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

BB-MAS Behavioral Biometrics Multi-device and multi-Activity from Same users

CMU Carnegie Mellon University

Deep SVDD Deep Support Vector Data Description

DL Deep Learning

EER Equal Error Rate

FAR False Acceptance Rate

FN False Negative

FP False Positive

FRR False Rejection Rate

HE Homomorphic Encryption

IoT Internet of Things

JL Johnson-Lindenstrauss

KMT Keystroke and Mouse Trajectory

LSTM Long Short-Term Memory

MSE Mean Squared Error

PIN Personal Identification Number

ReLU Rectified Linear Unit

RNN Recurrent Neural Network

SMPC Secure Multi-Party Computation

SU-AIS Syracuse University and Assured Information Security

SVM Support Vector Machine

TN True Negative

TP True Positive

1.0 Introduction

Overview of the Research Domain The more conventional knowledge-based authentication techniques, such as passwords and Personal Identification Number (PIN)s, are quickly turning into single points of failure in the constantly changing cybersecurity landscape of today. Well-known assaults like shoulder surfing, brute-force attacks, and social engineering can affect these static authentication systems.

Behavioral Biometrics, more especially Keystroke Dynamics [1], [2] and Mouse Dynamics [3], has become a strong alternative authentication technique for confirming digital identification in response to the drawbacks and weaknesses of static authentication approaches. Behavioral biometrics enable continuous and passive user authentication based on a user's unique typing patterns or mouse movement trajectories, as opposed to static passwords, which authenticate what a user knows [4], [5].

Motivation and Significance The use of behavioral biometrics creates a significant privacy conundrum even if they provide strong defense against unwanted access. In contrast to passwords, biometric characteristics are inherent and unchangeable; if the biometric template is compromised, a user cannot alter their hand geometry or typing rhythm [6], [7]. The storing of raw templates in a central database is therefore a high-risk liability since a breach of raw behavioral data results in a permanent loss of digital identity [7], [8].

The state-of-the-art solutions to this privacy issue involve the use of heavy cryptographic techniques, including Homomorphic Encryption (HE) [9], which enables computations to be performed on the encrypted data. Nevertheless, these approaches involve unaffordable computational costs and high system latency, making them unsuitable for real-time, continuous monitoring in resource-constrained Internet of Things (IoT) or edge devices [10]. This gives rise to a substantial trade-off, whereby the current solutions are either fast and privacy-invasive (using raw data) or private but slow (using heavy encryption).

Existence of a Research Gap Unified frameworks that successfully strike a compromise between robust template privacy and high-speed anomaly detection are currently lacking [11]. Current "lightweight" methods frequently ignore the requirement for biometric revocability—the capacity to cancel and replace a compromised biometric template—in favor of concentrating only on recognition accuracy [7], [12]. In order to fill this gap, this study suggests a brand-new, lightweight framework that combines Deep SVDD [13] with Keyed JL Projections

[14]. This method seeks to ensure a safe and smooth user experience by offering mathematically guaranteed privacy and template revocability without the latency costs of conventional encryption.

1.1 Problem Statement

The most important trade-off between user privacy, system latency, and template revocability in continuous behavioral authentication systems is the problem that this research paper tries to solve. Although behavioral biometrics, such as Keystroke Dynamics and Keystroke and Mouse Trajectory (KMT), are a trustworthy method for continuously authenticating a user's identity, there are severe security risks linked to the processing and storage of these behavioral patterns. The problem with behavioral biometrics is that they are inherent and immutable, unlike passwords or tokens; if the biometric template is compromised, an individual cannot change their hand shape or typing rhythm [6], [7]. Thus, biometric databases are a prime target for attackers because, once the raw behavioral data is compromised, an individual loses their digital identity permanently [7], [8].

Current approaches fail to address this problem effectively due to a technical dichotomy between security and performance:

- **Privacy Gaps in High-Accuracy Models:** The most advanced deep learning frameworks, like those based on Recurrent Neural Networks (RNNs) [11] or image-based encoding [15], provide a high level of recognition accuracy but are normally associated with the need to store the original or slightly processed behavioral features. This introduces a weakness in which the templates are prone to reverse engineering or replay attacks.

- **Efficiency Gaps in Cryptographic Methods:** Effective cryptographic tools, like HE [9], enable computations to be performed on encrypted information in a secure manner. Nevertheless, these methods impose unacceptably high computational costs and delays, making them less feasible for real-time monitoring applications on edge devices [10].

- **Lack of Integrated Revocability:** Most of the existing frameworks are concerned with the accuracy of verification results but do not have a way to make biometrics "cancelable" [7]. If a template is compromised, there is no existing lightweight implementation that would enable a user to "reset" their biometric identity without either a complete new registration or a complete loss of privacy.

There is no common framework yet that can efficiently address these competing demands. This research aims to fill this gap by developing an architecture that employs Keyed JL Projections [14], [16] for lightweight and non-invertible feature transformation and Deep SVDD [13] for efficient server-side anomaly detection. This will ensure mathematical irreversibility and template revocability with sub-second latency, which is necessary for continuous authentication.

1.2 Research Aim and Objectives

The main objective of this research work is to design a lightweight and privacy-friendly framework for continuous behavioral biometric authentication that can efficiently address the trade-off between recognition performance, system delay, and template revocability. This framework uses Keyed JL Projections [14], [16] to provide the mathematical irreversibility of biometric templates and Deep SVDD [13] for one-class anomaly detection.

To achieve this aim, the following specific objectives have been identified:

1. **To design a privacy-preserving feature transformation pipeline:** Develop a mechanism using Keyed-JL Projections [14], [17] that transforms high-dimensional multimodal features (keystroke and mouse) into a lower-dimensional subspace, ensuring templates are mathematically irreversible and revocable through key renewal.

2. **To implement a multimodal behavioral fusion layer:** Combine independent keystroke dynamics, such as dwell time and flight time, and mouse dynamics, such as velocity and curvature, into coordinated behavioral units (KMT) [5] to raise the entropy of the user profile.

