GENERAL INSTRUCTION

- **Authors: Please check and confirm whether the name of the corresponding author is correct as set.**
- **Authors: When you submit your corrections, please either annotate the IEEE Proof PDF or send a list of corrections. Do not send new source files as we do not reconvert them at this production stage.**
- **Authors: Carefully check the page proofs (and coordinate with all authors); additional changes or updates WILL NOT be accepted after the article is published online/print in its final form. Please check author names and affiliations, funding, as well as the overall article for any errors prior to sending in your author proof corrections. Your article has been peer reviewed, accepted as final, and sent in to IEEE. No text changes have been made to the main part of the article as dictated by the editorial level of service for your publication.**
- **Authors: Per IEEE policy, one complimentary proof will be sent to only the Corresponding Author. The Corresponding Author is responsible for uploading one set of corrected proofs to the IEEE Author Gateway.**

QUERIES

Q1. Author: Please confirm or add details for any funding or financial support for the research of this article.
Q2. Author: Please provide complete bibliographic details for Ref. [40].
Q3. Author: Please provide the page range for Ref. [77].

Privacy-Preserving Continuous Authentication of Smartphone Users via Secret Sharing

Mingming Hu, Ding Wang, Daojun Han, Jingtao Guo, and Bibo Tu

***Abstract*—Behavioral biometrics based continuous authentication has been proven to be an effective supplement to traditional one-time authentication schemes (like passwords), and it can continuously authenticate users throughout the session. Currently, most continuous authentication models leverage deep learning techniques to learn smartphone users' behavioral patterns from behavioral biometric data, and have made remarkable progress. However, training deep learning based continuous authentication models requires enormous computing power, which leads to resource-constrained mobile platforms relying on powerful cloud servers. Cloud servers need access to behavioral biometric data for training and inference, as this may *raise privacy concerns*. Existing deep learning based continuous authentication schemes pay more attention to authentication *performance*, but ignore the *protection* of behavioral biometric data. To solve this issue, we present a privacy-preserving continuous authentication scheme based on secret sharing secure multi-party computation (MPC). More specifically, we secretly share behavioral biometric data among two cloud servers that train continuous authentication systems on joint data using two-party computation (2 PC) without compromising data privacy. Further, we evaluate the practical feasibility of our proposed privacy-preserving scheme on two realistic privacy-preserving continuous authentication models, which are constructed with deep learning and traditional machine learning techniques, respectively. Extensive experiments demonstrate the effectiveness of our two privacy-preserving continuous authentication models.**

***Index Terms*—Continuous authentication, behavioral biometrics, secret sharing, privacy-preserving.**

I. Introduction

With the improvement in processing and storage capacity of smartphones, they have become an indispensable tool in our daily lives. Instead of just making phone calls and sending text messages, smartphones have other complicated functions, such as online shopping, electronic payment, video calling, online banking, taking photos, etc. More and more smartphone users choose to store private data and sensitive information on their smartphones [1]. Hence, attackers may obtain users' personal data and valuable information by hacking their smartphones. Therefore, authentication schemes are essential to protect the security of confidential data and sensitive information stored on smartphones.

The traditional one-time identity authentication schemes can be divided into two categories: 1) Knowledge-based schemes, such as passwords, PINs, and graphic patterns; 2) Physiological biometrics-based schemes, such as fingerprint, face, and iris authentication. However, these authentication schemes suffer some drawbacks. Knowledge-based authentication schemes are vulnerable to attacks, such as smudge attacks [2], [3], shoulder surfing attacks [4], side channel attacks [5], and dictionary attacks [6]. Physiological biometrics based authentication schemes require the support of additional hardware, and there is a risk of leakage of smartphone user's physiological biometrics. More importantly, both of them only authenticate the access user once at the beginning of a session and do not provide any protection against access by unauthorized after login.

The continuous authentication scheme based on behavioral biometrics has become a promising approach for protecting smartphone private data security [7]. Compared with traditional one-time authentication schemes, continuous authentication scheme based on behavioral biometrics has two advantages: 1) Seamless verification: It authenticate the access user's identity throughout the session and does not require the user to participate throughout the entire authentication process; 2) Hardware independence: No additional hardware devices need to be installed to collect behavioral biometric data, and the embedded sensors can be used to collect behavioral biometric data for modeling smartphone user's behavioral patterns.

Thanks to the raw data comprehending and deep feature extraction capabilities of deep learning techniques, continuous authentication models based on deep learning techniques have achieved exciting authentication performance. Compared with classical machine learning algorithms [8], [9], [10], [11], [12], [13], [14], [15], [16], deep learning models can extract more deep features from behavioral biometric data, and can better model smartphone user's behavioral patterns, such as Convolutional Neural Networks (CNN) [13], [14], [17], [18], [19], [20], [21], [22], [23], [24], [25] and Long Term Short Memory networks

Received 1 July 2025; revised 19 March 2026; accepted 27 March 2026. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFB3106303 and in part by the Tianjin Natural Science Foundation Project under Grant 25JCJQJC00230. *(Corresponding authors: Ding Wang; Bibo Tu.)*

Mingming Hu is with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100045, China, also with the School of Cyber Security, University of Chinese Academy of Sciences, Beijing 101408, China, and also with Yellow River Henan Bureau, Zhengzhou, Henan 453500, China (e-mail: 14120390@bjtu.edu.cn).

Ding Wang is with the College of Cyber Science, Nankai University, Tianjin 300350, China (e-mail: wangding@nankai.edu.cn).

Daojun Han is with the School of Computer and Information Engineering, Henan University, Kaifeng, Henan 475004, China (e-mail: hdj@henu.edu.cn).

Jingtao Guo is with the School of Software, Henan University, Kaifeng, Henan 475004, China (e-mail: jingtaoguo@henu.edu.cn).

Bibo Tu is with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100045, China, and also with the School of Cyber Security, University of Chinese Academy of Sciences, Beijing 101408, China (e-mail: tubibo@iie.ac.cn).

Digital Object Identifier 10.1109/TDSC.2026.3680150

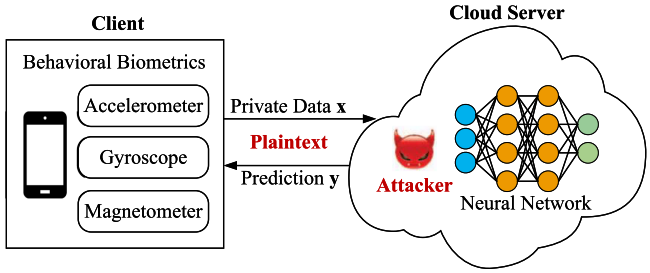


Fig. 1. Training deep neural networks based continuous authentication model with plaintext on the cloud server.

(LSTM) [26], [27], [28], [29], [30]. With the growing complexity of deep learning architectures and limited computing resources on mobile platforms, researchers have started to train deep learning techniques based continuous authentication models relying on powerful untrusted cloud servers without training data (behavioral biometric data) protection consideration (Fig. 1).

A. Motivations

Sending sensitive behavioral biometric data to an untrusted cloud server may compromise the data privacy of behavioral biometric data: As shown in Fig. 1, attackers may get the behavioral biometric data of smartphone users by attacking the cloud server. Studies have shown that attackers may infer the relevant information of the user through the behavioral biometric data, such as typing information [31], [32], [33], demographics [34], age and gender [35], [36].

To alleviate privacy concerns, different privacy-preserving methods have been utilized in classical machine learning based continuous authentication schemes, such as the variations of NTT-Sec [37], sensitive information removal [38], cancelable biometrics [39], [40], garbled circuits [41], and format-preserving encryption [42]. Unfortunately, the above privacy-preserving continuous authentication schemes are only capable of supporting models based on *basic classical machine learning algorithms*, such as SVM, Logistic Regression, and euclidean distances. Due to the *enormous computational overhead*, they are still *unsuitable* for training privacy-preserving continuous authentication models *based on deep learning techniques in practice*.

To solve this issue, this paper presents the privacy-preserving continuous authentication scheme based on secret sharing secure MPC protocol, which allows us to train continuous authentication models based on modern deep learning techniques on untrusted cloud servers without violating the privacy of behavioral biometric data. The proposed privacy-preserving continuous authentication scheme has two advantages: 1) The authentication accuracy closely matches that of the plaintext deep learning evaluation; 2) Training deep learning model is much faster than the homomorphic encryption-based approach. We secretly share the behavioral biometric data among two non-colluding cloud servers, and they can train continuous authentication models jointly without learning anything about the training data. Additionally, we generate Beaver triples using the additively homomorphic Paillier cryptosystem combined with additive secret sharing, eliminating the need for a trusted third party. To reduce communication overhead, we optimize ReLU and Maxpool operations through a secure comparison protocol based on MSB extraction. This approach utilizes the sign bit of the two's complement representation to streamline computation, making the framework highly efficient for mobile applications with limited bandwidth and power.

B. Contributions

To evaluate the practical feasibility of our proposed privacy-preserving scheme, we construct two realistic privacy-preserving continuous authentication models with deep learning techniques and traditional machine learning, respectively. First, we present a **P**rivacy-**P**reserving continuous authentication model based on **D**eep **L**earning techniques (**PPDL**), considering the correlation between the behavioral biometric data sequence. Second, we propose a **P**rivacy-**P**reserving continuous authentication model based on the **L**inear **SVM** with manually constructed features (**PPLSVM**). Extensive experiments demonstrate the effectiveness and security of our proposed two privacy-preserving continuous authentication models.

We summarize our contributions of this paper as follows:

- *A privacy-preserving deep learning based continuous authentication model:* Existing continuous authentication models based on deep learning techniques focus primarily on performance but ignore the protection of training data (behavioral biometric data). To avoid revealing smartphone users' behavioral biometric data to the cloud servers during training, we propose a **P**rivacy-**P**reserving **D**eep **L**earning based continuous authentication model (**PPDL**), which is trained among two non-colluding cloud servers without revealing sensitive data to each other. Based on the proposed PPDL, we can fully use modern deep learning techniques to improve authentication performance without compromising the privacy of behavioral biometric data.
- *A privacy-preserving linear SVM-based continuous authentication model:* Based on the collected behavioral biometric data, we manually construct 69 statistical features, such as mean, standard deviation, variance, skewness, kurtosis, etc. Since classical machine learning algorithms cannot automatically extract discriminative features from the 69 constructed statistical features, and some statistical features may negatively affect the authentication performance. Therefore, we apply four different feature selection algorithms to select discriminative features from 69 constructed statistical features: Chi2, Pearsonr, Recursive Feature Elimination (RFE), and ANOVA F_classif. Then the selected discriminative features are fed to the proposed **P**rivacy-**P**reserving **L**inear **SVM** (**PPLSVM**) for learning the behavioral patterns of smartphone users.
- *Extensive experimental evaluation:* We perform a series of experimental evaluations to demonstrate the effectiveness of our two privacy-preserving continuous authentication models on the HMOG dataset. Experimental results show that the proposed PPDL achieves an average accuracy of 99.54%, an average false acceptance rate (FAR) of 0.35%,

and an average false rejection rate (FRR) of 0.81%. In addition, we also obtain 97.61% average accuracy, 2.16% average FAR, and 2.82% average FRR with the proposed PPLSVM.

- *Random attack analysis:* We invite 30 volunteers to participate in the experiments to evaluate the ability of the proposed PPDL and PPLSVM to resist random attacks in practice. We randomly select one volunteer as the legitimate user for each round of random attack experiments, and the remaining volunteers serve as attackers. Each attacker interacts with the legitimate user's smartphone with three interaction scenarios (Website, WeChat, and TikTok), and we collect behavioral biometric data based on the developed data collection App. Experimental results demonstrate that the proposed two privacy-preserving continuous authentication models (PPDL and PPLSVM) exhibit high resilience to random attacks in real-world environments.

The remainder of this paper is organized as follows. Section II reviews the related work. We introduce the threat model in Section III. Section IV presents the designed privacy-preserving continuous authentication system. Comprehensive evaluation experiments of the proposed two privacy-preserving continuous authentication models are presented in Section V. We discuss our work in Section VII. Finally, we conclude this work in Section VIII.

II. Related Work

A. Continuous Authentication Based on Behavioral Biometrics

Modern smartphones have many built-in sensors, such as accelerometer, gyroscope, magnetometer, touchscreen sensor, gravity, etc., and the behavioral biometric data collected from these sensors can imply users' behavioral patterns when interacting with smartphones. This paper leverages the behavioral biometric data collected from three sensors (accelerometer, gyroscope, and magnetometer) to learn smartphone users' behavioral patterns.

Early research in continuous authentication primarily leveraged classical machine learning algorithms, such as KNN, Hidden Markov Model (HMM), Naive Bayes (NB), and SVM, to learn user behavioral patterns from touchscreen and motion data [8], [16], [43], [44]. The primary merit of these approaches is their low computational complexity, which makes them suitable for local execution on resource-constrained mobile devices. However, they rely heavily on manual feature engineering [9], [15], which often fails to capture the intricate, non-linear dependencies inherent in behavioral biometrics. Consequently, these models frequently struggle to maintain high accuracy and low FAR in complex, real-world interaction scenarios. To extract rich contextual semantic information for learning users' behavioral patterns, recent schemes [10], [45] have employed a two-model framework to achieve satisfactory authentication performance. A deep learning model is utilized to extract discriminative deep features from behavioral biometric data, while a traditional machine learning model is trained specifically for attacker detection. However, these methods incur significant computational overhead and increased latency during the authentication phase.

