FAMOS: Robust Privacy-Preserving Authentication on Payment Apps via Federated Multi-Modal Contrastive Learning

Yifeng Cai, *Key Laboratory of High Confidence Software Technologies (PKU), Ministry of Education; School of Computer Science, Peking University;* Ziqi Zhang, *Department of Computer Science, University of Illinois Urbana-Champaign;* Jiaping Gui, *School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University;* Bingyan Liu, *School of Computer Science, Beijing University of Posts and Telecommunications;* Xiaoke Zhao, Ruoyu Li, and Zhe Li, *Ant Group;* Ding Li, *Key Laboratory of High Confidence Software Technologies (PKU), Ministry of Education; School of Computer Science, Peking University*

https://www.usenix.org/conference/usenixsecurity24/presentation/cai-yifeng

This paper is included in the Proceedings of the 33rd USENIX Security Symposium.

August 14–16, 2024 • Philadelphia, PA, USA

978-1-939133-44-1

Open access to the Proceedings of the 33rd USENIX Security Symposium is sponsored by USENIX.

FAMOS: Robust Privacy-Preserving Authentication on Payment Apps via Federated Multi-Modal Contrastive Learning

Yifeng Cai[1,2], Ziqi Zhang[3], Jiaping Gui[4], Bingyan Liu[5], Xiaoke Zhao[6], Ruoyu Li[6], Zhe Li[6], and Ding Li[1,2]

[1]*Key Laboratory of High Confidence Software Technologies (PKU), Ministry of Education*
[2]*School of Computer Science, Peking University*
[3]*Department of Computer Science, University of Illinois Urbana-Champaign*
[4]*School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University*
[5]*School of Computer Science, Beijing University of Posts and Telecommunications*
[6]*Ant Group*

Abstract

The rise of mobile payment apps necessitates robust user authentication to ensure legitimate user access. Traditional methods, like passwords and biometrics, are vulnerable once a device is compromised. To overcome these limitations, modern solutions utilize sensor data to achieve user-agnostic and scalable behavioral authentication. However, existing solutions face two problems when deployed to real-world applications. First, it is not robust to noisy background activities. Second, it faces the risks of privacy leakage as it relies on centralized training with users' sensor data.

In this paper, we introduce FAMOS, a novel authentication framework based on federated multi-modal contrastive learning. The intuition of FAMOS is to fuse multi-modal sensor data and cluster the representation of one user's data by the action category so that we can eliminate the influence of background noise and guarantee the user's privacy. Furthermore, we incorporate FAMOS with federated learning to enhance performance while protecting users' privacy. We comprehensively evaluate FAMOS using real-world datasets and devices. Experimental results show that FAMOS is efficient and accurate for real-world deployment. FAMOS has an F1-Score of 0.91 and an AUC of 0.97, which are 42.19% and 27.63% higher than the baselines, respectively.

1 Introduction

Mobile payment apps [5], while enhancing transaction convenience, pose challenges in user authentication due to the prevalent device-sharing habit among friends and family. Traditional security measures like passwords and two-factor authentication occasionally fail, as evidenced by incidents of unauthorized transactions by minors using their parents' devices [66]. These issues underscore the need for innovative, non-intrusive authentication methods. While facial recognition via the front-facing camera is a potential solution [8], its intrusiveness makes it unpopular, particularly in public settings, prompting the exploration for more user-friendly authentication alternatives.

Recently, AuthentiSense [23] and KedyAuth [30] emerge as the state-of-the-art (SOTA) solution for transparent and user-friendly authentication. They utilize build-in sensors to capture distinctive user behavioral patterns in an imperceptible manner. These solution can prevent unauthorized payments and do not harm user experience.

Albeit the pioneering contribution of existing approaches, there are two limitations that impede their deployment in Alipay, a widely used payment application with one billion users, that we conduct experiments on in this paper. **The first limitation** is the negative influence of *background activities*[1] on sensor readings in real-world scenarios. They assume that users should keep stationary and similar actions while using mobile apps. Thus, they can detect subtle sensor patterns to differentiate individual users. However, in real-world usage, users are not always stationary. They may use their smartphones while walking or lying on a bed. These background activities introduce substantial noise to sensor readings, making it difficult to observe the subtle user-specific patterns and significantly degrading the accuracy of authentication. **The second limitation** is the violation of user privacy in the sensor data. The training strategy of existing approaches requires collecting users' sensor data and uploading all data to a remote server because the model must be trained in a centralized manner. However, collecting and uploading such data is not allowed by regulations (e.g., GDPR [62] and PIPL [16]). Thus many mobile apps opt not to collect users' sensor data to mitigate privacy concerns [75]. Therefore, it is not feasible to use them in real-world scenarios.