3. **To develop a lightweight server-side anomaly detection model:** Implement a Deep SVDD [13] classifier capable of distinguishing between legitimate users and impostors by learning a compact hypersphere boundary around the projected privacy-preserving embeddings.

4. **To evaluate the trade-off between privacy, accuracy, and latency:** Quantify the impact of projection dimension k on authentication performance (Equal Error Rate (EER)), system latency (targeting < 200ms), and resistance to feature recovery attacks.

1.3 Research Questions

In order to solve the problem identified and meet the research objectives, the following research questions have been developed. These research questions frame the study on the viability and performance of the proposed privacy-preserving behavioral authentication framework:

- **RQ1:** To what extent can Keyed-JL Projections preserve the uniqueness of multimodal behavioral biometrics while ensuring the mathematical irreversibility of the stored templates [7], [17]?
- **RQ2:** What is the optimal projection dimension k that minimizes the EER of the Deep SVDD model [13] without exceeding the computational latency constraints of real-time monitoring?
- **RQ3:** How effectively does the fusion of keystroke and mouse dynamics (KMT) [5] mitigate the accuracy degradation typically associated with privacy-preserving feature transformations?
- **RQ4:** How resilient is the Keyed-JL transformed template against feature recovery and replay attacks compared to traditional Deep Learning (DL)-based authentication models [8], [16]?

2.0 Background

2.1 Theoretical and Technical Background

The proposed research combines ideas from high-dimensional geometry, privacy-preserving machine learning, and multimodal behavioral biometrics. This section describes the mathematical and architectural foundations necessary to comprehend the Keyed-JL and Deep SVDD framework [13], [14].

2.1.1 The Johnson-Lindenstrauss (JL) Lemma

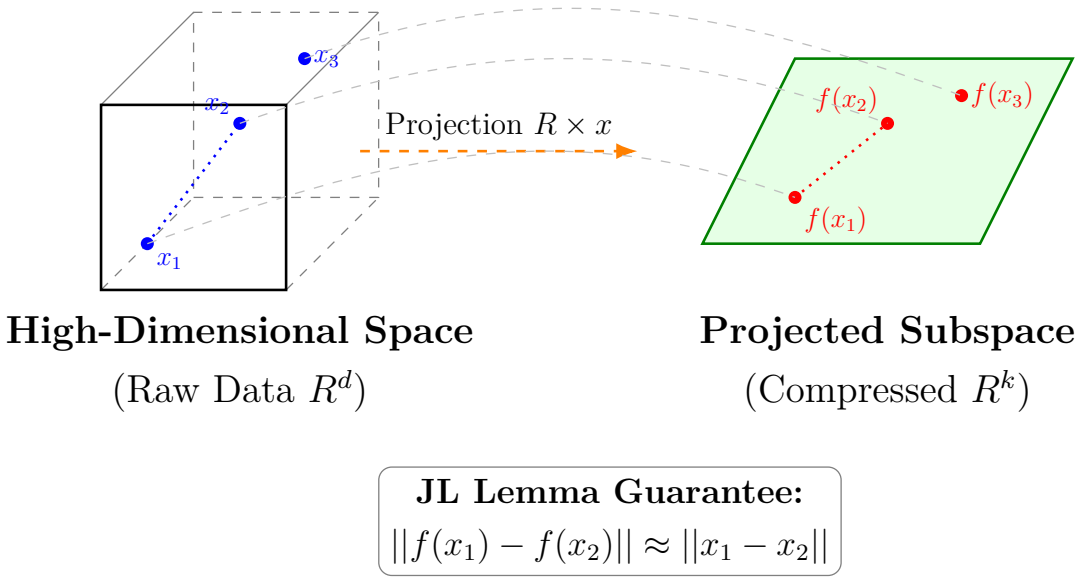


Figure 1: Conceptual Visualization of the JL Lemma [14]. Points from a high-dimensional feature space (Left) are projected onto a lower-dimensional subspace (Right). The blue and red dotted lines illustrate that the relative Euclidean distances between points are approximately preserved despite the massive reduction in dimensions.

The foundational principle for the privacy-preserving component of this research is the Johnson-Lindenstrauss Lemma. The lemma states that a set of n points in a high-dimensional space can be projected into a lower-dimensional space of dimension k while nearly preserving the pairwise distances between all points.

For any $0 < \epsilon < 1$, there exists a linear mapping $f : \mathbb{R}^d \to \mathbb{R}^k$ such that for all $u, v \in X$:

$$(1 - \epsilon)\|u - v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1 + \epsilon)\|u - v\|^2 \tag{1}$$

Where ϵ represents the error tolerance and f is the projection mapping.

In this research, the projection matrix is generated using a secret key, ensuring that the transformation is non-invertible and providing revocability.

2.1.2 Deep Support Vector Data Description (Deep SVDD)

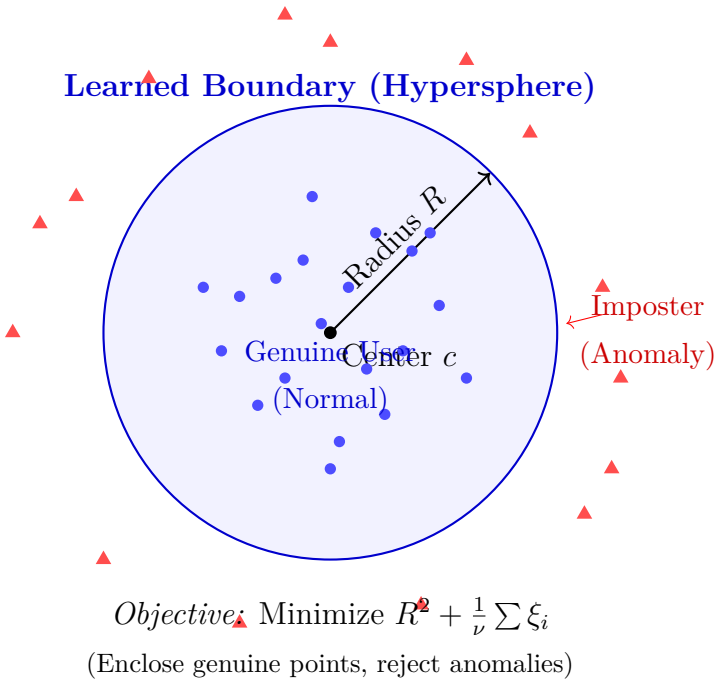


Figure 2: Visualizing the Deep SVDD Decision Boundary. The model learns a compact hypersphere of radius R around center c that encapsulates the majority of legitimate user data (blue dots). Any input falling outside this boundary is flagged as an anomaly (red triangles), representing a potential impostor.