Due to the powerful deep feature extraction capabilities, deep learning techniques have been introduced to behavioral biometrics based continuous authentication models. Spatiotemporal architectures have been widely explored to capture the complex dynamics of users' behavioral patterns. Wu [14] and Sun et al. [28] demonstrated that CNNs and multi-view bagging architectures can effectively extract intrinsic fingertip-touch patterns and sequential tapping information. To model long-term temporal dependencies, hybrid CNN-LSTM models [27] and specialized LSTM architectures [26], [29] were proposed for modeling users' behavioral patterns. The primary merit of these models lies in their high recognition accuracy within closed-set scenarios. However, the requirement for comprehensive labeled data from both legitimate users and attackers often proves prohibitive for individual smartphone deployments, where only the owner's behavioral patterns are accessible for model fitting. To enhance model discriminability and robustness, researchers have integrated advanced mechanisms such as metric learning and attention. Wang [17] and Wu et al. [20] utilized deep metric learning and Siamese networks to learn robust user embeddings, while Song et al. [21] designed a Feature Regularization Net (FRN) to fuse handcrafted and deep features. Recently, Zhao et al. [46] leveraged a gated two-tower Transformer to capture fine-grained motion patterns across temporal and channel dimensions. The strength of these methods is their superior generalization across diverse usage contexts. Nonetheless, the high complexity of Transformer-based self-attention and the intricate fusion logic in FRN increase the system's latency, potentially hindering the real-time responsiveness required for continuous authentication. Recognizing the lack of negative samples in real-world settings, unsupervised and generative models have gained traction. Li et al. [19] utilized GANs to synthesize training data, effectively alleviating data scarcity. Furthermore, DSVDD (Deep Support Vector Data Description) [47] and Autoencoder (AE) variants were applied to learn one-class representations from normal behavioral samples [18], [25], [48], [49], [50]. However, unsupervised models may inadvertently reconstruct anomalous patterns with low error, leading to a higher FAR compared to their supervised counterparts.

As deep learning architectures become more complex and mobile platforms face restricted computing capabilities, researchers have begun developing continuous authentication models utilizing deep learning. These models often rely on powerful, untrusted cloud servers, but may overlook the protection of training data, particularly behavioral biometric data. However, transmitting sensitive behavioral biometric data to a non-secure cloud server poses a risk to data privacy, as malicious entities could potentially obtain the biometric information of smartphone users by compromising the security of the cloud infrastructure. Existing deep learning-based continuous authentication schemes pay more attention to authentication performance, yet often overlook the safeguarding of behavioral biometric data.

B. Privacy-Preserving Continuous Authentication

Various privacy-preserving continuous authentication schemes based on basic classical machine learning algorithms

have been proposed to protect the leakage of behavioral biometric data during the authentication phase.

To enhance user privacy when outsourcing computation to trusted third parties, researchers have explored various privacy-preserving continuous authentication frameworks. Jakobsson et al. [51] focused on data-level anonymization, including sensitive information removal, pseudonymization, and data aggregation. The primary merit of these approaches is their low computational complexity. However, data coarsening leads to a substantial decrease in authentication accuracy.

To provide stronger security guarantees, several studies have leveraged sophisticated cryptographic primitives. Homomorphic encryption (HE) has been utilized to allow for profile matching and similarity calculations directly on encrypted behavioral features [52], [53], [54]. Similarly, Paillier encryption and oblivious transfer protocols are combined to safeguard biometric attributes [55], [56], [57]. Furthermore, Acar et al. [37] proposed password-authenticated key exchange (PAKE) for secure template generation. While these cryptographic methods provide robust mathematical security for outsourced data, their main limitation is the substantial computational burden and communication latency they impose. As noted in the literature, the high overhead of HE typically hinders system scalability and makes real-time application on mobile devices difficult.

In response to these efficiency concerns, recent research has shifted toward format-preserving and optimized outsourcing schemes. Gasti et al. [41] utilized garbled circuits to reduce computation consumption in the cloud, while Hern'andez-'Alvarez [42] and Jacob et al. [58] leveraged Format-Preserving Encryption (FPE) for motion sensor data. The strength of these FPE-based approaches lies in their ability to maintain the original data format, allowing machine learning models to operate with minimal accuracy loss and significantly lower latency compared to full HE. Nevertheless, these methods must be carefully implemented to avoid potential security vulnerabilities, as maintaining the data format can sometimes provide unintended leakage of the underlying statistical distribution compared to more traditional encryption standards.

Unfortunately, few works have focused on privacy-preserving continuous authentication schemes based on deep learning techniques. We believe this work makes a new direction for designing privacy-preserving continuous authentication schemes based on deep learning techniques.

III. Threat Model

This section describes the threat model of the proposed privacy-preserving continuous authentication scheme from two aspects: 1) Continuous authentication based on behavioral biometrics; 2) Secret sharing secure MPC protocol.

Continuous authentication system based on behavioral biometrics: We assume that attackers successfully crack the one-time authentication mechanism (passwords, PINs, and graphic patterns) of the legitimate user's smartphone through some attack methods, such as smudge attack [2], [3], shoulder surfing attack [4], side channel attack [5], and dictionary attacks [6]. Furthermore, the random attackers attempt to use their own behavioral biometric data or behavioral biometric data from public datasets to attack the continuous authentication system. Importantly, these attackers have no prior knowledge of the legitimate users' behavioral patterns when interacting with their smartphones.

Secret sharing secure MPC protocol: We consider that the client outsources the deep learning based continuous authentication model training to two untrusted cloud servers. The secret sharing secure MPC protocol is designed for a two-party semi-honest (honest-but-curious) setting, which is similar to many representative secure MPC protocols [59], [60], [61], [62], [63], [64], [65], [66], [67], [68]. We assume that S_0 and S_1 are non-colluding, meaning they do not exchange their internal states or secret shares with each other beyond what is explicitly required by the protocol, but attempt to learn information about other parties' private input, and the client is fully trusted. The client (the user's smartphone) is considered a fully trusted entity responsible for collecting raw sensor data and generating initial secret shares. We assume that the communication channels between the client and the cloud servers are secured (e.g., via TLS/SSL) to prevent external eavesdropping or tampering.

IV. Privacy-Preserving Continuous Authentication System

This paper aims to train the deep learning techniques based continuous authentication model on the cloud servers without compromising data privacy of behavioral biometric data. Therefore, we propose a privacy-preserving continuous authentication scheme based on secret sharing MPC protocol [69]. The proposed secret sharing MPC protocol enables untrusted servers to interact and compute a joint function of the confidential data while only revealing the output to each other. As shown in Fig. 2, our proposed privacy-preserving continuous authentication architecture consists of two main entities: one client for behavioral biometric data collection, additive secret sharing protocol execution, and preprocessing. The two service providers S_0 and S_1 for jointly training the deep learning techniques based continuous authentication model without compromising data privacy. To protect the privacy of behavioral biometric data x, we randomly split it into two secret shares $\langle x\rangle_0$ and $\langle x\rangle_1$ based on the additive secret sharing protocol. Then the two secret shares $\langle x\rangle_0$ and $\langle x\rangle_1$ are sent to the service providers S_0 and S_1 without revealing the actual value of two secret inputs, respectively. We will describe the implementation in detail below. Table I lists the notations used in this paper.

A. System Setup

The proposed privacy-preserving continuous authentication system's execution begins with a setup phase where server S_0 generates a Paillier public/private key pair (pk, sk) and shares the public key pk with S_1. This allows both servers to perform offline pre-computation using Algorithm 1 to generate Beaver triples (a, b, c), which are stored locally to facilitate secure multiplication during training and inference. In the online phase, the client executes the secret sharing algorithm on the raw behavioral biometric data x. The client then generates two

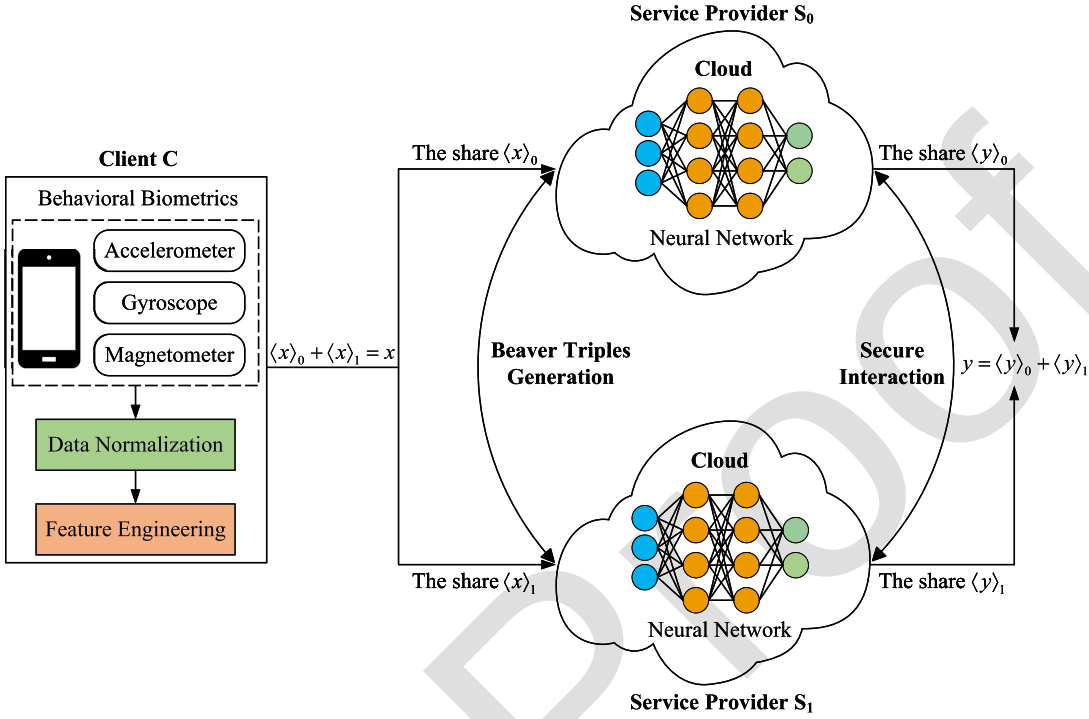


Fig. 2. The architecture of privacy-preserving continuous authentication system based on secret sharing secure MPC protocol.

TABLE I
THE ABBREVIATIONS AND NOTATIONS IN THIS PAPER

Notations	Definition
$\mathbb{Z}_{2^k}$	refers to the ring of integers modulo 2^k
$\langle x \rangle_i$	the portion of the secret share data held by Server i
a,b,c	refers to random intermediate values generated during the execution of the additive secret sharing protocol
W	the weight matrix of a specific layer in neural network
M_i	the trained continuous authentication model in Server i
$x_{acc}, y_{acc}, z_{acc}$	measures the acceleration force along three physical axes
$x_{gyr}, y_{gyr}, z_{gyr}$	measures the gyroscope force along three physical axes
$x_{mag}, y_{mag}, z_{mag}$	measures the magnetometer force along three physical axes
$\langle \delta^{(l)} \rangle$	the gradient of the loss function with respect to the pre-activation input of layer l
$b^{(l)}$	refers to the bias vector associated with the l-th layer of the neural network
$\langle y_{i,j} \rangle_i$	a specific secret share of an output value from the operation held by the server i
ReLU	the activation function used in neural networks
MSB(x)	returns the most significant bit of a binary number x

additive secret shares by selecting a random number $r \in \mathbb{Z}_{2^k}$ and computing $\langle x \rangle_0 = r$ and $\langle x \rangle_1 = x - r \mod 2^k$. These shares are independently transmitted to S_0 and S_1 via secure TLS/SSL channels. Because r is uniformly distributed, each share appears as random noise to the individual servers, ensuring that the original biometric data x cannot be reconstructed by either server individually, thereby preserving data privacy.

B. Secure Computation

This paper implements the secure two-party computation scheme based on the 2-out-of-2 additive secret sharing protocol. During function computation, the intermediate value is secret-shared between the two service providers S_0, S_1. For the original behavioral biometric data point x, the client generates a random number r in the ring $\mathbb{Z}_{2^k}$, The data is then split into two additive secret shares: $\langle x \rangle_0 = r$ and $\langle x \rangle_1 = x - r \mod 2^k$. The client independently transmits the share $\langle x \rangle_0$ to service provider S_0 and the share $\langle x \rangle_1$ to service provider S_1 via secure communication channels. Then the S_0 holds the $\langle x \rangle_0 = r$ and the S_1 holds the $\langle x \rangle_1 = x - r \mod 2^k$. Since the random number r is uniformly distributed, each individual share appears as random noise to the servers. As long as the two service providers S_0 and S_1 do not collude, they cannot reconstruct the original behavioral biometric data x. For reconstructing the secret data x, one party S_i should send to another party S_{1-i} who computes the secret data $x = \langle x \rangle_0 + \langle x \rangle_1 \mod 2^k$, where i = 0, 1. In this paper, the k is set to 64.