Addressing the two aforementioned limitations is inherently difficult. For the first limitation, how to deal with low signal-to-noise ratio data remains an open challenge in machine learning [31, 50]. In real-world scenarios, the noise introduced by background activities can overshadow the subtle user-specific features by several orders of magnitude [14, 57]. Isolating these subtle but distinct features from the massive dominant noise sources is difficult, if not impossible. For the

[1]We define background activities as the various conditions under which users are using their phones.

second limitation, although Federated Learning (FL) complies with privacy regulations, this technique is not compatible with the model design of existing approaches. This is because they require both positive (sensor data of the target user) and negative (sensor readings of other users) samples to train the model. This requirement inevitably shares sensor data across users, thus violating the basic privacy assumption of FL. Therefore, we need a novel solution to make user-transparent authentication more practical.

In this paper, we propose FAMOS, FederAted Multi-MOdal ContraStive Learning, to perform user-transparent authentication in noisy background activities while complying with privacy regulations. The key insight of FAMOS is that *by 1) fusing multi-modal sensor data and 2) clustering one user's data representation by action categories, we can effectively eliminate background activities and achieve user authentication without training data from other users.*

By fusing multi-modal sensors, we enhance user authentication by utilizing stable sensor data to counteract noise from unstable sensors. For example, while walking may affect accelerometer data, touch screen sensor readings remain consistent, allowing for accurate user authentication by combining these data sources. Conversely, when a user is lying down, the accelerometer provides more reliable data due to unstable hand movements affecting the touch screen sensor. This sensor data fusion aligns with FL paradigm, enabling training the user authentication only with their own data to ensure the privacy regulations.

However, we encountered three technical challenges while implementing our insights. The first challenge is how to identify the stable sensors under different background activities. To address this challenge, we propose an attention mechanism that dynamically assigns different weights to different sensors based on each sensor's stability. This attention mechanism can automatically select stable sensors under different background activities. The second challenge is the intricacy of clustering the data representations by their respective action category. Specifically, clustering representations of identical actions closely while ensuring that representations of different actions are distanced poses a difficulty. To tackle this challenge, we introduce an action-aware contrastive learning strategy. By leveraging the combination of cross-entropy loss and contrastive loss, we can effectively project varied actions into distinct and uniformly distributed clusters within the representation space. The third challenge is that, due to the limited user-side computation resources, the authentication model should be small and efficient enough to be deployed to user devices. To address this challenge, we adopt a residual DNN model architecture [28] to project representatives from fused features. The proposed architecture is compact enough to be deployed in TrustZone [33,52] of mobile devices, further boosting user privacy and framework security [72].

We conducted a comprehensive evaluation of FAMOS using a real-world dataset collected from Alipay and an in-lab dataset. Our results demonstrate that, under realistic background noises, FAMOS achieves an F1-Score of 0.91 and an AUC of 0.97 for user authentication. On the contrary, the highest F1-Score and AUC of baselines are only 0.64 and 0.76. FAMOS outperforms baselines by 42.19% and 27.63% for F1-Score and AUC, respectively. Compared with using only one sensor, FAMOS improves the F1-Score by up to 31.88%, which demonstrates the effectiveness of the multi-modal sensor scheme. We also deploy FAMOS in the TrustZone to evaluate the on-device efficiency. On average, the memory consumption of FAMOS is only 8.58 MB for training and 3.95 MB for inference. CPU utilization increased by 23.77% during training and 16.36% during inference. Besides, the increase in battery consumption per hour is 1.32% for training and 0.48% for inference. The averaged training and inference times for FAMOS are 26.9 minutes and 135 milliseconds, respectively. It means that FAMOS is lightweight enough and can be deployed in real-world smartphones.

We summarize our contributions as follows:

- We identify two practical limitations, the influence of background noises and privacy violations, that impede the deployment of SOTA user authentication solutions to real-world applications.
- We propose a novel authentication framework, FAMOS, based on federated multi-modal contrastive learning. FAMOS can robustly authenticate users under noisy background activities without compromising user privacy.
- We comprehensively evaluate FAMOS using real-world datasets and devices. Experimental results demonstrate the robustness and efficiency of FAMOS in real-world scenarios.

2 Background and Motivation

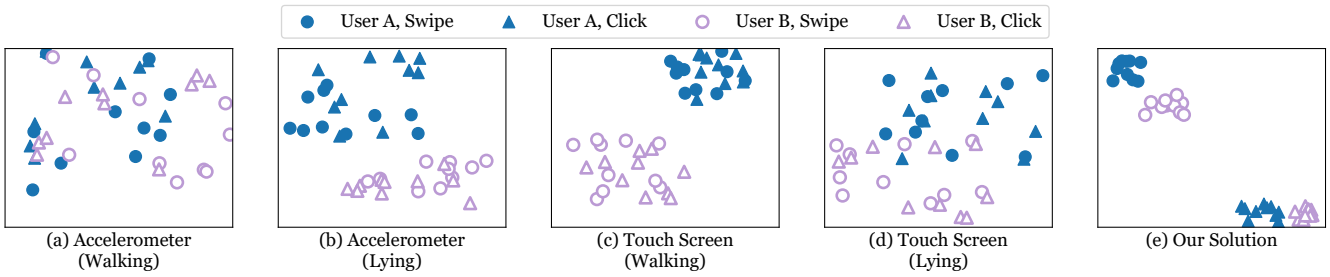
In this section, we will discuss the background and the motivation of this paper with realistic data.