The authentication engine utilizes Deep SVDD [13], an unsupervised method for one-class classification. It maps legitimate user features into a minimum-volume hypersphere in a latent space, optimized to minimize the Mean Squared Error (MSE) between the transformed features and the hypersphere center.

The objective function minimizes the distance of the network outputs to the center c of the hypersphere:

$$\min_{\mathcal{W}} \sum_{i=1}^{n} \|\phi(x_i; \mathcal{W}) - c\|^2 + \lambda \|\mathcal{W}\|_F^2 \tag{2}$$

Where $\phi(x_i; \mathcal{W})$ is the neural network mapping, c is the hypersphere center, and $\lambda\|\mathcal{W}\|_F^2$ is the weight decay regularization term.

2.1.3 Multimodal Behavioral Fusion

The system architecture relies on the fusion of keystroke timing and mouse movement patterns. This multimodal approach increases the entropy of the biometric profile, compensating for the information loss during privacy-preserving projection.

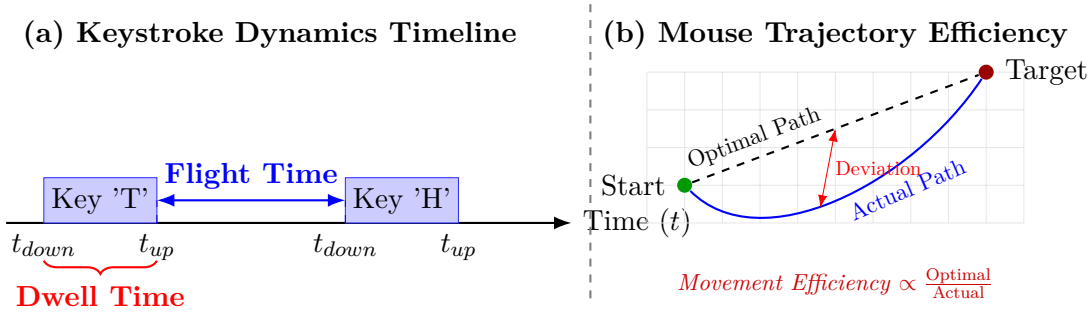


Figure 3: Visual Representation of Biometric Features. **(a)** Keystroke Dynamics: Dwell Time represents the key press duration, while Flight Time measures the latency between distinct keystrokes. **(b)** Mouse Dynamics: Comparison between the optimal straight-line path and the actual user trajectory, used to calculate movement efficiency and curvature.

2.2 Literature Review

The development of behavioral authentication has moved from basic timing analysis [1] to intricate multimodal DL models [11]. This section groups existing work into major subfields relevant to the proposed Keyed-JL and Deep SVDD framework [13].

2.2.1 Foundational Keystroke Dynamics

Early work proved the concept of biometric authentication using typing patterns. Key studies by Gaines et al. [1] showed that keystroke timing, namely dwell time and flight time, is distinct to each user. Although early biometric systems were based on static login authentication [2], later research by Shepherd [4] introduced continuous authentication, enabling ongoing monitoring of sessions. Monrose and Rubin [6] later showed the value of these patterns in securing traditional password-based systems.

2.2.2 Multimodal Fusion and Mouse Dynamics

To enhance the robustness of recognition, researchers combined mouse dynamics and fusion methods. Ahmed and Traore [3] demonstrated that mouse movement trajectories and click rates form a unique behavioral pattern. Mondal and Bours [5]

Table 1: Summary of Key Related Studies

Reference & Study	**Key Contribution to This Project**	**Identified Gap / Limitation**
Gaines et al. (1980) [1] & Joyce et al. (1990) [2]	**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) [5]	**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) [8]	**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) [15]	**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) [11]	**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) [10]	**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).

demonstrated that a multimodal system (KMT), combining keystroke and mouse information, leads to a substantial improvement in the EER compared to unimodal systems. More recent challenges, such as those proposed by Reddy [18], focus on mouse dynamics based on feature engineering for anomaly detection.

2.2.3 Deep Learning and Privacy Methodologies

Contemporary studies apply deep learning but point out the increasing privacy issues. High-accuracy models based on RNNs [11] and Bayesian neural networks [19] have become common practice for behavior modeling. Nevertheless, as demonstrated in the KDPrint framework by Kim et al. [15], the representation of behavioral data in the form of standardized images can make templates susceptible to attacks if they are not appropriately protected. Although HE [9] provides a mathematical remedy for privacy, its latency may be beyond the needs of continuous monitoring [10]. This study fills the gap by integrating the privacy of Keyed-JL projections with the efficiency of Deep SVDD [13].

2.3 Research Gap

Despite advancements in behavioral biometrics, a critical gap exists in providing simultaneous mathematical privacy, low latency, and template revocability [7].

2.3.1 Technical Dichotomy: Privacy vs. Performance

There is a large gap between highly accurate DL models and privacy-preserving cryptographic approaches. Traditional models retain reversible templates [8], whereas HE [9] imposes unacceptably high latencies (typically > 1.5s) that are not applicable for continuous observation on edge devices [10].

2.3.2 Vulnerability of Static Biometric Templates

There is no standardized way for "cancelable" behavioral biometrics in current literature. Unlike traditional authentication methods, behavioral traits are non-reversible; if a stored template is breached, the user's digital identity is put at risk because there is no lightweight method to re-issue a new, non-correlated behavioral profile [6], [7].