Secure addition and subtraction: The two parties S_0 and S_1 can perform the addition and subtraction locally without interacting. To compute the $z = x \pm y$, the party S_i holds the $\langle x \rangle_i$ and $\langle y \rangle_i$, and then computes the $\langle z \rangle_i = \langle x \rangle_i \pm \langle y \rangle_i \mod 2^k$ without communication.

Beaver triples generation: Beaver triples are used for secure multiplication, which can perform secure computation on two parties without compromising data privacy. We generate beaver triples using the additively homomorphic paillier cryptosystem [70] and additive secret sharing rather than a trusted third party. The goal of generated beaver triples are to mask private data x and y without revealing them to the party S_i during secure multiplication. Giving additively shared secrets $\langle a \rangle_i$ and $\langle b \rangle_i$, our goal is to obtain $\langle c \rangle_i$ where $c = a \cdot b$, and $a = \langle a \rangle_0 + \langle a \rangle_1 \mod 2^k$, $b = \langle b \rangle_0 + \langle b \rangle_1 \mod 2^k$ and $c = \langle c \rangle_0 + \langle c \rangle_1 \mod 2^k$.

Algorithm 1: Beaver Triples Generation Protocol Based on Additively Homomorphic Paillier Cryptosystem and Additive Secret Sharing. The Gen is Used to Generate the Public Key pk and Secret Key sk. The Function $Enc(pk, p)$ Generates the Ciphertext d of Plaintext Message p. The Function $Dec(sk, d)$ Returns the Plaintext Message p From the Ciphertext d.

Input: The user generates additively shared $\langle a\rangle$ and $\langle b\rangle$, and shares the shared secrets $\langle a\rangle_i$ and $\langle b\rangle_i$ with the party S_i. The party S_0 generates the public key pk and secret key sk, and shares pk with S_1.

Output: The additively shared $\langle c\rangle_i$, where $c = \langle c\rangle_0 + \langle c\rangle_1 \bmod 2^k$, and $c = (\langle a\rangle_0 + \langle a\rangle_1) \cdot (\langle b\rangle_0 + \langle b\rangle_1) \bmod 2^k$.

1: The party S_0 encrypts the $\langle a\rangle_0$ and $\langle b\rangle_0$, and send the $Enc(pk, \langle a\rangle_0)$ and $Enc(pk, \langle b\rangle_0)$ to S_1.
2: The party S_1 selects a random value $v \in \mathbb{Z}_{2^k}$.
3: With the public key pk, the party S_1 computes and sends the $q = Enc(\langle a\rangle_0 \cdot \langle b\rangle_1 + \langle b\rangle_0 \cdot \langle a\rangle_1 + v) = (Enc(pk, \langle a\rangle_0))^{\langle b\rangle_1} \cdot (Enc(pk, \langle b\rangle_0))^{\langle a\rangle_1} \cdot Enc(pk, v)$ to S_0.
4: The party S_0 computes the $\langle c\rangle_0 = \langle a\rangle_0 \cdot \langle b\rangle_0 + Dec(q) \bmod 2^k$.
5: The party S_1 computes the $\langle c\rangle_1 = \langle a\rangle_1 \cdot \langle b\rangle_1 - v \bmod 2^k$.

Algorithm 2: Secure Multiplication Protocol for Jointly Computing the $f(x, y) = x \cdot y$ With Two Parties S_0, S_1.

Input: With the private shared data x and y generated by the client, the party S_i gets the shared secrets $\langle x\rangle_i$ and $\langle y\rangle_i$ as inputs. Beaver triple $\langle a\rangle$ and $\langle b\rangle$, where $c = a \cdot b$.

Output: S_i outputs f_i, where $i = 0, 1$.

1: The party S_i gets the additively shared $\langle a\rangle_i$, $\langle b\rangle_i$ and $\langle c\rangle_i$ by the Algorithm 1: $a = \langle a\rangle_0 + \langle a\rangle_1 \bmod 2^k$, $b = \langle b\rangle_0 + \langle b\rangle_1 \bmod 2^k$, and $c = \langle c\rangle_0 + \langle c\rangle_1 \bmod 2^k$.
2: The party S_0 computes the $\langle e\rangle_0 \leftarrow \langle x\rangle_0 - \langle a\rangle_0$ and the $\langle m\rangle_0 \leftarrow \langle y\rangle_0 - \langle b\rangle_0$.
3: The party S_1 computes the $\langle e\rangle_1 \leftarrow \langle x\rangle_1 - \langle a\rangle_1$ and the $\langle m\rangle_1 \leftarrow \langle y\rangle_1 - \langle b\rangle_1$.
4: The two parties S_0 and S_1 share the value e and m.
5: The party S_0 computes the $f_0 \leftarrow m \cdot \langle a\rangle_0 + e \cdot \langle b\rangle_0 + \langle c\rangle_0$.
6: The party S_1 computes the $f_1 \leftarrow e \cdot m + m \cdot \langle a\rangle_1 + e \cdot \langle b\rangle_1 + \langle c\rangle_1$.
7: **return** $f \leftarrow f_0 + f_1$.

The process of beaver triples generation with additively homomorphic paillier cryptosystem and secret sharing protocol is elaborated in Algorithm 1.

Secure multiplication: In this paper, we implement secure multiplication with the beaver triples [71], which can perform multiplication computation between two parties while protecting the privacy of input data. Two parties S_0 and S_1 can jointly compute the $f(x, y) = x \cdot y$. The S_i will compute the f_i locally, where $f_0 + f_1 = f$. First, the party S_i generates the additively shared $\langle a\rangle_i$, $\langle b\rangle_i$ and $\langle c\rangle_i$ by Algorithm 1. Second, each party S_i computes the $\langle e\rangle_i = \langle x\rangle_i - \langle a\rangle_i$ and $\langle m\rangle_i = \langle y\rangle_i - \langle b\rangle_i$ locally. Third, the party S_i computes the $f_i = i \cdot e \cdot m + m \cdot \langle a\rangle_i + e \cdot \langle b\rangle_i + \langle c\rangle_i$, and then $f_0 + f_1 = f$. We elaborate the computation process of secure multiplication in Algorithm 2.

Most significant bit: The most significant bit (MSB) is the bit with the largest value in multi-bit binary numbers. For example, we can represent $x \in \{0, 1\}^n$ as an n-bit binary number: $\{x_{n-1}, x_{n-2}, \cdots, x_1, x_0\}$, and the x_{n-1} is called the most significant bit (MSB).

In this paper, we compute the ReLU function and Maxpool function based on the secure MSB protocol, which is achieved by masking the inputs with random values that will be converted into sharing of bits. The bits of original inputs can be revealed by performing the bit-wise operations, which could be reduced to the bit extraction operations. Specifically, given two shares u_i, v_i, S_0 and S_1 can compute $w_0 = u_0 - v_0$ and $w_1 = u_1 - v_1$, respectively. By invoking the secure MSB protocol, two parties can get the most significant bit of w, where $w = w_0 + w_1$. As $u = u_0 + u_1$ and $v = v_0 + v_1$, the most significant bit of w denotes the comparison result of u and v.

Privacy-preserving training: Training privacy-preserving deep neural networks is a challenging task that requires huge computational overhead. This paper aims to train the privacy-preserving continuous authentication models on the cloud server without compromising data privacy. In this case, we perform the function f on the two parties S_0 and S_1 with the shared private data $(\langle x\rangle_0, \langle x\rangle_1)$, and the $x = \langle x\rangle_0 + \langle x\rangle_1 \bmod 2^k$. Then the party S_i learns nothing about the training data while training the privacy-preserving continuous authentication model.

Privacy-preserving inference: Compared with privacy-preserving training, privacy-preserving influence requires less computational overhead and faster inference time. Therefore, more researchers pay more attention to privacy-preserving inference with pre-trained deep neural networks. In this paper, we evaluate the trained privacy-preserving continuous authentication models M_0 and M_1 on the two parties S_0 and S_1 with the shared test private data $(\langle x\rangle_0, \langle x\rangle_1)$, where the $x = \langle x\rangle_0 + \langle x\rangle_1 \bmod 2^k$ and the $M \leftarrow \text{Reconstruct}(\langle M\rangle_0, \langle M\rangle_1)$. Then, the party S_i does not learn continuous authentication model parameters of M, the private behavioral biometric data x, and the inference result.

C. Data Normalization

The proposed privacy-preserving continuous authentication scheme is engineered as a multimodal privacy-preserving continuous authentication framework. By consolidating data streams from multiple motion sensors (accelerometer, gyroscope, and magnetometer), the system captures multifaceted smartphone user's behavioral patterns. This multimodal fusion not only improves authentication accuracy by exploiting the complementary nature of different sensor data but also ensures robust user privacy through an additive secret sharing mechanism, which prevents any single cloud server from accessing the raw behavioral biometric data. We collect the sensory data from three built-in motion sensors (accelerometer, gyroscope, and

magnetometer), and they can be used to capture the behavioral patterns of smartphone users. The accelerometer measures the direction of acceleration in three dimensions, which can be used to sense the relevant actions of smartphone users in continuous authentication tasks. The gyroscope measures a physical quantity's angular velocity when deflected and tilted in three dimensions. The magnetometer is applied to test the strength and direction of magnetic fields.

The sensory data collected from different motion sensors have orders of magnitude differences in value. Therefore, we need to apply data normalization methods to make the behavioral biometric data from different motion sensors fall into the same range. The data normalization operation has the following advantages: 1) Reduces the difference in feature weights and improves deep learning model convergence speed and accuracy; 2) Behavioral biometric data from different motion sensors can be visualized and analyzed in the same coordinate system, which is convenient for finding the laws and trends in the data; 3) It can effectively reduce the influence of errors and outliers in the data.

This paper uses the min-max normalization method to perform a linear transformation with the behavioral biometric data. The behavioral biometric data collected from each sensor can be denoted as $(x, y, z, m) \in \mathbb{R}^4$ at each time point, and the magnitude (m) is formulated as:

$$m = \sqrt{x^2 + y^2 + z^2} \tag{1}$$

where the x, y, and z indicate the three-axis of motion sensor data. Then the behavioral biometric data collected from three motion sensors can be denoted as $(x_{acc}, y_{acc}, z_{acc}, m_{acc}, x_{gyr}, y_{gyr}, z_{gyr}, m_{gyr}, x_{mag}, y_{mag}, z_{mag}, m_{mag}) \in \mathbb{R}^{12}$ at each time point, and the acc, gyr, and mag represent the accelerometer, gyroscope, and magnetometer, respectively. The min-max normalization operation is formulated as:

$$\tilde{\mathrm{x}}_i = \frac{x_i - \min\limits_{1 \le j \le n} \{x_j\}}{\max\limits_{1 \le j \le n} \{x_j\} - \min\limits_{1 \le j \le n} \{x_j\}} \tag{2}$$

then data can be denoted as $(\tilde{x}_{acc}, \tilde{y}_{acc}, \tilde{z}_{acc}, \tilde{m}_{acc}, \tilde{x}_{gyr}, \tilde{y}_{gyr}, \tilde{z}_{gyr}, \tilde{m}_{gyr}, \tilde{x}_{mag}, \tilde{y}_{mag}, \tilde{z}_{mag}, \tilde{m}_{mag}) \in \mathbb{R}^{12}$ at each time point after normalization.

D. Feature Engineering

Feature engineering is essential in constructing continuous authentication models based on machine learning. In this paper, we construct two privacy-preserving continuous authentication models with different feature engineering schemes based on deep learning techniques and classical machine learning algorithms, respectively. We will describe the proposed two feature engineering schemes in detail.

1) Learning Correlation Between Data Sequence With Deep Learning Model: To make the deep learning based continuous authentication model better understand the correlation between behavioral biometric data sequences and capture comprehensive semantic information, we consider the time dependence of behavioral biometric data sequences in sample generation operations. As described in Section IV-C, the behavioral biometric data sequence can be denoted as $s_i = (\tilde{x}_{acc}, \tilde{y}_{acc}, \tilde{z}_{acc}, \tilde{m}_{acc}, \tilde{x}_{gyr}, \tilde{y}_{gyr}, \tilde{z}_{gyr}, \tilde{m}_{gyr}, \tilde{x}_{mag}, \tilde{y}_{mag}, \tilde{z}_{mag}, \tilde{m}_{mag})^T \in \mathbb{R}^{12}$ at each time point. For a time window t, the data sequences can be represented as a $d \times n$ matrix $(s_1, s_2, s_3, \ldots, s_{n-1}, s_n)$, where $d = 12$ is the number of axes of data sequence at each time point from three sensors and $n = t \times f$. The behavioral biometric data are collected with a sample rate $f = 100$ HZ, and we utilize a time window $t = 0.5$ s to segment the behavioral biometric data sequence without overlap. Then the training samples have a shape of 12×50, which are used to train privacy-preserving deep learning based continuous authentication model (PPDL). In this paper, we construct a privacy-preserving convolutional neural network architecture for semantic information learning, which will be described in Section IV-E. In addition, we also perform experiments to evaluate authentication performance with different time windows in Section V-D1, such as 0.75 s, 1 s, 1.25 s, 1.5 s, and 1.75 s.