2.1 The Challenge of Background Noise

User-transparent authentication leverages the unique action patterns of users, discernible in sensor readings during mobile app interactions, for authentication [3, 10, 18, 23]. For example, variations in click strength produce distinct accelerometer vibrations [9, 25], enabling distinguishing users without hindering smartphone usage.

However, the use of sensor data in practice faces a significant challenge: *sensor readings are unstable due to noise from background activities* [17]. For instance, when users are walking, the accelerometer readings exhibit massive noise due to the larger body movements of walking. In contrast, when users lie down, the pressure differences on the touch screen become more significant, resulting in greater disparities between data.

Background activities can disrupt sensor readings, making distinguishing users challenging. We illustrate this with t-SNE



(a) Accelerometer (Walking) (b) Accelerometer (Lying) (c) Touch Screen (Walking) (d) Touch Screen (Lying) (e) Our Solution

Figure 1: The t-SNE representations of sensor readings.

visualizations [60] of Click and Swipe actions from two users in Figure 1, represented by blue solid (User A) and purple hollow (User B) markers, after converting sensor signals to embedding vectors via a CNN encoder [15]. Figure 1 (a) and (b) show the visualizations under walking and lying conditions show that users' accelerometer data overlap significantly during walking, indicating instability, but are distinct when lying, reflecting the relative stability of sensor readings in this state.

Existing research often overlooks the variances in sensor data due to background activities, which can restrict the practicality and user experience of interactive systems. Different sensors exhibit varying levels of stability across activities. As shown in Figure 1 (c) and (d), touch screen sensors perform better when users are walking but less stable when users are lying. This variability highlights the impracticality of manually selecting a universally stable sensor, underscoring the need for an automated system to determine the most reliable sensors for any given sensor.

Our key insight to address background noise is to *fuse different sensors*. We take advantage of the attention mechanism to automatically identify the stable sensors for different background activities. To further validate the effectiveness of sensor fusion, we use our proposed approach to generate feature vectors by combining sensor readings and visualize the fused feature vectors of the two users in Figure 1 (e). We can observe that the fused sensor readings are clearly clustered into different groups, and thus the two users can be accurately differentiated.

2.2 Privacy Challenge

Existing work has recognized sensor data as a privacy-sensitive resource and thus cannot be uploaded to centralized servers [23, 75]. Sensor data are private because they can be used to predict private user attributes such as age and gender [51]. Besides, regulations such as China's PIPL [16] and EU's GDPR [62] consider sensor data as personal data and do not allow applications to upload such data.

FL [47] is an effective privacy-preserving training approach for user authentication. Specifically, FL allows participants to share learned models without uploading raw sensitive data, thus enabling the collaborative training of multiple users. Thus, we can employ FL to train local models for each user and mitigate the risk of privacy breaches. However, the latest works [23, 30] cannot support FL framework because such work requires data from other users as negative samples to train the model. This requirement violates the constraint of FL because it inevitably shares data across different users.

3 Threat Model

We consider two stakeholders in our system model. One stakeholder is end *users* who utilize mobile apps to access online services such as payment and money transfer. The other stakeholder is a central *server* responsible for user authentication and the provision of online services. However, there are potential attackers who seek to compromise a user's mobile device or acquire their credentials. To achieve this goal, the attacker may exploit vulnerabilities inside the mobile device or induce the user to install malware [4, 11, 58]. The attacker can also be children who gain access to the parents' credentials [1]. Once the attacker successfully acquires the user's credentials, they can impersonate the user and deceive the server during the authentication process. As a result, the attacker has unauthorized access to downstream services, including those that are typically sensitive and rely on user authentication (e.g., bank service [34]).

In our threat model, we follow prior work to make assumptions on the attacks and defenses of user authentication [23]:

- The device is used by the owner in most cases but maybe occasionally accessed by the attacker [23, 30].
- The attack can get the credentials of the victim through technical or non-technical approaches.
- The sensor data is highly related to sensitive personal attributes, including a user's age, gender, and even potential health conditions. Thus, the collected data is protected by secure hardware such as TrustZone [33, 52], to prevent possible injection attacks or data leaks.
- The authentication model can be jointly trained by the user and the application server. We assume that the model is trustworthy since sensor data can be assigned exclusively to the model in the secure world. Even if the device is compromised, the malicious attacker cannot read the model from the secure world. Therefore, the authentication model is immune to typical model backdoor or poisoning attacks [26, 56].
- The server is trustworthy, adhering to the training protocol

to securely store models and execute computations [73]. The server can also invoke advanced authentication techniques (e.g., live-video face authentication) to verify that the collected data during a specific time period is from the victim him/herself.