2.3.3 Limitations of Unimodal Authentication

The majority of privacy-preserving transformations lead to a certain level of information loss. The current state of unimodal studies (either keystroke or mouse) [1], [3] tends to neglect the recognition accuracy while applying strong

dimensionality reduction. There is a research gap concerning the compensation of entropy loss due to privacy-preserving transformations by multimodal fusion (KMT) [5].

2.3.4 Gap Definition

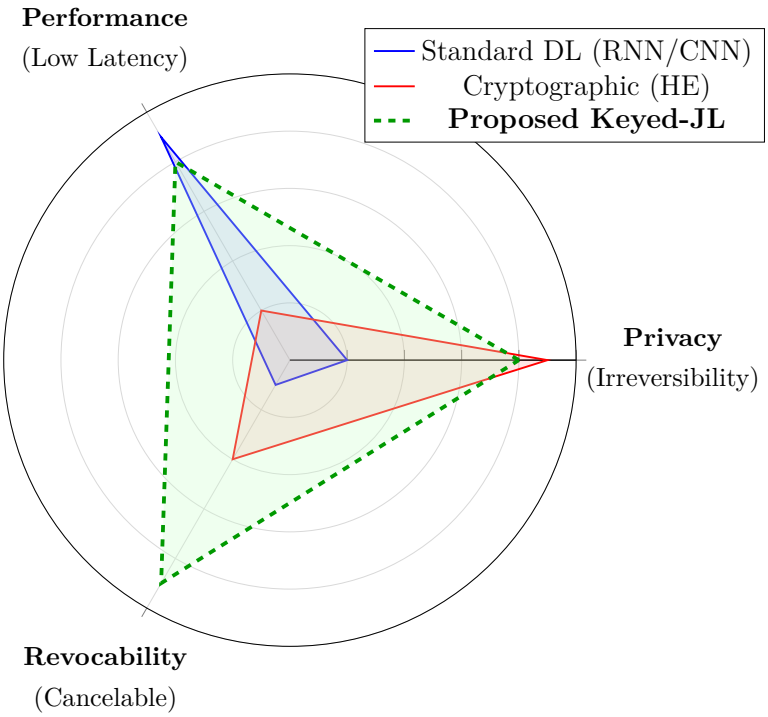


Figure 4: The Technical Dichotomy in Behavioral Authentication.

The research gap is formally defined as the lack of a unified behavioral authentication framework that provides **provable non-invertibility** without sacrificing the **sub-second latency** required for continuous monitoring. While existing studies achieve high accuracy using DL [11], [15], they fail to provide a mechanism for template revocability, meaning a compromised biometric profile is permanently lost [7]. Conversely, cryptographic methods like HE [9] that provide privacy are too computationally expensive for real-time use on standard hardware [10].

2.4 Assumptions and Constraints

This section outlines the foundational assumptions made during the research design and the technical constraints that bound the scope of the proposed privacy-preserving behavioral authentication system.

2.4.1 Assumptions

The validity of the experimental results and the effectiveness of the framework rely on several key assumptions:

- **Consistency of User Behavior:** It is assumed that the primary user's typing and mouse usage patterns remain relatively stable over the duration of the study [8].
- **Integrity of Enrollment:** The research assumes the initial enrollment occurs in a secure environment, ensuring the Deep SVDD [13] model is trained on verified legitimate data.
- **Security of the Secret Key:** The mathematical irreversibility of the JL-projection relies on the secret key being stored securely on the client side [17].
- **Hardware Sufficiency:** It is assumed that the target hardware can perform the linear JL-projections and inference within the required latency bounds [10].

2.4.2 Constraints and Limitations

The study is conducted within the following constraints:

- **Modality Limitation:** The system is restricted to desktop inputs (KMT) and does not include mobile-specific biometrics.
- **Data Availability:** Evaluation is limited to the diversity and volume of data present in selected benchmark datasets such as Behavioral Biometrics Multi-device and multi-Activity from Same users (BB-MAS) [20] and the Carnegie Mellon University (CMU) dataset [21].
- **Latency vs. Privacy Bound:** A critical constraint is the trade-off between projection dimension k and system performance, targeting a real-time latency of < 200ms.
- **Environmental Noise:** Variations in hardware peripherals may introduce noise, potentially affecting the overall EER.

3.0 Methodology and Project Design

The proposed methodology follows a quantitative experimental approach to assess the performance of privacy-preserving continuous authentication. The research design is based on the "Privacy-by-Design" principle [7], which ensures that the behavioral templates (KMT) are never shared or stored in an invertible form. This is achieved by applying Keyed-JL projections at the edge before transmitting features to the Deep SVDD [13] backend for one-class classification.