TABLE II
MANUALLY CONSTRUCTED FEATURES FOR PRIVACY-PRESERVING LINEAR SVM ALGORITHM

Domain	Statistical features	Number
Time domain	Coefficient of variation, Kurtosis, Variance, Minimum, Coefficient of Range, Mean absolute deviation, Interquartile range, Range, Standard deviation, Mean, Median absolute deviation, Quartiles (25%, 50%, 75%, 95%), Skewness, Maximum	17*3
Frequency domain	Entropy, Median frequency, Peak2f, Mean frequency, Peak1, Peak2	6*3
Total		69

2) Manually Constructed Features for Classical Machine Learning Algorithm: Classical machine learning algorithms rely heavily on the quality of manually constructed features. High-quality manually constructed features can reduce the noise in the original data, improve authentication performance, and improve stability and training efficiency.

In this paper, we manually construct statistical features based on each motion sensor's magnitude (m), and the m is calculated by (1). The manually constructed statistical features consist of time-domain features and frequency-domain features, which are described in Table II. For each motion sensor, we construct 17 time-domain features and 6 frequency-domain features, and then 69 features are constructed for three motion sensors. The role of time-domain features is to extract the characteristics of time series data so that classical machine learning algorithms can better understand and process these data sequences. For example, the Mean, Range, Maximum, Minimum, Quartile Range, Variance, and Quartile measure the trend and fluctuation degree of the behavioral biometric data sequence. The Coefficient of variation measures the degree of dispersion of data sequences.

The Kurtosis and Skewness reflect the distribution characteristics of the behavioral biometric data. Frequency-domain features benefit classical machine learning algorithms better to distinguish the frequency components and characteristics of different signals. The Entropy measures the amount of information and complexity of behavioral biometric data sequence. Based on these manually constructed features, we proposed a privacy-preserving linear SVM-based continuous authentication model (PPLSVM), which will be described in Section IV-F.

E. Privacy-Preserving Convolutional Neural Networks Architecture

Convolutional neural networks (CNNs) have proven their powerful feature extraction ability in many fields, such as image classification, object detection, semantic segmentation, and image synthesis. In this paper, we attempt to construct a privacy-preserving CNN architecture to learn rich semantic information from behavioral biometric data, which is jointly trained on two cloud servers S_0 and S_1 without compromising data privacy. Typically, a standard CNN architecture usually comprises convolutional, activation, pooling, and fully connected layers.

Convolutional layer: The input of the convolutional layer can be represented as a matrix (C, W, H), where C is the number of input channels, W is the width of input, and H is the height of input. The convolution operation performs element-wise multiplication between the kernel and a local area of the input, summing the results to produce one element of the output feature map. Convolution operation scans the whole input by sliding the position of the convolution kernel and gradually generates the output feature map. Giving the shared input $\langle x_{i,j}\rangle_i$, the party S_0 computes the shared output $\langle y_{i,j}\rangle_0$ as below:

$$\langle y_{i,j}\rangle_0 = \sum_{a=0}^{n-1}\sum_{b=0}^{n-1}\langle w_{a,b}\rangle\langle x_{i+a,j+b}\rangle_0 + \langle b\rangle \tag{3}$$

and the party S_1 computes shared output $\langle y_{i,j}\rangle_1$ as below:

$$\langle y_{i,j}\rangle_1 = \sum_{a=0}^{n-1}\sum_{b=0}^{n-1}\langle w_{a,b}\rangle\langle x_{i+a,j+b}\rangle_1 \tag{4}$$

where the $x_{i,j}$ is the input value at position (i, j), the b is bias, and the weight matrix w has the same shape $n \times n$ with the kernel size.

Activation layer: The activation layer plays an important role in CNN architecture. The nonlinear activation function introduces nonlinear transformations, which increase the expression ability of the neural network and the ability to fit complex functions. The ReLU is a commonly used nonlinear activation function in CNN architecture, and we apply the ReLU in this paper. The ReLU function can be denoted as:

$$\mathrm{ReLU}(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{5}$$

In this paper, the two parties S_0 and S_1 hold the shared data $\langle x\rangle_0$ and $\langle x\rangle_1$, where the $x \leftarrow \mathrm{Reconstruct}(\langle x\rangle_0, \langle x\rangle_1)$. The MSB protocol allows two parties S_0 and S_1 work together to determine the MSB of x without revealing the actual value of x. The ReLU function on S_0 and S_1 can be formulated as:

$$\langle y_{i,j}\rangle_0 = \begin{cases} \langle x_{i,j}\rangle_0, & MSB(x) = 0 \\ 0, & MSB(x) = 1 \end{cases},$$
$$\langle y_{i,j}\rangle_1 = \begin{cases} \langle x_{i,j}\rangle_1, & MSB(x) = 0 \\ 0, & MSB(x) = 1 \end{cases} \tag{6}$$

Pooling layer: The pooling layer reduces spatial dimension of the feature map by downsampling the feature map, thus reducing the parameters and computations of neural networks and effectively reducing the over-fitting problem.

In CNN architecture, two commonly used pooling functions are the max pooling function and the average pooling function. The max pooling selects the maximum value in each pooled window as the output, which can keep the most prominent features in the window and improve the robustness and abstraction of features. The average pooling computes the average of the characteristic values in each pooled window as the output, which can smooth the feature map, reduce noise and redundant information. This paper uses the max pooling function to downsample the feature map, which is a commonly used pooling function in deep neural network architectures. Due to the two parties S_0 and S_1 hold the shared data $(\langle x\rangle_0, \langle x\rangle_1)$, and we can't find the maximum value by comparison on each party S_i without revealing the private data x. In this case, the two parties S_0 and S_1 need to cooperate to find the maximum value in a pooled window indirectly. Giving the two value $x_{i,j}, x_{m,n}$, and the party S_0 holds the shared value $\langle x_{i,j}\rangle_0, \langle x_{m,n}\rangle_0$ and the party S_1 holds the shared value $\langle x_{i,j}\rangle_1, \langle x_{m,n}\rangle_1$. Then the two parties S_0 and S_1 compute the $(\langle x_{i,j}\rangle_0 - \langle x_{m,n}\rangle_0)$ and $(\langle x_{i,j}\rangle_1 - \langle x_{m,n}\rangle_1)$ locally without compromising the privacy of $x_{i,j}$ and $x_{m,n}$. Leveraging secure MSB protocol, the two parties can extract the MSB of $(x_{i,j} - x_{m,n})$ and judge the maximum value of $x_{i,j}$ and $x_{m,n}$ by:

$$\max(x_{i,j}, x_{m,n}) = \begin{cases} x_{i,j}, & \mathrm{MSB}\,(x_{i,j} - x_{m,n}) = 0 \\ x_{m,n}, & \mathrm{MSB}\,(x_{i,j} - x_{m,n}) = 1 \end{cases} \tag{7}$$

Fully connected layer: Each neuron in the fully connected layer is connected with all neurons in the previous layer and is linearly combined by weight parameters. Like the convolutional layer, the activation of neurons can be computed by matrix multiplication and bias in the fully connected layer. The difference between the fully connected layer and the convolutional layer is that the neurons in the convolutional layer are connected to the local field of the input through shared weight parameters. Like (3) and (4), the party S_i can perform the fully connected operation locally based on the additive secret sharing protocol.

Forward propagation and backward propagation: Forward propagation refers to transmitting and processing input data from the input layer to the output layer of the neural network, in which the hidden layer includes a convolutional layer, activation layer, pooling layer, and fully connected layer. Backward propagation refers to the process of gradient computation and parameter updating of the parameters in the neural network according to the loss function. Fig. 3 shows the pipeline of forward propagation and backward propagation with the secretly shared data $\langle x\rangle_0$ and $\langle x\rangle_1$. During the process of backward propagation, we first

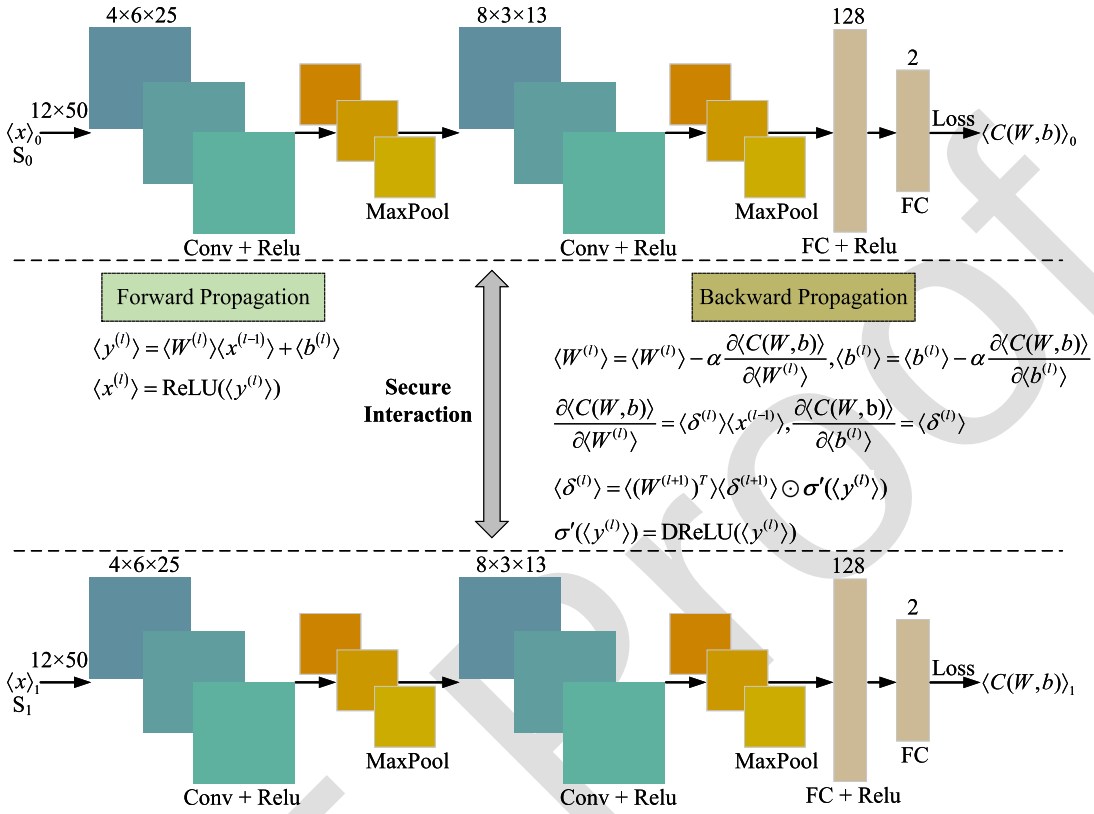


Fig. 3. The forward propagation and backward propagation of privacy-preserving CNN architecture.

compute the error $\langle C(W,b)\rangle$ between the predicted value $\langle \hat{y}\rangle$ and the real value $\langle y\rangle$ through the loss function. Then we can compute the gradient of the $l-th$ hidden layer by:

$$\frac{\partial\langle C(W,b)\rangle}{\partial\langle W^{(l)}\rangle} = \langle\delta^{(l)}\rangle\langle x^{(l-1)}\rangle, \frac{\partial\langle C(W,\mathrm{b})\rangle}{\partial\langle b^{(l)}\rangle} = \langle\delta^{(l)}\rangle \tag{8}$$

and the $\langle\delta^{(l)}\rangle$ is formulated as:

$$\langle\delta^{(l)}\rangle = \langle(W^{(l+1)})^T\rangle\langle\delta^{(l+1)}\rangle \odot \sigma'\left(\langle y^{(l)}\rangle\right) \tag{9}$$

and the σ' is the derivative of the ReLU function $\mathrm{DReLU}(x)$, which is 0 if $x<0$ and 1 if $x\geq 0$. Then we update W and b using the gradient descent algorithm, which can be formulated as:

$$\begin{aligned}\langle W^{(l)}\rangle &= \langle W^{(l)}\rangle - \alpha\frac{\partial\langle C(W,b)\rangle}{\partial\langle W^{(l)}\rangle}, \langle b^{(l)}\rangle \\ &= \langle b^{(l)}\rangle - \alpha\frac{\partial\langle C(W,\mathrm{b})\rangle}{\partial\langle b^{(l)}\rangle}\end{aligned} \tag{10}$$

and the α is the learning rate.

F. Privacy-Preserving Linear SVM Algorithm

The linear SVM is a classical machine learning algorithm, which is usually used for binary classification and multi-classification problems. Its goal is to find an optimal hyperplane that separates different samples and maximize the interval between samples and hyperplane as much as possible. Compared with deep learning techniques, the SVM algorithm has the following advantages: 1) Achieving satisfactory performance with fewer training samples; 2) It can adapt to different data types by choosing different kernel functions.