• The communication channel is secure and trusted because it can be protected by a standard secure communication protocol (e.g., SSL/TLS [36]). Attackers cannot manipulate transmitted data or conduct attacks such as the MITM attack [23].

4 Design of FAMOS

We now introduce FAMOS, a new user-transparent authentication system designed for real-world applications, including mobile payment platforms like Alipay. We aim to achieve the following goals:

• **G1: Robustness.** FAMOS should be robust to noises introduced by background activities. For example, when a user is walking, accelerometer readings can be noisy. FAMOS should effectively filter out this noise and extract pertinent features for authentication purposes.

• **G2: Lightweight.** FAMOS should be deployable within the TrustZone of users' smartphones, safeguarding the model from potential theft by malicious apps. Additionally, FAMOS should ensure model security under attacks.

• **G3: Privacy-Preserving.** Since the sensor data constitutes sensitive user privacy, FAMOS should prioritize user privacy, minimizing the risk of data leakage. For instance, the raw sensor readings should not be uploaded to the server.

We propose multiple approaches to achieve the above three goals. For **G1**, we propose to fuse multi-modal sensors to extract more information and attenuate the side effects of noises. Specifically, we utilize the attention mechanism and contrastive learning to better fuse features and cluster representations.

For **G2**, we propose to design a residual DNN model architecture to achieve good performance while minimizing the model size. Thus, during deployment, both the model and sensor readings can be secured by TrustZone on devices. This solution can effectively prevent malicious attackers from potentially tampering with the model.

For **G3**, we propose to leverage the FL paradigm, which ensures local storage of private data and privacy-preserving model training while ensuring the model accuracy. We only upload the learned model to the server to aggregate the learned knowledge. Additionally, FAMOS does not require readings from other users as negative samples, further reducing the risk of privacy exposure.

4.1 Overview

FAMOS operates in three phases: data collection, training and authentication. The goal of data collection is to collect data in a user-transparent manner. The goal of the training phase is to train a model that can closely clusters the representation vectors of each action. The goal of the authentication phase is to distinguish whether a new action belongs to the device owner.

Data Collection Phase. FAMOS intermittently collects user data and trains the model in the background of users' daily usage, without impacting users' experience. Before data collection, FAMOS first provides a comprehensive notification to users, detailing the purpose of data collection, the specific types of collected data, and how the data will be stored locally. After obtaining user authorization, FAMOS notifies users when data collection starts and ends.

Specifically, each data collection step occurs only within one minute after the user passes real-time video face authentication, which ensures that the collected data originates from the target user. Each collection step will be terminated preventively if the user inputs an incorrect password, locks the screen, switches Alipay to the background, or remains inactive for 20 seconds. FAMOS repeats the collection step until the required amount of data is attained. Based on our empirical evaluation, active users typically spend over 2.4 minutes per Alipay session, execute approximately 26.3 valid actions per minute, and open the app 3.3 times daily. Therefore, collecting 350 action data samples is usually achieved within three days. Note that FAMOS is a user-transparent authentication method, which is supplementary to conventional authentification methods. The three-day data collection time is acceptable in practice since users can still use other authentication methods when FAMOS is not ready. Therefore, FAMOS does not require immediate user authentication upon the first login. Instead, it allows the system to take some time, with low power consumption, to complete data collection and model training. Besides, the data collection only needs to be conducted once before model training. After the model is trained, FAMOS does not need to collect new data in the next several months. Thus, the data collection phase does not impose a substantial burden on users' devices in the long term.

Training Phase. The training phase of FAMOS is also conducted in the background intermittently. The high-level pipeline of the training phase is illustrated in Figure 2. It consists of three modules: the sensor fusion module (Mod①), the contrastive learning module (Mod②), and the federated learning aggregation module (Mod③). The first two modules are combined as a deep neural network model and deployed in the secure area (e.g., TrustZone) of the users' devices. The Mod③ is deployed on a centralized server. All three modules do not require heavy computation and have low power consumption. Therefore, the training phase does not occupy many computational resources and will not introduce observable lag. As the training phase begins, FAMOS will send a notification to users to inform them that the training has started and another notification when the training is completed. This ensures that users are aware that their data is being processed to enhance security measures. In the notification of the training comple-