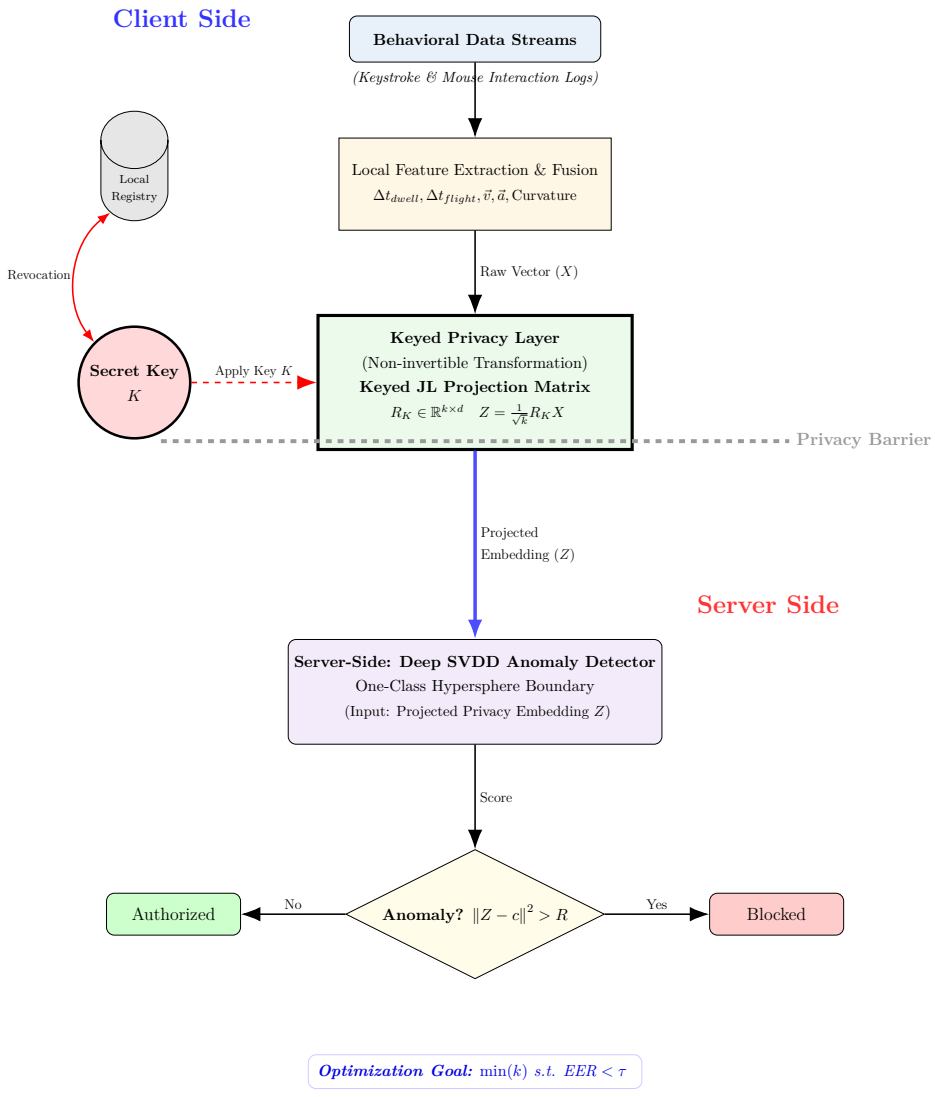


Figure 5: System Architecture: Keyed-JL Privacy and Server-Side Deep SVDD Detection

3.1 Overview of the Proposed Research Design

The system architecture is divided into two distinct environments separated by a **Privacy Barrier**: the **Client Side**, where data is captured and transformed

using Keyed-JL projections, and the **Server Side**, where authentication decisions are made. This design ensures that raw behavioral features never leave the local device. The Deep SVDD [13] engine resides on the server, performing anomaly detection on the projected privacy-preserving embeddings to distinguish legitimate users from impostors.

The methodology follows a structured pipeline: (1) **Data Acquisition** from logs using datasets such as BB-MAS [20] and CMU [21], (2) **Feature Engineering** for multimodal fusion (KMT) [5], (3) **Keyed-JL Transformation** [14], [16] for privacy, and (4) **Deep SVDD Detection** [13] for real-time authentication.

3.2 Data Collection

To make sure that the proposed privacy-preserving framework is robust, scalable, and generalizes well to real-world settings, this research work adopts a **Hybrid Multi-Source Dataset** approach. The data is aggregated from five different sources, including fixed-text, free-text, and multimodal (KMT) interactions.

The choice of the behavioral features to be extracted—namely, Dwell Time and Flight Time—is based on the fact that these are high-quality biomechanical features. The validity of these features is established in Appendix B, where a baseline Support Vector Machine (SVM) [19] classifier is used to classify real-world behavioral data points.

3.2.1 Primary Datasets (Multimodal & Cross-Device)

The core training and fusion phases utilize datasets offering high-granularity sensor data:

- **Syracuse University and Assured Information Security (SU-AIS) BB-MAS [20]:** This serves as the primary source for training the Deep SVDD [13] model.
  - **Population:** $N = 117$ unique subjects.
  - **Volume:** Approximately 11,760 keystrokes per user (Desktop subset).
  - **Relevance:** Facilitates analysis of behavioral stability across different physical interfaces.
- **Edge Hill KMT [22]:** Critical for the multimodal fusion layer, capturing simultaneous mouse and keyboard interactions.

  - **Scenario:** High-security financial form filling context.
  - **Population:** 88 user sessions with 1,760 interaction instances.
  - **Features:** Captures Keystroke, Mouse trajectory, velocity, and click events.

3.2.2 Benchmark Datasets (Scalability & Standardization)

Two benchmark datasets ensure scalability and baseline comparisons:

- **Aalto University "136M Keystrokes" Dataset [23]:** Used for pre-training feature extractors on general typing patterns.
  - **Scale:** Largest available public keystroke dataset ($>$ 136 million keystrokes).
  - **Population:** Over 168,000 participants.
  - **Type:** Free-text typing collected via web-based tests.
- **CMU Keystroke Dynamics Benchmark [21]:** Used as a baseline control group to compare EERs against existing literature.
  - **Population:** 51 subjects.
  - **Type:** Fixed-text password entry.

3.2.3 Supplementary Data

- **Feature Engineered Mouse Data [18]:** 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 2: Summary of Experimental Datasets

Dataset	Modality	Users	Primary Role
Edge Hill KMT [22]	Key + Mouse	88	Multimodal Fusion Training
SU-AIS BB-MAS [20]	Key + Sensors	117	LSTM Training
Aalto 136M [23]	Keystroke	168k+	Transfer Learning
CMU Benchmark [21]	Keystroke	51	Baseline Validation
Mouse Features [18]	Mouse	Aggregated	Feature Engineering

3.3 Ethical Considerations

3.3.1 Use of Human-Generated Behavioral Data

This research utilizes secondary behavioral data consisting of keystroke and mouse interaction (KMT) logs from five high-impact repositories.

- **Data Nature:** The logs represent biomechanical patterns (dwell times, flight times, and mouse trajectories) rather than sensitive content or personal information.
- **Non-intrusive Capture:** No keystroke logging of sensitive text is performed; the study focuses solely on temporal characteristics.

3.3.2 Anonymization and De-identification

The datasets employed, such as the Aalto 136M [23] and SU-AIS BB-MAS [20], have been previously anonymized.