Giving the training sample pairs $(x_1,y_1),(x_2,y_2),(x_3,y_3),\ldots,(x_n,y_n)$, the goal of the SVM is to find an optimal hyperplane. The optimal hyperplane can be represented with a vector w and the bias b, which is formulated as:

$$w^T x + b = 0 \tag{11}$$

where the x is the training sample, and the y is the label. If we can find the optimal hyperplane with the data pairs correctly, the function used to predict the label can be formulated as:

$$y(x) = sign\left(w^T x + b\right) \tag{12}$$

To find the optimal hyperplane, our optimization goal can be formulated as:

$$\begin{aligned}&\min_{w,b}\frac{1}{2}\parallel w\parallel^2, \\ &\mathrm{s.t.}\ \mathrm{y_i}\left(\mathrm{w^T x_i} + \mathrm{b}\right) \geq 1, \mathrm{i} = 1,2,3\ldots,\mathrm{n}\end{aligned} \tag{13}$$

In this paper, our goal is to construct a privacy-preserving linear SVM-based continuous authentication model, which is trained on the two parties S_0 and S_1 with secretly shared data pair $(\langle x\rangle_0,\langle y\rangle_0)$ and $(\langle y\rangle_0,\langle y\rangle_0)$, where the $x=\langle x\rangle_0+\langle x\rangle_1$ mod 2^k and the $y=\langle y\rangle_0+\langle y\rangle_1$ mod 2^k.

V. Experimental Results

The proposed two privacy-preserving continuous authentication models (PPDL and PPLSVM) are evaluated on the HMOG dataset. The evaluation studies will answer the following Research Questions (RQs).

- *RQ1:* Can the proposed PPDL and PPLSVM achieve satisfactory authentication performance with providing guarantees of privacy preservation?
- *RQ2:* Are the proposed PPDL and PPLSVM resilient to random attacks in the real-world environment?
- *RQ3:* For the proposed PPDL, what is the impact on authentication performance when using different time window sizes to aggregate data and calculate features?
- *RQ4:* For the proposed PPLSVM, do the manually constructed 69 statistical features all positively impact authentication performance?
- *RQ5:* Compared with the foremost counterparts, do the proposed PPDL and PPLSVM have authentication performance advantages and security advantages?
- *RQ6:* Whether the time overhead of private training and inference is suitable for application in practice?

A. Experiment Setup

The proposed two privacy-preserving continuous authentication models are implemented with Python 3.7.12 and PyTorch 1.8.1 framework, which are trained on two parties S_0, S_1 with two Intel(R) Xeon(R) Gold 5120 CPUs @ 2.20 GHz and 128 GB of RAM. The operating system version is Ubuntu 16.04.1. The training batch size for the proposed PPDL is set to 64, and the learning rate is 0.001. For the proposed PPLSVM, the learning rate is 0.1. The behavioral biometric data collection App is developed with Android Studio basic development environment.

B. Evaluation Metrics

We apply the Accuracy, False Acceptance Rate (FAR), and False Rejection Rate (FRR) to evaluate the effectiveness of the proposed privacy-preserving authentication models.

Accuracy: Accuracy refers to the probability of the number of correctly predicted samples to all predicted samples, which can be formulated as $Accuracy = \frac{TP+TN}{TP+FN+TN+FP}$.

False Acceptance Rate (FAR): FAR refers to the probability of samples from attackers predicted to be samples from legitimate users, which can be formulated as $\mathrm{FAR} = \frac{\mathrm{FP}}{\mathrm{TN+FP}}$.

False Rejection Rate (FRR): FRR refers to the probability of samples from legitimate users predicted to be samples from attackers, which can be formulated as $\mathrm{FRR} = \frac{\mathrm{FN}}{\mathrm{TP+FN}}$.

Besides, True Positive (TP) means positive samples from legitimate users are correctly predicted as positive. False Positive (FP) means negative samples from attackers are incorrectly predicted as positive. True Negative (TN) means negative samples from attackers are correctly predicted as negative. False Negative (FN) means positive samples from legitimate users are incorrectly predicted as negative.

C. Dataset

We evaluate the effectiveness of the proposed two privacy-preserving continuous authentication models (PPDL and PPLSVM) on the HMOG dataset. Since behavioral biometric data belongs to the privacy of smartphone users, to avoid the leakage of private data, current researchers rarely open source the datasets used for the continuous authentication task. HMOG [44] is a more commonly used and convincing behavioral biometrics dataset for the continuous authentication task, which is used as a benchmark dataset by many representative works [11], [16], [18], [19], [22], [25], [30], [45], [72], [73]. This paper extracts the behavioral biometric data from three motion sensors (accelerometer, gyroscope, and magnetometer) from the HMOG dataset.

HMOG dataset: Sitová et al. [44] constructed a new behavioral biometrics dataset for the continuous authentication task, consisting of Hand Movement, Orientation, and Grasp (HMOG). They attempted to model the smartphone user's behavioral patterns with the collected behavioral biometric data from motion sensors (accelerometer, gyroscope, and magnetometer). They collected the motion sensor data from 100 participants (53 male and 47 female) with eight free text typing sessions when the participants tapped on their smartphone screens. Some smartphone users' behavioral biometric data are unavailable because the sensory data of three motion sensors (accelerometer, magnetometer, and gyroscope) are not collected simultaneously at a certain time or period during the data collection process. In addition, less behavioral biometric data collected from some smartphone users resulted in insufficient samples to train the proposed privacy-preserving continuous authentication models. After data preprocessing, 90 participants are used for the evaluation experiments in this paper.

D. Evaluation of the Proposed PPDL

1) Authentication Performance With Different Time Window Sizes: Segmenting samples with different time window sizes may affect the understanding and prediction ability of the continuous authentication model. Therefore, we need to find the optimal time window size to capture and model the behavioral patterns of smartphone users. In addition, a suitable time window can provide richer contextual semantic information, enabling continuous authentication models to extract deep features from behavioral biometric data. In this paper, we attempt to apply different time windows (0.5 s, 0.75 s, 1 s, 1.25 s, 1.5 s, and 1.75 s) to segment the behavioral biometric data to generate training samples with different shapes (12×50, 12×75, 12×100, 12×125, 12×150, and 12×175). For each smartphone user, we designate his behavioral biometric data as the positive data and the remaining smartphone users' behavioral biometric data as negative data. And then, we train and test the proposed PPDL individually for each smartphone user with a sequence of time windows and report the average experimental results of 90 smartphone users.

Table III lists the average experimental results of 90 smartphone users with different time window sizes. As shown in

TABLE III
THE EXPERIMENTAL RESULTS OF FAR (%), FRR (%), AND ACCURACY (%) WITH DIFFERENT TIME WINDOW SIZES

Window size (s)	FAR ↓	FRR ↓	Accuracy ↑
0.5	0.35	0.81	99.54
0.75	0.33	0.99	99.56
1	0.41	1.47	99.45
1.25	0.27	1.57	99.58
1.5	0.29	2.05	99.54
1.75	0.30	3.93	99.31

Table III, the proposed PPDL achieves satisfactory authentication performance even with a small time window size (0.5 s). With the increase of time window size, the authentication accuracy and the FAR fluctuate within a small range, but the FRR significantly increases. We can explain this phenomenon by the following aspects: 1) As the time window size increases, the number of positive samples gradually decreases, and the proposed PPDL is unable to learn the distribution of positive samples, resulting in an increase in the FAR; 2) Behavioral biometric data sequences are interrelated in time, and the longer behavioral biometric data sequences may not be conducive to modeling the behavioral patterns of legitimate smartphone users.

After weighing the authentication performance and the time overhead per authentication, we apply a 0.5 s time window size of behavioral biometric data sequences to generate the training and testing samples. Note that due to too many smartphone users participating in the evaluation experiments (90 smartphone users), we cannot report the detailed authentication performance of the PPDL on each smartphone user. To demonstrate the fine-grained performance of the PPDL on some smartphone users, we report the authentication performance of thirty randomly selected smartphone users with a 0.5 s time window size. Without loss of generality, we apply the *random.sample* function[1] to randomly select thirty smartphone users from the HMOG dataset, which is also used to randomly select four users for feature visualization in Section V-D2. As shown in Table IV, the proposed PPDL can achieve excellent performance with FAR (less than 1%), FRR (less than 2%), and Accuracy (more than 99%) for each randomly selected smartphone user on the HMOG dataset. We demonstrate that the proposed PPDL can effectively learn the behavioral patterns of each legitimate smartphone user.

2) Feature Visualization: To further demonstrate the proposed PPDL's ability to learn the behavioral patterns of smartphone users, we visualize the learned deep features to understand feature discriminability intuitively. Feature visualization allows us to more intuitively discover and analyze the more abstract semantic feature representations. We randomly selected four users from the HMOG dataset and projected their features into a two-dimensional space. As shown in Fig. 4, the learned deep features by the proposed PPDL can aggregate samples of the same class while also separating samples of different classes from each other. Experimental results demonstrate the effectiveness of the proposed PPDL in extracting discriminative deep features from behavioral biometric data, which can better model the behavioral patterns of smartphone users.

[1]https://docs.python.org/3.7/library/random.html

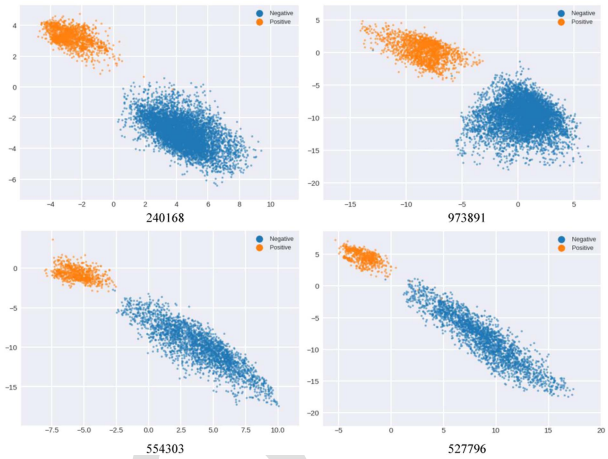


Fig. 4. The visualization of deep features from four randomly selected smartphone users. The deep features are extracted from the last layer of the CNN architecture. The positive samples are plotted in orange, and the negative samples are plotted in blue.

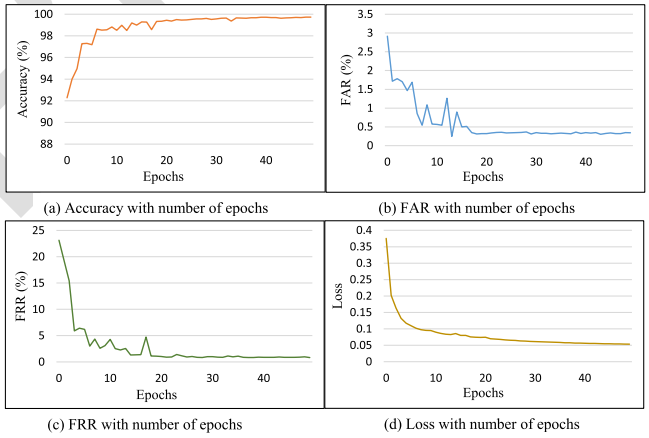


Fig. 5. Authentication performance with varying training epochs.

3) Parameter Study: We study and analyze the effect of parameter choices. What are the appropriate training epochs that yield the best trade-off between accuracy and time efficiency?

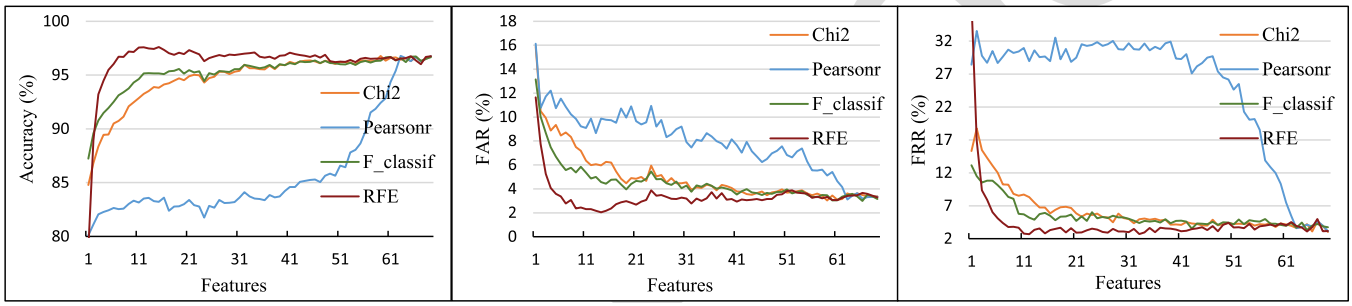
We train the proposed PPDL for 50 training epochs and report the accuracy, FAR, FRR, and loss for each epoch to analyze the impact of training epochs on authentication performance. As shown in Fig. 5(a), (b), and (c), the proposed PPDL achieves the optimal authentication performance when trained only with 20 epochs. The loss in Fig. 5(d) drops rapidly and reaches convergence after 20 training epochs. Experimental results demonstrate that the proposed PPDL can quickly converge and achieve satisfactory performance during the training phase.