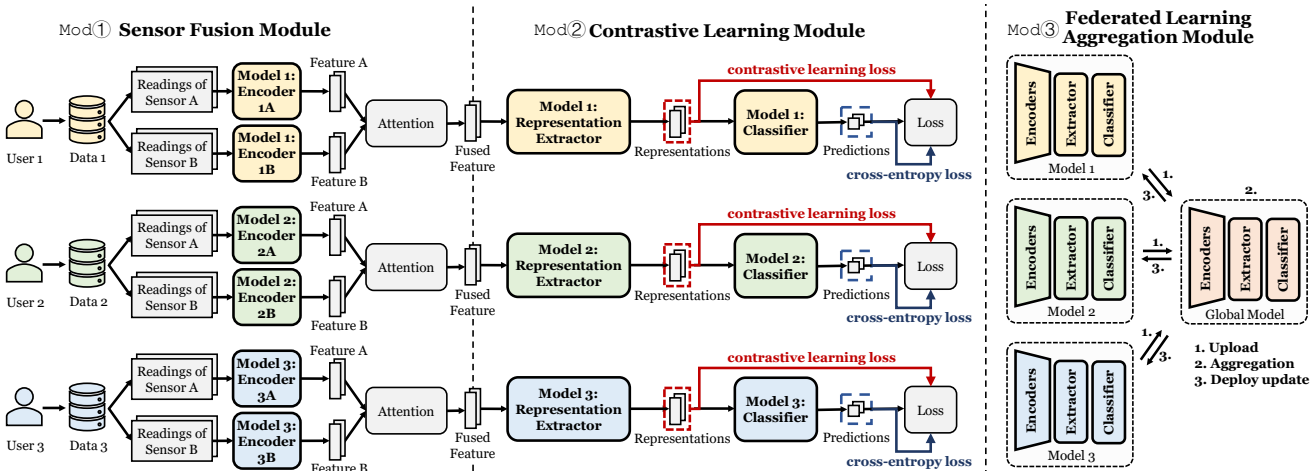


Figure 2: The high-level architecture of FAMOS's training phase.

tion, FAMOS will also inform users that the app could further provide authentication services.

During the training phase, FAMOS launches Mod① and Mod② following an FL paradigm (Mod③). First, it fuses readings from different sensors and clusters representation vectors based on actions. In FAMOS, we support five standard and common user actions within a mobile application, as presented in Table 1. These five actions cover all interactions in Alipay. Second, FAMOS uploads the trained model to the server, which employs Mod③ to aggregate the acquired knowledge from user models. Then, the server sends the aggregated model back to the user. The distributed models are used as the initial model for the next round of training. This process repeats multiple times until the desired performance is achieved.

For the new users that are continuously added to the framework, FAMOS adopts an incremental user selection strategy after the model has been trained [21]. For each new user, FAMOS automatically downloads the latest global model and uses the local data to compute the loss value. The loss values of all participants are uploaded to the server and the server selects the top k users with the highest loss values. This selection scheme steadily enhances the accuracy of local models when new users join the system. It is noteworthy that this process does not impose a substantial burden on computational resources, because new users tend to achieve convergence rapidly within a small amount of time.

Authentication Phase. During the authentication phase, FAMOS authenticates users periodically in the background in a user-transparent way. Specifically, FAMOS leverages the trained model to detect potential adversaries. This phase consists of two stages: an offline pre-processing stage and an online vector-comparison stage. Both stages are conducted on the user's device and do not upload any data to the server. The offline pre-processing stage computes the ground-truth vector for each action and defines a distance threshold.

After the pre-processing stage is complete, FAMOS authenticates the user at one-minute intervals. Specifically, given the sensor readings of a new action, the online vector-comparison stage predicts the action label a and produces the representation vector $\vec{r}_{new}$ based on sensor readings. Then FAMOS compares $\vec{r}_{new}$ with the ground-truth vector of the predicted action $\vec{r}_{gt}^{\,a}$ and computes the distance between them. If the distance exceeds the threshold, FAMOS considers the action pattern abnormal and the device may be compromised. Then, a more stringent authentication method will be invoked for the next payment.

Table 1: Five user actions that FAMOS supports.

Label	Action	Abbreviation
0	Click on the screen	Clk.
1	Swipe on the screen	Swp.
2	Click on the screen and hold	Hld.
3	Two fingers swipe on the screen	Two.
4	Shake the phone	Shk.

4.2 Sensor Fusion Module (Mod①)

The goal of Mod① is to fuse the readings from different sensors. This module takes the raw readings of sensors as input and generates fused feature vectors, in which the noises from background activities are eliminated.

The key challenge in fusing readings from different sensors is how to identify the stable sensor in the noisy environment and mitigate the negative effects of other unstable sensors. In practice, the data stability of different sensors varies under different background activities. For example, the accelerometer can capture more user-related activities when the user is lying compared to when the user is walking. Conversely, for walking, the touch-screen sensor can better capture the user-related activities. However, manually creating rules to identify stable sensors for different activities is impractical due to the diverse range and combinations of user activities and background activities.

To solve this problem, we leverage the attention mecha-

nism [61] to dynamically assign importance weights to different sensors and generate *fused features*. The attention mechanism enables us to automatically select stable sensor readings under different background activities. It is trained to learn the complementary knowledge through different sensor modalities, allowing FAMOS to capture the variance in both data quality and contributions from sensor data. This design takes advantage of multiple sensors to enhance the performance of fused features, thereby enabling FAMOS to achieve **G1**, i.e., robustness to noises from background activities.

