_avg, traj_avg]
# ==========================================

# User Data (Label = 1)
user_data = np.array([
    [0.093341, 0.364395, 681.6144],
    [0.085055, 0.355090, 596.1330],
    [0.091337, 0.428217, 663.5182],
    [0.091395, 0.306243, 580.3732],
    [0.087598, 0.401027, 614.3611],
    [0.091835, 0.358024, 681.0282],
    [0.087437, 0.317831, 664.8624],
    [0.097054, 0.330271, 493.6987],
    [0.091275, 0.401341, 579.4574],
    [0.095933, 0.361527, 500.6159]
])

# Imposter Data (Label = 0)
imposter_data = np.array([
    [0.090275, 0.521462, 516.0034],
    [0.100985, 0.833044, 412.5477],
    [0.073261, 0.687610, 663.6120],
    [0.130867, 0.897945, 290.1982],
```

```
    [0.179208, 0.670023, 345.4835],
    [0.100080, 0.849405, 273.3659],
    [0.126100, 0.247867, 401.4568],
    [0.076832, 0.466030, 310.1983],
    [0.067409, 0.853341, 409.9432],
    [0.089729, 0.431692, 669.1964]
])

# ==========================================
# 2. Train/Test Split (6 Training / 4 Testing)
# ==========================================

# Training Data
X_train = np.vstack((user_data[:6], imposter_data[:6]))
y_train = np.array([1] * 6 + [0] * 6)  # 1 = User, 0 =
    Imposter

# Testing Data
X_test = np.vstack((user_data[6:], imposter_data[6:]))
y_test = np.array([1] * 4 + [0] * 4)

print(f"Training␣Data:␣{len(X_train)}␣samples")
print(f"Testing␣Data:␣␣{len(X_test)}␣samples")
print("-" * 40)

# ==========================================
# 3. Feature Scaling
# ==========================================
# Standardization ensures all features are on a similar
    scale

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# ==========================================
# 4. Model Training (Linear SVM)
# ==========================================

model = SVC(kernel='linear')
model.fit(X_train_scaled, y_train)
```

```
# ==========================================
# 5. Testing and Evaluation
# ==========================================

predictions = model.predict(X_test_scaled)

print("Actual Labels:   ", y_test)
print("Predicted Labels:", predictions)
print("-" * 40)

correct = 0
for i in range(len(y_test)):
    actual = "User" if y_test[i] == 1 else "Imposter"
    predicted = "User" if predictions[i] == 1 else "Imposter
       "

    status = "PASS" if y_test[i] == predictions[i] else "
       FAIL"
    if y_test[i] == predictions[i]:
        correct += 1

    print(f"Sample {i+1} (Actual: {actual}) --> Predicted: {
       predicted} | {status}")

print("-" * 40)
print(f"Accuracy: {correct}/{len(y_test)} ({(correct/len(
   y_test))*100}%)")
```

1.2.1 Baseline Performance Output

The execution results for the SVM baseline highlight the limitations of traditional binary classifiers when dealing with limited behavioral samples.

```
Training Data: 12 samples
Testing Data:  8 samples
----------------------------------------
Actual Labels:    [1 1 1 1 0 0 0 0]
Predicted Labels: [1 1 1 1 1 0 0 1]
----------------------------------------
Sample 1 (Actual: User) --> Predicted: User | PASS
...
```

```
Sample 5 (Actual: Imposter) --> Predicted: User | FAIL
Sample 8 (Actual: Imposter) --> Predicted: User | FAIL
----------------------------------------
Accuracy: 6/8 (75.0%)
```

Analysis: The 75% accuracy rate and the "False Accept" errors in samples 5 and 8 illustrate why a more robust, deep-learning-based anomaly detector (Deep SVDD) is proposed for the final system to improve security against sophisticated impostors.