E. Evaluation of the Proposed PPLSVM

Classical machine learning based continuous authentication model usually requires manually constructed features (i.e., feature engineering). As described in Section IV-D2, based on the behavioral biometric data, we manually construct 51

TABLE IV
THE DETAIL AUTHENTICATION PERFORMANCE OF THIRTY RANDOMLY SELECTED SMARTPHONE USERS WITH A 0.5 S TIME WINDOW SIZE

User ID	FAR ↓	FRR ↓	Accuracy↑	User ID	FAR↓	FRR↓	Accuracy ↑	User ID	FAR↓	FRR↓	Accuracy ↑
937904	0.24	0.75	99.64	856302	0.31	0.82	99.56	501973	0.43	1.71	99.26
240168	0.25	0.65	99.65	261313	0.33	0.61	99.6	785873	0.48	0.92	99.41
892687	0.22	0.68	99.66	201848	0.08	1.09	99.67	366286	0.58	1.23	99.26
561993	0.25	0.58	99.67	777078	0.3	0.72	99.59	733568	0.4	0.61	99.54
998757	0.17	0.82	99.67	489146	0.34	0.48	99.63	751131	0.84	1.31	99.04
277905	0.48	0.61	99.48	841866	0.25	0.44	99.7	657486	0.28	0.85	99.58
990622	0.08	0.44	99.83	893255	0.15	0.38	99.8	693572	0.29	0.44	99.67
278135	0.09	0.27	99.86	865881	0.24	0.82	99.62	815316	0.26	0.75	99.62
865501	0.63	0.82	99.32	248252	0.04	0.1	99.94	336172	0.2	0.55	99.71
808022	0.43	0.55	99.54	398248	0.19	0.55	99.72	923862	0.26	0.48	99.69



(a) Accuracy with number of features (b) FAR with number of features (c) FRR with number of features

Fig. 6. Experimental results of varying number of selected features (69) with different feature selection algorithms.

time-domain features and 18 frequency-domain features, such as mean, standard deviation, variance, skewness, kurtosis, etc. However, classical machine learning algorithms cannot extract discriminative features from manually constructed features automatically, and some constructed features may affect the authentication performance negatively. In this paper, we explore the effectiveness of different feature selection algorithms in selecting discriminative features from 69 statistical features such as Chi2,[2] Pearsonr,[3] Recursive Feature Elimination (RFE),[4] ANOVA F_classif.[5] And then, the selected discriminative features by the feature selection algorithms will be fed to the proposed PPLSVM to learn the behavioral patterns of smartphone users.

We traverse all 69 features mentioned above to select the most discriminative features via four feature selection algorithms. Fig. 6 plots the mean experimental results of 90 smartphone users according to a varying number of selected features with four feature selection algorithms. As shown in Fig. 6, the RFE algorithm achieves the best performance with fewer selected features among four feature selection algorithms. In addition, we also list the best authentication performance of four feature selection algorithms with the optimal number of selected features in Table V.

TABLE V
THE EXPERIMENTAL RESULTS OF FOUR FEATURE SELECTION ALGORITHMS WITH OPTIMAL NUMBER OF FEATURES

Algorithm	Features	FAR ↓	FRR ↓	Accuracy ↑
Chi2	59	3.03	3.76	96.78
Pearsonr	63	3.11	3.6	96.79
F_classif	66	2.99	4	96.73
RFE	15	2.16	2.82	97.61

As shown in Table V, the RFE algorithm achieves the best performance of 97.61% accuracy, 2.16% FAR, and 2.82% FRR with the selected feature number of 15. Fewer selected features mean the proposed PPLSVM will spend less time during sample generation and training. Then we use these 15 discriminative features selected by the RFE algorithm to perform the evaluation experiments with the proposed PPLSVM. The top 15 discriminative features selected by the RFE algorithm are specified in Table VI. As shown in Table VI, most manually constructed features from the accelerometer and gyroscope have richer semantic information than those from the magnetometer, indicating that behavioral biometric data from the accelerometer and gyroscope contribute more to improve authentication performance.

F. Comparison With Representative Authentication Methods

We conduct extensive experiments with representative continuous authentication work to demonstrate the superiority of

[2]https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html
[3]https://docs.scipy.org/doc/scipy-1.2.1/reference/generated/scipy.stats.pearsonr.html
[4]https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html
[5]https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_classif.html

TABLE VI
THE TOP 15 DISCRIMINATIVE FEATURES SELECTED BY THE RFE ALGORITHM

No.	Selected Features
1	Peak2f of the accelerometer data
2	Standard deviation of the accelerometer data
3	Max of the accelerometer data
4	Min of the accelerometer data
5	Coefficient of range of the accelerometer data
6	Quartile (75%) of the accelerometer data
7	Quartile (95%) of the accelerometer data
8	Peak1 of the gyroscope data
9	Peak2f of the gyroscope data
10	Skewness of the gyroscope data
11	Min of the gyroscope data
12	Quartile (25%) of the gyroscope data
13	Quartile (75%) of the gyroscope data
14	Peak2f of the magnetometer data
15	Quartile (25%) of the magnetometer data

the proposed two privacy-preserving continuous authentication models. The Acc., Gyr., Mag., To., El., and Ori. indicate the accelerometer, gyroscope, magnetometer, touch, elevation, and orientation in Tables VII and VIII.

1) Evaluation and Comparison Under the Same Conditions: Most researchers conduct performance evaluation experiments on their datasets, and the authentication models and datasets are not public to us. Therefore, we need help evaluating representative authentication models' performance on a common dataset. For fair and convincing performance evaluation and comparison, we reproduce some representative authentication models on the HMOG dataset, including a privacy-preserving continuous authentication scheme [53] and some representative continuous authentication schemes without considering privacy protection [16], [17], [18], [23], [24], [44]. Note that we cannot completely reproduce some representative privacy-preserving continuous authentication schemes because they involve complicated encryption details and encryption processes. Therefore, we can only try our best to reproduce the literature [53] in this paper. The experimental results reported here might vary slightly from those in the original papers due to undisclosed experimental details and parameter settings.

Table VII lists the data source, classifier, privacy protection scheme, and the FAR and the FRR. Compared with these representative authentication schemes, our proposed privacy-preserving continuous authentication models have the following two advantages: 1) The proposed PPDL and PPLSVM achieve superior authentication performance when these representative continuous authentication schemes are evaluated on the same dataset; 2) We train the proposed PPDL and PPLSVM without compromising data privacy of behavioral biometric data.

Note that the quantitative comparison experiments only provide preliminary evaluation results, and we should refrain from using this to demonstrate the advantages and disadvantages of different continuous authentication schemes. Different continuous authentication schemes are suitable for different scenarios. Therefore, qualitative comparisons of different scenarios and diverse datasets are necessary.

2) Comparison With Previous Work: We compare the proposed PPDL and PPLSVM to some representative continuous authentication schemes, including privacy-preserving continuous authentication schemes [42], [53], [54], [55], and some representative continuous authentication schemes without considering behavioral biometric data privacy protection [13], [14], [16], [17], [18], [23], [24], [25], [29], [44]. Sitová et al. [44] and Shen et al. [16] both proposed the classical machine learning based authentication models with manually constructed features, obtaining the authentication performance of 7.16% EER in [44] and 3.98% FAR, and 5.03% FRR in [16], respectively. To train the classical machine learning based continuous authentication model without compromising the behavioral biometric data privacy, Govindarajan et al. [55], Wei et al. [54], Hernández-Álvarez et al. [42], and Gan et al. [53] both applied encryption schemes to encrypt the behavioral biometric data that allow them to train the continuous authentication models on the ciphertext. Literature [13], [23], [24] applied different CNN architectures to extract valuable deep features from behavioral biometric data automatically, and the deep features are used to train the classical machine learning based continuous authentication models. Abuhamad et al. [29], Wang et al. [17], Shen et al. [18], and Li et al. [25] both applied the end-to-end deep learning architectures to learn and model smartphone users' behavioral patterns.

Table VIII compares our experimental settings and evaluation results against representative authentication work. Compared with these representative continuous authentication schemes without considering data privacy protection [13], [14], [16], [17], [18], [23], [24], [25], [29], [44], our proposed two privacy-preserving continuous authentication models achieve satisfactory authentication performance. More importantly, the proposed PPDL and PPLSVM consider behavioral biometric data protection, which will not compromise data privacy during training and inference. Although some authentication schemes attempt to apply encryption techniques to train authentication models on the ciphertext [42], [53], [54], [55], they are primarily basic classical machine learning based continuous authentication schemes with unsatisfactory authentication performance. Training deep learning-based models with encryption techniques, such as homomorphic encryption, remains a significant computational challenge. In this paper, we secretly share the behavioral biometric data among two cloud servers that jointly train the deep learning techniques based continuous authentication models without compromising data privacy.

G. Time Cost of Per Batch for Privacy Training and Inference

We evaluate the runtime and communication cost (MB) of the proposed PPDL and PPLSVM in the training and inference phase. Table IX lists the training samples, testing samples, batch size, training time cost, and test time cost per training batch. For the proposed PPDL, we generate 364970 training samples and 243313 testing samples for each smartphone user, and the time cost is 8.97 s for one privacy training batch, and the time cost is 5.36 s for one privacy inference batch. For the proposed PPLSVM, we generate 944070 training samples and 629380 testing samples for each smartphone user, and the time cost is

TABLE VII
THE FAR (%), FRR (%) OF DIFFERENT REPRESENTATIVE AUTHENTICATION METHODS ON THE HMOG DATASET

Work	Data Source	Classifiers[1]	Data Protection[2]	FAR	FRR
Sitová et al. (2015) [44]	Acc., Gyr.	SM	None	15.94	17.19
Shen et al. (2017) [16]	Acc., Gyr., Mag.	HMM	None	8.64	10.13
Li et al. (2020) [23]	Acc., Gyr.	OC-SVM	None	5.16	6.79
Wang et al. (2021) [17]	Acc.	Deep Metric Learning	None	3.85	5.07
Li et al. (2021) [24]	Acc., Gyr., Mag.	IF	None	4.45	6.03
Shen et al. (2022) [18]	Acc., Gyr., Mag.	DeSVDD	None	6.11	7.68
Gan et al. (2022) [53]	Acc., Gyr.	SVM	Homomorphic Encryption	1.97	10.69
PPDL	Acc., Gyr., Mag.	CNN	MPC	0.35	0.81
PPLSVM	Acc., Gyr., Mag.	Linear SVM	MPC	2.16	2.82

[1] SM: Scaled Manhattan, HMM: Hidden Markov Model, OC-SVM: One Class Support Vector Machine, IF: Isolation Forest, DeSVDD: Deep Support Vector Data Description, LOF: Local Outlier Factor.
[2] None: Without privacy protection consideration.

TABLE VIII
QUALITATIVE COMPARISON WITH REPRESENTATIVE AUTHENTICATION SCHEMES

Work	Classifiers[1]	Data Protection	Performance
Govindarajan et al. (2013) [55]	SE	Homomorphic Encryption	EER: 21.27%
Sitová et al. (2015) [44]	SM	None	EER: 7.16% Walking
Shen et al. (2017) [16]	HMM	None	FAR: 3.98%, FRR: 5.03%
Wu et al. (2020) [14]	CNN and LOF	None	FAR: 0.86%, FRR: 3.16%
Li et al. (2020) [23]	CNN and OC-SVM	None	FAR: 2.35%, FRR: 2.30%
Abuhamad et al. (2020) [29]	LSTM	None	FAR: 0.96%, FRR: 8.08%
Wei et al. (2020) [54]	Cosine Similarity	Homomorphic Encryption	EER: 17%
Wang et al. (2021) [17]	CNN	None	Mcgill: 95.3% Accuracy
Li et al. (2021) [24]	CNN and IF	None	FAR: 2.94%, FRR: 6.67%
Hernández-Álvarez et al. (2021) [42]	SVM	Format-preserving Encryption	Accuracy: 76.85%
Shen et al. (2022) [18]	DeSVDD	None	FAR: 14.4%, FRR: 15.2%
Gan et al. (2022) [53]	SVM	Homomorphic Encryption	F1-Score: 0.86
Li et al. (2023) [13]	CNN and LOF	None	FAR: 3.27%, FRR: 6.11%
Li et al. (2023) [25]	CNN	None	FAR: 0.89%, FRR: 0.97%
PPDL	CNN	MPC	FAR: 0.35%, FRR: 0.81%
PPLSVM	Linear SVM	MPC	FAR: 2.16%, FRR: 2.82%

[1] LOF: Local Outlier Factor, SE: Scaled Euclidean.

TABLE IX
THE TIME COST (S) OF THE PROPOSED TWO PRIVACY-PRESERVING CONTINUOUS AUTHENTICATION MODELS FOR ONE TRAINING AND INFERENCE BATCH

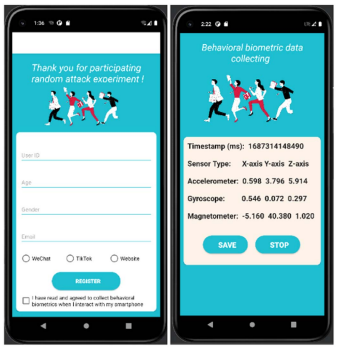
Model	Input Size	Training Samples	Testing Samples	Batch Size[1]	Training Cost (s)	Inference Cost (s)	Comm. (MB)
PPDL	12×50	364970	243313	64	8.97	5.36	96.73
PPLSVM	1×15	944070	629380	—	0.57	0.21	3.29

[1] —: The whole samples are in one batch.

0.57 s for one privacy training batch, and the time cost is 0.21 s for one privacy inference batch.