Mod① comprises two steps. The first step is projecting the heterogeneous sensor readings to a unified feature space. This step is important because, in practice, the readings of different sensors can exhibit substantial variation. For example, the accelerometer sensor produces readings in three dimensions, while the data from the touch screen sensor only has six dimensions. The second step fuses the unified features from different sensors using the attention mechanism. We will elaborate on each of the steps in detail below.

Step 1: Sensor Feature Projection. Formally, for M sensors, let $\vec{x} = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_M\}$ represent the readings from the sensors, where j is the sensor index. $\vec{x}_j$ represents the reading from the j-th sensor. Hence, this step can be formalized as Equation 1, in which $\mathcal{G}_j$ is the encoder for the j-th sensor. Let Norm$(\cdot)$ represent the normalization operation and Flatten$(\cdot)$ represent the flatten operation. The goal of these two operations is to project the features to the unified dimensions, thus making it easier to fuse. The output $\vec{f}_j$ can be formulated as:

$$\vec{f}_j = \text{Norm}(\text{Flatten}(\mathcal{G}_j(\vec{x}_j))), j \in \{1, ..., M\} \quad (1)$$

In FAMOS, each encoder $\mathcal{G}_j$ consists of a single-layer convolutional neural network. The encoders take each sensor readings as input and output a feature vector of a uniform dimension. We chose the CNN architecture because it is better at extracting information from sparse sensor data and is commonly applied to sensor data [50].

Step 2: Attention-based Fusion. The second step of Mod① is to fuse the feature vectors of different sensors using an attention module. In this way, FAMOS can identify the stable sensors under different background activities. The attention mechanism can be formalized as Equation 2. Different sensors are assigned to different attention weights and each weight w_j represents the stability and importance of the j-th sensor. The output $\vec{f}'$ is the fused vector:

$$\vec{f}' = \sum_{j=1}^{M} w_j \vec{f}_j \quad (2)$$

To obtain w_j, we design the attention mechanism following prior work [50] that can fuse multi-modal heterogeneous data [45]. More specifically, we first compute the hidden representation $\vec{\mu}_j$ of $\vec{f}_j$ through a one-layer MLP (shown in Equation 3). Besides, we normalize the attention weights using a softmax function to control the magnitude of the importance weights (shown in Equation 4).

$$\vec{\mu}_j = \tanh(W \cdot \vec{f}_j + b), j \in \{1, ..., M\} \quad (3)$$

$$w_j = \frac{exp(\vec{\mu}_j \cdot \vec{f}_j)}{\sum_{j=1}^{M} exp(\vec{\mu}_j \cdot \vec{f}_j)}, j \in \{1, ..., M\} \quad (4)$$

4.3 Contrastive Learning Module (Mod②)

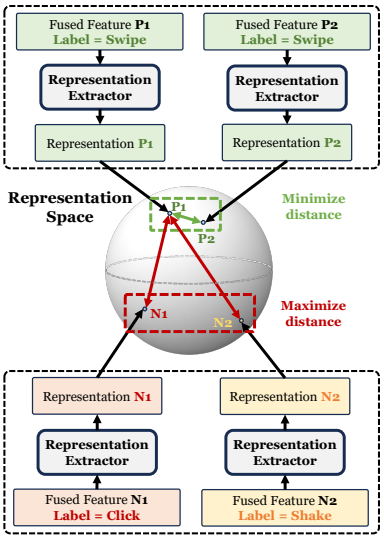


Figure 3: Design of the contrastive learning module.

The goal of Mod② is to cluster the representation vectors of the same action (positive samples) while pushing those of different actions (negative samples) far away. FAMOS can categorize representations into several compact clusters based on different actions. After training, the user's data samples are distributed to the corresponding cluster based on the action representation, while the data samples from other users are excluded by all clusters due to the feature dissimilarity. Therefore, during the authentication stage, we detect the identity of the user by measuring the distance between the representation vector of the current user and the representation vector of the registered user (i.e., ground truth).

Mod② is inspired by the recent advances of contrastive learning [50], which is an effective self-supervised learning approach to learn representations from unlabeled data. The advantage of this module is that we can train FAMOS only with the data from the current user and we do not need the data from other users. In practice, it is difficult to share such data across devices because, according to the regulations such as the GDPR, collecting and sharing such privacy-related data is illegal without the explicit consent of the users.

Module Design. At a high level, this module projects the fused sensor representation vectors to a representation space and minimizes the distances between the representations of the same user. Note that this is challenging due to the diversity

of user actions. A naive approach is to define the loss function as the cosine distance between the representation vectors of the same user. However, this approach converges to a sub-optimal solution in practice since different actions of the same user could be quite different. For example, the representation vectors of clicking are very different from those of swiping.