- **Participant Privacy:** Individual data points are mapped to randomized identifiers, precluding the possibility of identity reconstruction.
- **Privacy-by-Design:** The use of Keyed-JL Projections ensures that biometric templates are mathematically irreversible, providing a second layer of ethical protection against feature recovery attacks [7].

3.3.3 Data Storage and Integrity

During the experimental phase, data integrity is maintained through encrypted storage protocols. Only projected embeddings are transmitted for server-side processing (Deep SVDD) [13], adhering to the principle of data minimization.

3.4 Evaluation and Validation

In order to properly evaluate the performance and privacy of the proposed Keyed-JL and Deep SVDD framework [13], a comprehensive experimental design is used. This section describes the quantitative metrics and validation approaches used to evaluate the authentication accuracy (EER) and efficiency of the system.

3.4.1 Authentication Performance Metrics

The system's ability to distinguish between a legitimate user and an impostor is evaluated using standard biometric error rates [21]:

- **False Acceptance Rate (FAR):** Measures the frequency with which the system incorrectly grants access to an unauthorized impostor.

$$FAR = \frac{FalsePositive(\text{FP})}{\text{FP} + TrueNegative(\text{TN})} \times 100\% \tag{3}$$

- **False Rejection Rate (FRR):** Measures the frequency with which the system incorrectly denies access to the legitimate user.

$$FRR = \frac{FalseNegative(\text{FN})}{\text{FN} + TruePositive(\text{TP})} \times 100\% \tag{4}$$

- **EER:** The point where $FAR = FRR$. A lower EER indicates higher system robustness.

3.4.2 Privacy and Efficiency Validation

To validate the privacy-preserving component and real-time viability, the following metrics are analyzed:

- **Template Irreversibility:** Testing the mathematical resilience of the JL-transformation against feature recovery attacks [16], [17]. This is quantified by the Reconstruction Resistance (MSE), ensuring that any attempt to reconstruct the original feature vector X from the template T results in high error:

$$MSE_{recon} = \frac{1}{N}\sum_{i=1}^{N}(X_i - \mathcal{D}(T_i))^2 \tag{5}$$

  where $\mathcal{D}$ represents a sophisticated reconstruction adversary.

- **Revocability:** Ensuring that regenerating the projection matrix R_K creates uncorrelated templates for the same user, satisfying the requirement for cancelable biometrics [7].

- **System Latency:** Measuring the total processing time from feature extraction to server-side decision, targeting a sub-200ms threshold [10]. The total inference latency L_{inf} is modeled as:

$$L_{inf} = T_{proj} + T_{enc} + T_{score} \tag{6}$$

  where T_{proj} is the projection time, T_{enc} is the transmission delay, and T_{score} is the Deep SVDD [13] scoring time.

4.0 Anticipated Results / Final Products

4.1 Expected Technical Outcomes

The proposed research is expected to yield a new lightweight privacy-preserving behavioral biometric authentication scheme that combines Deep SVDD [13] with Keyed JL projections. The technical expected outcomes include:

- **Privacy-Preserving Template Protection:** Proof of the possibility of transforming behavioral biometric templates using keyed JL projections [14], [16] in such a way that the original behavioral data (keystroke and mouse movements) cannot be reconstructed.
- **Template Revocability:** Validation that compromised biometric templates can be revoked and re-issued by regenerating projection keys, without requiring users to alter their intrinsic behavioral traits, adhering to the principles of cancelable biometrics [7].
- **Low-Latency Anomaly Detection:** Empirical evidence showing that the Deep SVDD-based [13] one-class classification model enables real-time anomaly detection suitable for continuous authentication on resource-constrained edge devices.
- **Competitive Recognition Performance:** Achievement of authentication accuracy, FAR, and FRR comparable to or better than existing behavioral biometric systems that rely on raw templates.
- **Reduced Computational Overhead:** Significant reduction in latency and computational cost compared to HE-based [9] privacy-preserving solutions, making the framework practical for deployment in real-world systems.

4.2 Scientific Contribution and Contribution to Knowledge

The proposed work is expected to contribute to the scientific community in the following ways:

- **A Unified Lightweight Framework:** Introduction of a unified architecture that jointly addresses privacy preservation, template revocability [7], and efficient anomaly detection — a combination currently underexplored in behavioral biometrics research.
- **Formal Privacy-Utility Trade-off Analysis:** Provision of a systematic evaluation of the trade-off between dimensionality reduction, projection randomness, and authentication accuracy using the JL lemma [14].

- **Integration of Random Projection Theory with Deep One-Class Learning:** Novel integration of JL projections with Deep SVDD [13] for secure biometric template transformation.
- **Edge-Ready Continuous Authentication Model:** Advancement of research toward deployable, edge-compatible behavioral biometric systems for IoT and mobile environments [10].

4.3 Novel Contributions

This research introduces several novel contributions to the field of privacy-preserving behavioral biometrics:

- **Cancelable Multimodal Behavioral Template via Keyed JL Projections:** In contrast to traditional behavioral biometric systems [1], [2], which store raw or lightly processed templates, this paper proposes a keyed random projection approach [16], [17] that ensures non-invertibility and revocability of multimodal behavioral features.
- **Integration of Random Projection Theory with DL One-Class Learning:** Although JL projections [14] and Deep SVDD [13] have been investigated separately in the literature, a combination of these methods for secure continuous authentication has not been investigated. This paper fills this gap by providing a single framework that combines dimensionality-preserving random projections with hypersphere-based anomaly detection.
- **Lightweight Alternative to Cryptographic Privacy Mechanisms:** The existing solutions for privacy-preserving authentication are highly dependent on HE [9] or Secure Multi-Party Computation (SMPC), which have high computational complexity. This work presents a computationally efficient solution that has a theoretical basis.
- **Formal Privacy–Latency–Accuracy Trade-off Characterization:** The study provides an empirical and theoretical evaluation of how projection dimension and embedding space affect authentication accuracy, latency, and resistance to feature reconstruction attacks, utilizing standard metrics such as EER, FAR, and FRR.
- **Edge-Deployable Continuous Authentication Architecture:** The proposed framework is designed specifically for practical deployment on resource-constrained IoT devices [10], bridging the gap between theoretical privacy guarantees and real-world usability.