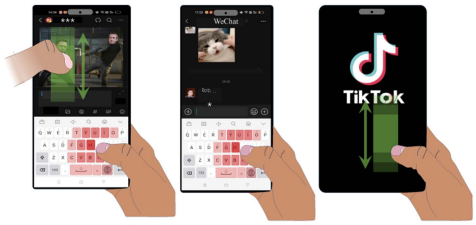
Additionally, a critical component of this process involves the communication between two servers, which inherently influences the overall system efficiency. The communication cost primarily stems from the data exchange required for maintaining privacy protocols during the training and inference phases. Though this introduces additional latency compared to a single-server setup, our optimized communication protocols ensure minimal overhead by leveraging efficient data transmission techniques and compressing communication payloads. In this section, we present the communication cost of the proposed PPDL and PPLSVM. As shown in Table IX, the communication cost of the proposed PPDL is 96.73 MB, which is primarily attributed to ReLU and pooling operations. In contrast, the communication cost of the proposed PPLSVM is only 3.29 MB. This is primarily because PPLSVM consists mostly of linear operations, which incur significantly lower communication overhead compared to the non-linear operations in PPDL.

Like most representative privacy-preserving machine learning schemes [61], [62], [64], [65], [74], [75], [76], [77], [78], [79], compared with plaintext training and inference, the secure computation operation (Section IV-B) in the CNN architecture (Section IV-E) will take more runtime and communication



(a) (b)

Fig. 7. Behavioral biometric data collection App based on Android system. (a) Participant registration and interaction scenario selection page; (b) Behavioral biometric data collection, display, and storage page.



(a) Interaction with Website (b) Interaction with WeChat (c) Interaction with TikTok

Fig. 8. Touch postures of three interaction scenarios when interacting with smartphones. (a) It involves sliding up and down for browsing information and keystroke for information search; (b) It involves frequent keystrokes for editing messages; (c) It involves sliding up and down to browse short videos.

cost. Although privacy training and inference take longer than plaintext training and inference, it is within the acceptable time cost range in practice. More importantly, privacy training and inference can protect the behavioral biometric data in the process of training and inference.

H. Random Attack Analysis

We conduct extensive experiments to evaluate the ability of the proposed PPDL and PPLSVM to resist random attacks in practice. We assume that attackers successfully crack the one-time authentication mechanism of the legitimate user's smartphone through some attack methods, such as smudge attacks [2], [3], shoulder surfing attacks [4], side channel attacks [5], and dictionary attacks [6]. Then the attackers attempt to interact with the legitimate user's smartphone with their behavioral patterns. We aim to evaluate whether the proposed privacy-preserving continuous authentication models can authenticate the attacker's identity in time.

We invite 30 smartphone users (10 females and 20 males) to participate in the random attack experiments. We informed participants of the experimental details and obtained their signed consent to collect their behavioral biometric data. We develop a data collection App based on the Android system, which can collect behavioral biometric data in the background when participants interact with their smartphones. Fig. 7 shows the behavioral biometric data collection App. As shown in Fig. 7(a), participants register and select an interaction scenario (Website, WeChat, and TikTok) to interact with their smartphones. The developed data collection App can display and record the collected behavioral biometric data, as shown in Fig. 7(b). The behavioral biometric data file (.csv) is stored at /data/data/com.example.Collection/files/ in the Android system.

The selected three scenarios (Website, WeChat, and TikTok) are common interaction scenarios in our daily life.

- *Website:* Smartphone users can search and browse interesting information through websites such as Twitter and Weibo, which involves simple sliding up and down operations and complex keystroke operations to complete information search (Fig. 8(a)).

TABLE X
THE MEAN EXPERIMENTAL RESULTS OF TWO PRIVACY-PRESERVING CONTINUOUS AUTHENTICATION MODELS UNDER RANDOM ATTACKS IN THE REAL-WORLD ENVIRONMENT

Model	Attack Scenario	FAR ↓	FRR ↓
PPDL	Website	1.61	2.25
	WeChat	1.83	2.61
	TikTok	2.47	3.01
PPLSVM	Website	3.14	3.85
	WeChat	3.64	4.13
	TikTok	4.06	4.42

- *WeChat:* It is an instant messaging software launched by Tencent, which is aimed at smartphone users. Users can share text, pictures, and stickers with friends through the client, which involves frequent keystroke operations for editing (Fig. 8(b)).
- *TikTok:* It is a short video-sharing and browsing social software, and users mainly interact with it by sliding up and down operations on the touchscreen (Fig. 8(c)).

For each round (30 rounds total), we randomly select one participant as the legitimate user, and remaining participants act as attackers. Each attacker uses three interaction scenarios (Website, WeChat, and TikTok) to interact with the legitimate user's smartphone, and we apply the data collection App (Fig. 7) to collect behavioral biometric data in the background. We collect the behavioral biometric data for more than four hours for each interaction scenario for each attacker. Table X lists the mean experimental results of the proposed PPDL and PPLSVM under random attacks. As shown in Table X, the proposed PPDL and PPLSVM can better identify attackers (lower FAR) with Website and WeChat interaction scenarios. We can explain this phenomenon in that attackers need to use frequent and more complex operations to interact with WeChat and Website applications, such as frequent keystroke operations. We demonstrate that the proposed PPDL and PPLSVM are highly resilient to random attacks in the real-world environment.

VI. SECURITY ANALYSIS

In this section, we analyze the security of our proposed privacy-preserving continuous authentication scheme, focusing on data privacy, protocol security, and communication security.

A. Privacy of Behavioral Biometric Data

The core privacy guarantee of our continuous authentication system stems from the information-theoretic security of the additive secret sharing protocol. For any sensitive biometric data x, the client generates two shares $\langle x\rangle_0 = r$ and $\langle x\rangle_1 = x - r \pmod{2^k}$, where r is a random number uniformly distributed in $\mathbb{Z}_{2^k}$. Since r is chosen independently of x, each individual share appears as random noise to the service providers S_0 and S_1. Consequently, without access to both shares, it is mathematically impossible for an individual untrusted server to reconstruct the original behavioral biometric data x.

B. Security Under the Semi-Honest Model

Following the standard security definitions for MPC, we prove the security of our joint training and inference process under the semi-honest (honest-but-curious) model. 1) non-collusion assumption: We assume S_0 and S_1 are non-colluding. In this setting, the internal states and messages exchanged during the protocol do not reveal any information beyond the final result. 2) multiplication security: Linear operations are performed locally on secret shares. For multiplication, we utilize beaver triples generated via the semantically secure paillier cryptosystem. The use of random masks in the online multiplication phase ensures that servers only see masked values, which are indistinguishable from random elements. 3) non-linear operation security: For non-linear layers like ReLU and Maxpool, we employ a secure comparison protocol based on MSB extraction. This protocol allows S_0 and S_1 to determine the sign bit of a shared value without revealing the magnitude or the actual value itself.

C. Communication Security

All communication channels between the client and the cloud servers, as well as between S_0 and S_1, are protected by secure protocols (e.g., TLS/SSL). This prevents external eavesdroppers from capturing secret shares or intermediate computation results through sniffing attacks.

VII. Discussion

A. A New Direction for Designing Privacy-Preserving Continuous Authentication Scheme

Smartphones must rely on powerful cloud servers for deep learning-based continuous authentication because their local hardware cannot handle the massive training load. Nevertheless, offloading sensitive behavioral biometric data to a cloud server introduces significant privacy vulnerabilities, as the integrity and trustworthiness of the server cannot always be guaranteed. By compromising cloud servers, attackers can obtain users' behavioral biometric data, which research indicates can be leveraged to infer highly sensitive personal attributes such as typing patterns, demographics, age, and gender. Existing deep learning techniques based continuous authentication models pay more attention to improving authentication performance, but no measures were provided to guarantee data privacy. To solve this issue, we propose a privacy-preserving continuous authentication scheme based on secret sharing MPC protocol, which allows us to train deep learning techniques based continuous authentication models on untrusted cloud servers without compromising data privacy. Although some continuous authentication schemes attempt to apply encryption techniques to train continuous authentication models on the ciphertext [42], [53], [54], [55], they only can train continuous authentication models based on traditional machine learning, which usually achieve unsatisfactory authentication performance. Compared with these continuous authentication schemes, our proposed privacy-preserving continuous authentication scheme can combine privacy-preserving protocols with deep learning to improve authentication performance and ensure privacy guarantee. We believe this work makes a solid step towards introducing data privacy-preserving into deep learning based continuous authentication schemes. However, we also recognize that privacy-invasive nature of activity tracking remains a critical issue associated with biometric modality indices. In our future work, we are committed to investigating more privacy-preserving schemes to specifically address these privacy-invasive concerns regarding user activities.

B. Random Attacks Analysis in Real-World Environment

Past work [8], [9], [10], [11], [13], [16], [18], [19], [29], [44] evaluate the effectiveness and authentication performance of the proposed continuous authentication schemes on their datasets while ignoring the ability to resist various random attacks in the real-world environment. Section V-H examined the authentication performance of the proposed PPDL and PPLSVM under random attacks in the real-world environment. We develop a data collection App based on the Android system (Fig. 7), which can collect smartphone users' behavioral biometric data when interacting with their smartphones. In the random attacks experiments, the attackers select the interaction scenarios (Website, WeChat, and TikTok) to interact with legitimate users' smartphones (Fig. 8). Experimental results demonstrate that the proposed PPDL and PPLSVM are highly resilient to random attacks in the real-world environment.

VIII. Conclusion

In this paper, we propose a privacy-preserving continuous authentication scheme, which secretly shares the behavioral biometric data among two untrusted cloud servers that train the continuous authentication models on the joint data via secret sharing MPC. We evaluate the practical feasibility of our proposed privacy-preserving scheme on two constructed privacy-preserving continuous authentication models (PPDL and PPLSVM). Experimental results demonstrate that they achieve satisfactory authentication performance without violating the privacy of behavioral biometric data. Moreover, random attack experiments demonstrate that the proposed PPDL and PPLSVM are highly resilient to random attacks from diverse attackers in the real-world environment. We believe this work provides a brand new technical route for designing privacy-preserving deep learning techniques based continuous authentication models.

REFERENCES

[1] S. Egelman, S. Jain, R. S. Portnoff, K. Liao, S. Consolvo, and D. Wagner, "Are you ready to lock?," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2014, pp. 750–761.

[2] A. J. Aviv, K. L. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," *Woot*, vol. 10, pp. 1–7, 2010.

[3] S. Cha, S. Kwag, H. Kim, and J. H. Huh, "Boosting the guessing attack performance on android lock patterns with smudge attacks," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2017, pp. 313–326.

[4] G. D. Clark and J. Lindqvist, "Engineering gesture-based authentication systems," *IEEE Pervasive Comput.*, vol. 14, no. 1, pp. 18–25, 1st Quart., 2015.

[5] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "Accessory: Password inference using accelerometers on smartphones," in *Proc. 12th Workshop Mobile Comput. Syst. Appl.*, 2012, pp. 1–6.

[6] L. Bošnjak, J. Sreš, and B. Brumen, "Brute-force and dictionary attack on hashed real-world passwords," in *Proc. 41st Int. Conv. Inf. Commun. Technol. Electron. Microelectronics*, 2018, pp. 1161–1166.

[7] M. M. Koushki, B. Obada-Obieh, J. H. Huh, and K. Beznosov, "On smartphone users' difficulty with understanding implicit authentication," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2021, pp. 1–14.

[8] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 1, pp. 136–148, Jan. 2013.

[9] T. Zhu et al., "RiskCog: Unobtrusive real-time user authentication on mobile devices in the wild," *IEEE Trans. Mobile Comput.*, vol. 19, no. 2, pp. 466–483, Feb. 2020.

[10] Y. Li, P. Tao, S. Deng, and G. Zhou, "DeFFusion: CNN-based continuous authentication using deep feature fusion," *ACM Trans. Sensor Netw.*, vol. 18, no. 2, pp. 1–20, 2021.

[11] A. Ray-Dowling, D. Hou, S. Schuckers, and A. Barbir, "Evaluating multi-modal mobile behavioral biometrics using public datasets," *Comput. Secur.*, vol. 121, 2022, Art. no. 102868.

[12] Y. Li, H. Hu, and G. Zhou, "Using data augmentation in continuous authentication on smartphones," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 628–640, Feb. 2019.

[13] Y. Li, J. Luo, S. Deng, and G. Zhou, "SearchAuth: Neural architecture search based continuous authentication using auto augmentation search," *ACM Trans. Sensor Netw.*, vol. 19, 2023, Art. no. 92.

[14] C. Wu, K. He, J. Chen, Z. Zhao, and R. Du, "Liveness is not enough: Enhancing fingerprint authentication with behavioral biometrics to defeat puppet attacks," in *Proc. USENIX Secur. Symp.*, 2020, pp. 2219–2236.

[15] Y. Jiang, H. Zhu, S. Chang, and B. Li, "MAUTH: Continuous user authentication based on subtle intrinsic muscular tremors," *IEEE Trans. Mobile Comput.*, vol. 23, no. 2, pp. 1930–1941, Feb. 2024.

[16] C. Shen, Y. Li, Y. Chen, X. Guan, and R. A. Maxion, "Performance analysis of multi-motion sensor behavior for active smartphone authentication," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 1, pp. 48–62, Jan. 2018.