To address this challenge, we propose to make the training phase *action-aware*, which distinguishes different actions from the same user. The motivation is that the representation vectors of different actions should be far away from each other. The design of our action-aware contrastive learning module is shown in Figure 3. P1 and P2 represent two positive actions, and N1 and N2 are two negative actions. Mod② minimizes the representation distances from the same action (the green arrow in Figure 3) and maximizes the distances from different actions (the red arrows in Figure 3). Thus, we cluster the representation vectors of the same action close to each other and push other actions far away. In addition, the above action-aware design enables Mod② to use different actions from the same user as negative samples to train the model. Thus, Mod② does not need to use other users' data.

Loss Design. To realize the training procedure, we propose a new action-aware loss function $\mathcal{L}_{action}$ which is shown in Equation 5. The loss consists of two parts: $\mathcal{L}_{ce}$ is the classification loss of different actions, and $\mathcal{L}_{cont}$ is the contrastive loss between representation vectors of the same action. λ is a hyperparameter to balance the magnitude between two losses.

$$\mathcal{L}_{action} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{cont} \tag{5}$$

The goal of $\mathcal{L}_{ce}$ is to identify different actions and push the representation vectors of different actions far away from each other. We use a cross-entropy loss to classify different actions. Let B denote a batch of training data, and $s \in B$ is a sample in the batch. $\vec{r}_s$ denotes the representation that is calculated as $\vec{r}_s = \mathcal{M}(\vec{f}'_s)$, where $\mathcal{M}$ denotes the representation extractor. Let $\mathcal{C}$, y_s, and cross_entropy$(\cdot)$ denote the classifier, the label of s, and the cross-entropy function, respectively. We formulate $\mathcal{L}_{ce}$ as:

$$\mathcal{L}_{ce} = \sum_{s \in B} \text{cross\_entropy}(y_s, \mathcal{C}(\vec{r}_s)). \tag{6}$$

By minimizing the cross-entropy loss, the model can automatically push the feature vectors of samples from different classes far away from each other [7].

The goal of $\mathcal{L}_{cont}$ is to minimize the distances of representation vectors of the same action from the same user by pushing these vectors close to each other in the space. The formulation of $\mathcal{L}_{cont}$ is shown in Equation 7. $PB(s)$ is the set of positive samples of s in the batch B, respectively. We define a sample as positive to s if it is from the same action as s. Otherwise, the sample is negative.

$\mathcal{L}_{cont}$ consists of two components. For the action s, $\cos(\vec{r}_s, \vec{r}_p)$ is the cosine distance between its representation vectors $\vec{r}_s$ and the representation vector of a positive sample p. $\cos(\vec{r}_s, \vec{r}_q)$ is the cosine distance between $\vec{r}_s$ and the representation of a negative sample q in the same batch. We then minimize the ratio between $\cos(\vec{r}_s, \vec{r}_p)$ and $\cos(\vec{r}_s, \vec{r}_q)$ to cluster the representation vectors of the same action:

$$\mathcal{L}_{cont} = -\sum_{s \in B} \sum_{p \in PB(s)} \log \frac{\cos(\vec{r}_s, \vec{r}_p)}{\sum_{q \in B - \{s\}} \cos(\vec{r}_s, \vec{r}_q)} \tag{7}$$

Note that we do not necessitate data samples from other users during training. This is based on findings from previous research [59], which suggests that in the representation space, the distance between samples from attackers and victims is sufficiently large. This distance arises because contrastive learning effectively clusters the representations of the victim's samples closely.

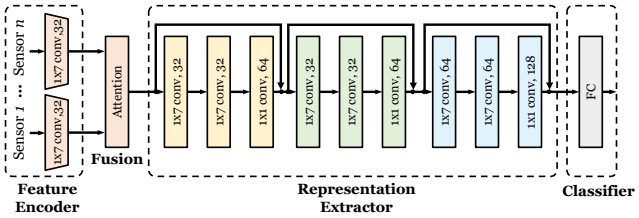


Figure 4: Residual DNN model architecture.

Model Architecture. Another challenge of Mod② is that the encoders should be small enough to fit in the TrustZone (**G2**). The size of TrustZone is typically limited on a mobile device, thus we must carefully design the model architecture so that Mod② can fit in the TrustZone and still has enough capacity to capture the features of sensor readings. A straightforward design will either introduce too many parameters (e.g., MobileNet [29]) or too little capacity which harms accuracy (e.g., LeNet-5 [35]). To balance the model size and performance, we choose a modified residual DNN architecture [28] because its skip connection design can achieve high accuracy with fewer parameters than other candidates [2].