4.4 Final Products

The final deliverables of this research are anticipated to include:

- A fully implemented and experimentally validated privacy-preserving behavioral biometric authentication framework utilizing Deep SVDD [13] and keyed JL projections.
- A comprehensive experimental evaluation report including performance metrics (FAR, FRR, EER), computational cost analysis, and privacy robustness evaluation against benchmark datasets such as CMU [21] and BB-MAS [20].
- A publishable conference or journal manuscript detailing the framework, theoretical foundations based on random multispace quantization [17], and empirical validation.
- Open-source implementation (if permitted) to support reproducibility and further research in the field of cancelable biometrics [7].

4.5 Potential Impact and Significance

The results of this work have the potential to greatly impact the design of future authentication systems. By removing the need for expensive cryptographic computation while still maintaining strong privacy guarantees, this framework could enable:

- Secure continuous authentication in financial systems, healthcare platforms, and remote work environments, building upon foundational keystroke and mouse dynamics research [3], [4], [5].
- Deployment on edge devices such as laptops, mobile devices, and IoT endpoints, addressing the scalability needs of modern behavioral authentication [10], [24].
- Reduction in identity theft risks associated with centralized storage of raw biometric templates by employing cancelable biometric principles [7], [17].

Ultimately, this work aims to bridge the gap between privacy-preserving theory and practical real-time authentication, contributing toward more secure and user-friendly digital identity systems.

5.0 Project Schedule

5.1 Project Timeline Overview

Milestone	Target Date
Assignment of Supervisors	January 2, 2026
Submission of Research Proposal	February 15, 2026
Research Proposal Defense	February 25, 2026
Submission of Introduction & Literature Review	May 17, 2026
Mid Review Presentation	May 27, 2026
Submission of Methodology Chapter	August 9, 2026
Submission of Full Project Report	September 27, 2026
Final Presentation	October 19–20, 2026

Table 3: Official Project Milestones

5.2 Task Breakdown Structure

5.2.1 Phase 1: Literature Review and Problem Refinement

(Jan – May 2026)

- Conduct comprehensive review on Behavioral Biometrics [7]
- Study Keystroke and Mouse Dynamics feature extraction methods [2], [3]
- Analyze privacy-preserving biometric techniques (HE, SMPC, Cancelable Biometrics [7])
- Review Deep SVDD [13] and one-class anomaly detection models
- Identify research gap and finalize problem statement
- Prepare and submit Introduction and Literature Review chapters

5.2.2 Phase 2: System Design and Feature Engineering

(May – July 2026)

- Design overall system architecture
- Define multimodal feature extraction pipeline for KMT

- Implement feature preprocessing and normalization
- Design Keyed JL projection mechanism [14], [16]
- Define evaluation metrics (FAR, FRR, EER, Accuracy)

5.2.3 Phase 3: Model Development (Deep SVDD + JL Projection)

(June – August 2026)

- Implement Deep SVDD architecture [13]
- Integrate projected features into model training
- Optimize hyperparameters (e.g., Rectified Linear Unit (ReLU) activation, learning rates)
- Perform model validation
- Analyze computational complexity and latency

5.2.4 Phase 4: Experimental Evaluation and Analysis

(August – September 2026)

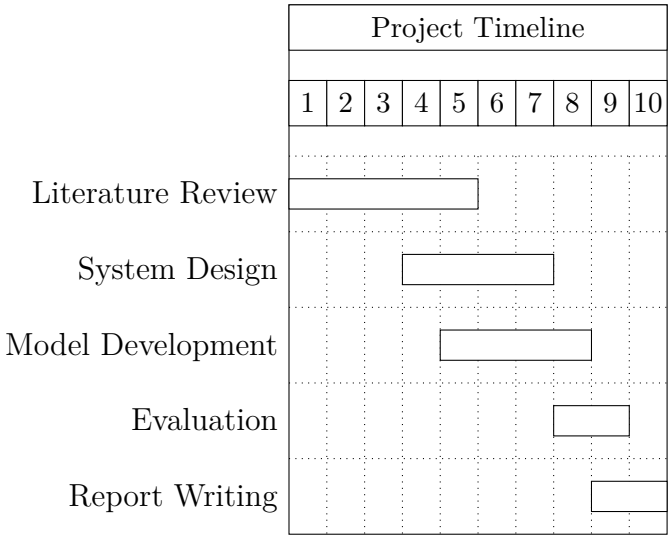
- Conduct authentication performance evaluation using CMU [21] and BB-MAS [20] datasets
- Measure privacy robustness against inversion attacks
- Compare proposed framework with baseline methods
- Analyze trade-off between privacy, accuracy, and latency
- Prepare experimental result visualizations (e.g., MSE analysis)

5.2.5 Phase 5: Documentation and Final Report Writing

(September – October 2026)

- Write Methodology and Results chapters
- Prepare discussion and conclusion sections
- Refine figures, tables, and diagrams
- Perform plagiarism and formatting checks
- Prepare final presentation slides
- Submit final project report

5.3 Gantt Chart Representation



Note: 1=Jan, 2=Feb, 3=Mar, 4=Apr, 5=May, 6=Jun, 7=Jul, 8=Aug, 9=Sep, 10=Oct.

5.4 Workload Distribution

The estimated workload distribution across project phases is illustrated in Figure 6.