[17] C. Wang, Y. Xiao, X. Gao, L. Li, and J. Wang, "A framework for behavioral biometric authentication using deep metric learning on mobile devices," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 19–36, Jan. 2023.

[18] Z. Shen, S. Li, X. Zhao, and J. Zou, "MMAuth: A continuous authentication framework on smartphones using multiple modalities," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 1450–1465, 2022.

[19] Y. Li, L. Liu, H. Qin, S. Deng, M. A. El-Yacoubi, and G. Zhou, "Adaptive deep feature fusion for continuous authentication with data augmentation," *IEEE Trans. Mobile Comput.*, vol. 22, no. 10, pp. 5690–5705, Oct. 2023.

[20] C. Wu et al., "Use it-no need to shake it! accurate implicit authentication for everyday objects with smart sensing," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 6, no. 3, pp. 1–25, 2022.

[21] Y. Song and Z. Cai, "Integrating handcrafted features with deep representations for smartphone authentication," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 6, no. 1, pp. 1–27, 2022.

[22] M. Hu, K. Zhang, R. You, and B. Tu, "AuthConFormer: Sensor-based continuous authentication of smartphone users using a convolutional transformer," *Comput. Secur.*, vol. 127, 2023, Art. no. 103122.

[23] Y. Li, H. Hu, Z. Zhu, and G. Zhou, "SCANet: Sensor-based continuous authentication with two-stream convolutional neural networks," *ACM Trans. Sensor Netw.*, vol. 16, no. 3, pp. 1–27, 2020.

[24] Y. Li, J. Luo, S. Deng, and G. Zhou, "CNN-based continuous authentication on smartphones with conditional Wasserstein generative adversarial network," *IEEE Internet Things J.*, vol. 9, no. 7, pp. 5447–5460, Apr. 2022.

[25] Y. Li, L. Liu, S. Deng, H. Qin, M. A. El-Yacoubi, and G. Zhou, "Memory-augmented autoencoder based continuous authentication on smartphones with conditional transformer GANs," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 4467–4482, May 2024.

[26] S. Amini, V. Noroozi, A. Pande, S. Gupte, P. S. Yu, and C. Kanich, "DeepAuth: A framework for continuous user re-authentication in mobile apps," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 2027–2035.

[27] Q. Zou, Y. Wang, Q. Wang, Y. Zhao, and Q. Li, "Deep learning-based gait recognition using smartphones in the wild," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3197–3212, 2020.

[28] L. Sun et al., "KOLLECTOR: Detecting fraudulent activities on mobile devices using deep learning," *IEEE Trans. Mobile Comput.*, vol. 20, no. 4, pp. 1465–1476, Apr. 2021.

[29] M. Abuhamad, T. Abuhmed, D. Mohaisen, and D. Nyang, "AUToSen: Deep-learning-based implicit continuous authentication using smartphone sensors," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5008–5020, Jun. 2020.

[30] G. Giorgi, A. Saracino, and F. Martinelli, "Using recurrent neural networks for continuous authentication through gait analysis," *Pattern Recognit. Lett.*, vol. 147, pp. 157–163, 2021.

[31] D. X. Song, D. Wagner, and X. Tian, "Timing analysis of keystrokes and timing attacks on SSH," in *Proc. 10th USENIX Secur. Symp.*, 2001, Art. no. 25.

[32] K. Zhang and X. Wang, "Peeping tom in the neighborhood: Keystroke eavesdropping on multi-user systems," in *Proc. USENIX Secur. Symp.*, 2009, Art. no. 23.

[33] H. Wang, T. T.-T. Lai, and R. Roy Choudhury, "MoLe: Motion leaks through smartwatch sensors," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 155–166.

[34] D. G. Brizan, A. Goodkind, P. Koch, K. Balagani, V. V. Phoha, and A. Rosenberg, "Utilizing linguistically enhanced keystroke dynamics to predict typist cognition and demographics," *Int. J. Hum.- Comput. Stud.*, vol. 82, pp. 57–68, 2015.

[35] A. Pentel, "Predicting age and gender by keystroke dynamics and mouse patterns," in *Proc. 25th Conf. User Model. Adapt. Personalization*, 2017, pp. 381–385.

[36] R. Giot and C. Rosenberger, "A new soft biometric approach for keystroke dynamics based on gender recognition," *Int. J. Inf. Technol. Manage.*, vol. 11, no. 1/2, pp. 35–49, 2012.

[37] A. Acar et al., "A lightweight privacy-aware continuous authentication protocol-paca," *ACM Trans. Privacy Secur.*, vol. 24, no. 4, pp. 1–28, 2021.

[38] Y. Sun and S. Upadhyaya, "Secure and privacy preserving data processing support for active authentication," *Inf. Syst. Front.*, vol. 17, pp. 1007–1015, 2015.

[39] J. Hatin, E. Cherrier, J.-J. Schwartzmann, and C. Rosenberger, "Privacy preserving transparent mobile authentication," in *Proc. 3rd Int. Conf. Inf. Syst. Secur. Privacy*, 2017, pp. 354–361.

[40] M. Al-Rubaie, "Towards privacy-aware mobile-based continuous authentication systems," PhD dissertation, Iowa State Univ., Ames, IA, USA, 2018. Q2

[41] P. Gasti, J. Šeděnka, Q. Yang, G. Zhou, and K. S. Balagani, "Secure, fast, and energy-efficient outsourced authentication for smartphones," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 11, pp. 2556–2571, Nov. 2016.

[42] L. Hernández-Álvarez, J. M. De Fuentes, L. González-Manzano, and L. H. Encinas, "SmartCAMPP-Smartphone-based continuous authentication leveraging motion sensors with privacy preservation," *Pattern Recognit. Lett.*, vol. 147, pp. 189–196, 2021.

[43] D. Buschek, A. De Luca, and F. Alt, "Improving accuracy, applicability and usability of keystroke biometrics on mobile touchscreen devices," in *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst.*, 2015, pp. 1393–1402.

[44] Z. Sitová et al., "HMOG: New behavioral biometric features for continuous authentication of smartphone users," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 5, pp. 877–892, May 2016.

[45] M. Hu, K. Zhang, R. You, and B. Tu, "Multi-sensor-based continuous authentication of smartphone users with two-stage feature extraction," *IEEE Internet Things J.*, vol. 10, no. 6, pp. 4708–4724, Mar. 2023.

[46] C. Zhao, F. Gao, and Z. Shen, "Multi-motion sensor behavior based continuous authentication on smartphones using gated two-tower transformer fusion networks," *Comput. Secur.*, vol. 139, 2024, Art. no. 103698.

[47] L. Ruff et al., "Deep one-class classification," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4393–4402.

[48] M. Hu, D. Wang, C. Li, Y. Xu, and B. Tu, "Behavioral biometrics-based continuous authentication using a lightweight latent representation masked one-class autoencoder," *IEEE Trans. Dependable Secure Comput.*, vol. 22, no. 3, pp. 1908–1923, May/Jun. 2025.

[49] Z. Yang, Y. Li, and G. Zhou, "Unsupervised sensor-based continuous authentication with low-rank transformer using learning-to-rank algorithms," *IEEE Trans. Mobile Comput.*, vol. 23, no. 9, pp. 8839–8854, Sep. 2024.

[50] Y. Li, C. Ouyang, and H. Huang, "AEGANAuth: Autoencoder GAN-based continuous authentication with conditional variational autoencoder generative adversarial network," *IEEE Internet Things J.*, vol. 11, no. 16, pp. 27635–27650, Aug. 2024.

[51] M. Jakobsson et al., "Implicit authentication for mobile devices," in *Proc. 4th USENIX Conf. Hot Top. Secur.*, 2009, pp. 25–27.

[52] N. A. Safa, R. Safavi-Naini, and S. F. Shahandashti, "Privacy-preserving implicit authentication," in *Proc. 29th IFIP Int. Inf. Secur. Conf.*, 2014, pp. 471–484.

[53] W. Gan et al., "Multi-device continuous authentication mechanism based on homomorphic encryption and SVM algorithm," in *Proc. 8th Int. Conf. Artif. Intell. Secur.*, 2022, pp. 625–638.

[54] F. Wei, P. Vijayakumar, N. Kumar, R. Zhang, and Q. Cheng, "Privacy-preserving implicit authentication protocol using cosine similarity for Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5599–5606, Apr. 2021.

[55] S. Govindarajan, P. Gasti, and K. S. Balagani, "Secure privacy-preserving protocols for outsourcing continuous authentication of smartphone users with touch data," in *Proc. IEEE 6th Int. Conf. Biometrics: Theory Appl. Syst.*, 2013, pp. 1–8.

[56] A. F. Baig, S. Eskeland, and B. Yang, "Privacy-preserving continuous authentication using behavioral biometrics," *Int. J. Inf. Secur.*, vol. 22, no. 6, pp. 1833–1847, 2023.

[57] D. Monschein and O. P. Waldhorst, "Optimizing privacy-preserving continuous authentication of mobile devices," in *Proc. Int. Conf. Netw. Syst. Secur.*, 2024, pp. 63–81.

[58] S. Jacob, P. Vinod, and V. G. Menon, "CAPP: Context-aware privacy-preserving continuous authentication in smartphones," *Secur. Privacy*, vol. 8, no. 2, 2025, Art. no. e478.

[59] Z. Huang, W.-j. Lu, C. Hong, and J. Ding, "Cheetah: Lean and fast secure two-party deep neural network inference," in *Proc. 31st USENIX Secur. Symp.*, 2022, pp. 809–826.

[60] D. Rathee, A. Bhattacharya, R. Sharma, D. Gupta, N. Chandran, and A. Rastogi, "SecFloat: Accurate floating-point meets secure 2-party computation," in *Proc. IEEE Symp. Secur. Privacy*, 2022, pp. 576–595.

[61] J.-L. Watson, S. Wagh, and R. A. Popa, "Piranha: A GPU platform for secure computation," in *Proc. 31st USENIX Secur. Symp.*, 2022, pp. 827–844.

[62] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, "GAZELLE: A low latency framework for secure neural network inference," in *Proc. 27th USENIX Secur. Symp.*, 2018, pp. 1651–1669.

[63] J. Liu, M. Juuti, Y. Lu, and N. Asokan, "Oblivious neural network predictions via MiniONN transformations," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 619–631.

[64] P. Mohassel and P. Rindal, "ABY3: A mixed protocol framework for machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2018, pp. 35–52.

[65] M. S. Riazi, C. Weinert, O. Tkachenko, E. M. Songhori, T. Schneider, and F. Koushanfar, "Chameleon: A hybrid secure computation framework for machine learning applications," in *Proc. Asia Conf. Comput. Commun. Secur.*, 2018, pp. 707–721.

[66] Q. Pang, J. Zhu, H. Möllering, W. Zheng, and T. Schneider, "BOLT: Privacy-preserving, accurate and efficient inference for transformers," in *Proc. IEEE Symp. Secur. Privacy*, 2024, pp. 4753–4771.

[67] N. Jawalkar, K. Gupta, A. Basu, N. Chandran, D. Gupta, and R. Sharma, "Orca: FSS-based secure training and inference with GPUs," in *Proc. IEEE Symp. Secur. Privacy*, 2024, pp. 597–616.

[68] D. Rathee, A. Bhattacharya, D. Gupta, R. Sharma, and D. Song, "Secure floating-point training," in *Proc. 32nd USENIX Secur. Symp.*, 2023, pp. 6329–6346.

[69] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[70] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, 1999, pp. 223–238.

[71] D. Beaver, "Efficient multiparty protocols using circuit randomization," in *Proc. Annu. Int. Cryptol. Conf.*, Springer, 1992, pp. 420–432.

[72] M. Agrawal, P. Mehrotra, R. Kumar, and R. R. Shah, "GANTouch: An attack-resilient framework for touch-based continuous authentication system," *IEEE Trans. Biom., Behav., Ident. Sci.*, vol. 4, no. 4, pp. 533–543, Oct. 2022.

[73] Y.-T. Yeh, F. Matulic, and D. Vogel, "Phone sleight of hand: Finger-based dexterous gestures for physical interaction with mobile phones," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2023, pp. 1–19.

[74] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 201–210.

[75] P. Mohassel and Y. Zhang, "SecureML: A system for scalable privacy-preserving machine learning," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 19–38.

[76] N. Kumar, M. Rathee, N. Chandran, D. Gupta, A. Rastogi, and R. Sharma, "CrypTFlow: Secure tensorflow inference," in *Proc. IEEE Symp. Secur. Privacy*, 2020, pp. 336–353.

[77] S. Sav et al., "POSEIDON: Privacy-preserving federated neural network learning," in *Proc. 28th Annu. Netw. Distrib. Syst. Secur. Symp.*, 2021. **Q3**

[78] S. Tan, B. Knott, Y. Tian, and D. J. Wu, "CryptGPU: Fast privacy-preserving machine learning on the GPU," in *Proc. IEEE Symp. Secur. Privacy*, 2021, pp. 1021–1038.

[79] H. Tian et al., "Sphinx: Enabling privacy-preserving online learning over the cloud," in *Proc. IEEE Symp. Secur. Privacy*, 2022, pp. 2487–2501.