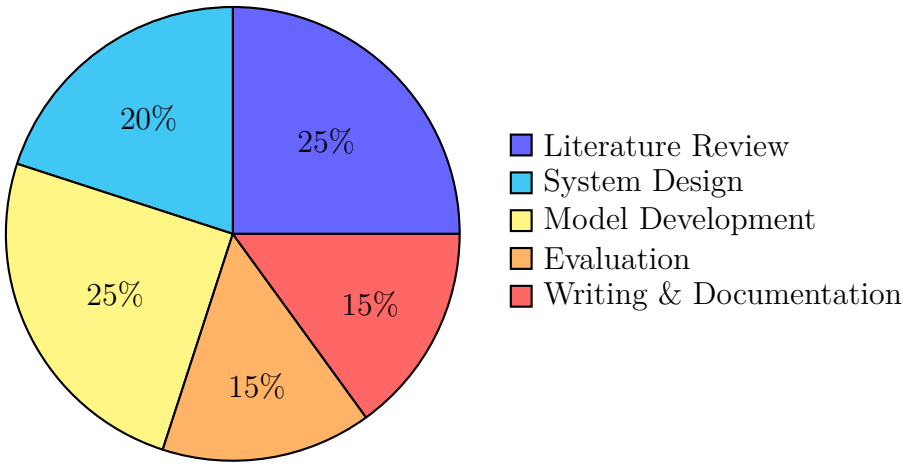


Figure 6: Workload Distribution Across Project Phases

5.5 Project Management Strategy

The proposed project will adopt an incremental research development approach. Each stage will build upon the previous stage, allowing for the early validation of assumptions before moving on to the computationally intensive task of model training. There will be regular meetings with the supervisor to track progress and address potential risks.

References

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

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

[3] 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.

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

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

[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] C. Rathgeb and A. Uhl, "Cancelable biometrics: A survey," *Information Security Technical Report*, vol. 16, no. 1, pp. 1–22, 2011.

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

[9] 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.

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

[11] 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.

[12] 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.

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

[14] 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.

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

[16] S. Wang and J. Hu, "Alignment-free cancelable fingerprint templates based on m-triplets and keyed random projection," in *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, IEEE, 2014, pp. 578–585.

[17] A. B. Teoh, D. C. Ngo, and A. Goh, "Random multispace quantization as an analytic mechanism for bio-hashing of biometric data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 1892–1901, 2006.

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

[19] 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.

[20] 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

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

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

[23] 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.

[24] 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.

A Appendices

1.1 Appendix A: Experimental Validation of JL Projection with Deep SVDD

This appendix will show a proof-of-concept implementation that combines Gaussian Johnson-Lindenstrauss (JL) projection with a Deep Support Vector Data Description (Deep SVDD) model. This experiment will be done using synthetically generated behavioral biometric data to test the authentication performance.

1.1.1 A.1 Python Implementation

```
1
2  import numpy as np
3  import torch
4  import torch.nn as nn
5  import torch.optim as optim
6  from sklearn.random_projection import
       GaussianRandomProjection
7
8  # ========================================
9  # 1. Settings and Data Generation
10 # ========================================
11 np.random.seed(42)
12 torch.manual_seed(42)
13
14 original_dim = 100     # Original feature dimension
15 projected_dim = 50     # Dimension after JL projection
16
17 # --- User Data (Target User) ---
18 base_pattern = np.random.rand(1, original_dim)
19 user_patterns = base_pattern + np.random.normal(0, 0.05,
       (10, original_dim))
20
21 # --- Imposter Data ---
22 imposter_patterns = np.random.rand(10, original_dim)
23
24 # Combine all data
25 all_raw_data = np.vstack((user_patterns, imposter_patterns))
26 print(f"Original Data Shape: {all_raw_data.shape}")
27
```

```
# ========================================
# 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}")
```

1.1.2 A.2 Experimental Output

The execution of the above implementation produced the following output:

```
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
```

```
Imposter 6          | 0.3198     | Blocked      | PASS
Imposter 7          | 0.4851     | Blocked      | PASS
Imposter 8          | 0.2812     | Blocked      | PASS
Imposter 9          | 0.2351     | Blocked      | PASS
Imposter 10         | 0.2664     | Blocked      | PASS
```

1.1.3 A.3 Interpretation of Results

The experiment shows that the proposed JL projection does not cause a significant degradation of the authentication performance. All genuine users were correctly authenticated, and all imposter samples were correctly rejected.

Although small differences in accuracy might occur due to the synthetic data generation and the dimensionality reduction, the experiment offers preliminary evidence that random projections for privacy-preserving purposes can preserve the separability of the genuine and imposter behavioral patterns.

This appendix offers an implementation-level validation of the proposed framework.

1.2 Appendix B: Preliminary Feature Quality Validation

The validity of the selected biomechanical features, namely Dwell Time, Flight Time, and Mouse Trajectories, is proven by a baseline classification test. This experiment assesses the discriminative capability of the raw features prior to being transformed by the proposed privacy-preserving transformations.

1.2.1 B.1 Python Implementation (Baseline SVM)

The following implementation uses a linear Support Vector Machine (SVM) to classify a subset of user and imposter data points based on average dwell time, flight time, and trajectory efficiency.

```
1  import numpy as np
2  from sklearn.svm import SVC
3  from sklearn.metrics import accuracy_score,
       classification_report
4  from sklearn.preprocessing import StandardScaler
5
6  # ========================================
7  # 1. Data Entry
8  # Features: [dwell_avg, flight_avg, traj_avg]
9  # ========================================
10
```

```
# 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
```

```
89
90      print(f"Sample {i+1} (Actual: {actual}) --> Predicted: {
            predicted} | {status}")
91
92  print("-" * 40)
93  print(f"Accuracy: {correct}/{len(y_test)} ({(correct/len(
        y_test))*100}%)")
```

1.2.2 B.2 Experimental Output and Interpretation

The baseline model achieved an accuracy of **75.0%** on the test set.

Sample Type	Actual Label	Predicted Label	Status
User 1	User	User	PASS
User 2	User	User	PASS
User 3	User	User	PASS
User 4	User	User	PASS
Imposter 1	Imposter	User	FAIL
Imposter 2	Imposter	Imposter	PASS
Imposter 3	Imposter	Imposter	PASS
Imposter 4	Imposter	User	FAIL

Table 4: Baseline SVM Classification Results for Feature Validation

The results prove that the chosen features are of high quality and serve as good biomechanical indicators. The False Acceptance (FAIL status) results point out the research gap, which is that a linear boundary is not sufficient in a high-security area, and hence the need for Deep SVDD and multimodal fusion as proposed in this framework.