The model architecture is shown in Figure 4, where we design encoders ($\mathcal{G}_j$) for different types of sensors. Specifically, we leverage a convolution layer as the architecture of the encoder that unifies the readings of sensors with different dimensions into features of the same size. Besides the sensor-specific encoders, we have two universal components: the representation extractor ($\mathcal{M}$) and the classifier ($\mathcal{C}$). The former component consists of three residual blocks (represented in yellow, green, and blue). Each block includes three convolution layers, in which the kernel size of the first two layers is 1×7 and the kernel size of the last layer is 1×1. The input and output of each block are connected by a skip connection to facilitate gradient propagation [28]. The classifier is composed of one FC layer. The representation extractor is used to compute $\mathcal{L}_{cont}$ and the classifier component computes $\mathcal{L}_{ce}$. The model size is only 1.81 MB with 1.17M parameters, which is small enough to fit in mobile secure world such as TrustZone.

4.4 FL Aggregation Module (Mod③)

FAMOS aims to achieve a high model performance while avoiding uploading the raw user data to the central server. With FL, users only need to upload locally-trained models. Without FL, FAMOS either suffers from degraded performance (when trained only with local data) or faces the risk of breaching user privacy (when user data is collected to a central server). For the former case, FAMOS experiences lower accuracy due to insufficient data. For the latter case, uploading raw user data to the server may violate regulations.

Using model aggregation in FL to improve the model performance in FL is a recognized practice in the AI community [39, 47, 73]. There are two reasons. First, aggregation enables the global model to learn patterns and features from different data distributions of diverse devices, thereby enhancing the robustness and generalization of the global model. Second, aggregation allows different devices to collaboratively contribute to the improvement of the global model.

In FAMOS, Mod③ aims to aggregate the general knowledge from the user-side model while leaving the user-specific knowledge on the device. This module takes the locally trained user model as input and outputs an aggregated model. The aggregated model is sent back to the user device to update the local model. The aggregation process is conducted on the server of the app owner. Because FAMOS only uploads the trained model, it does not leak any user data (**G3**).

Challenge. The key challenge of this module is that each user model contains not only general (i.e., user-agnostic) knowledge but also user-specific information. The general knowledge can improve FAMOS's ability to extract representation from sensor readings. Differently, FAMOS uses user-specific information to distinguish the user action from the potential adversary. Thus, an ideal aggregation strategy should only aggregate the general knowledge while not aggregating user-specific knowledge, to better accommodate diverse data distributions of different devices. However, if we directly use conventional FL techniques (e.g., FedAvg [47]), we would aggregate both general and user-specific information, which is undesirable.

Solution. To address this challenge, we aggregate all layers across the whole model except the `BatchNormalization` layer. Prior work has proved that the `BatchNormalization` layers contain most of the user-specific information, and thus leaving them on the device can prevent the aggregation of user-specific information [39]. Formally, let $\mathcal{M}_i$ be the representation extractor of user i and $\mathcal{M}_{global}$ be the aggregated extractor. Let ℓ be the layer index except the `BatchNormalization` layers. The aggregation rule is as follows:

$$\mathcal{M}_{global}^{(\ell)} = \frac{1}{\sum_{i=1}^{N} n_i} \sum_{i=1}^{N} n_i \mathcal{M}_i^{(\ell)}, \tag{8}$$

where n_i is the amount of training data of user i. Similarly, this aggregation strategy is applied to both feature encoders $\{\mathcal{G}_i\}$ and the classifier $\mathcal{C}_i$. The local training and server aggregation process repeat multiple times until the user models achieve desirable performance in the classification task. Therefore, Mod③ can enhance the representation extractor, accelerate convergence, and improve effectiveness.

4.5 Authentication

After the training, FAMOS uses the trained model to authenticate users on the devices. The authentication phase consists of two stages: the offline pre-processing stage and the online vector comparison stage. The first stage computes a *ground-truth vector* for each action and defines a distance threshold. The second stage continuously takes sensor readings as input, predicts the action label, and computes a *representation vector* for the input readings. Then FAMOS compares the representation vector with the ground-truth vector of the predicted action to determine whether the user action is abnormal.

Offline Pre-Processing. Due to the contrastive learning scheme, representations of the actions performed by the same user tend to be spatially proximate in the representation space [20]. Thus we construct ground-truth vectors $\vec{r}_{gt}^{\,a}$ for each action a by computing the center of correctly classified samples in the training data. Let $\mathcal{A}_i^a$ be the set of correctly classified samples of action a for the i-th user, the ground-truth vector is computed as:

$$\vec{r}_{gt}^{\,a} = \frac{1}{|\mathcal{A}_i^a|} \sum_{\vec{r}_i^{\,a} \in \mathcal{A}_i^a} \vec{r}_i^{\,a}. \tag{9}$$

The ground-truth vector acts as a reference point for the user's own data. After the vectors are computed, they are stored in the TrustZone of the user's device to prevent privacy leakage and tampering.

In the offline stage, FAMOS also computes the threshold t from the validation set to distinguish the abnormal user actions. Specifically, FAMOS chooses the threshold that achieves the best trade-off between the False Acceptance Rate and False Rejection Rate are equal in the validation set. A more detailed illustration of how to choose the threshold is included in Section 5.4.

Online Vector Comparison. This stage begins after the user logs into the payment application and periodically captures the interaction actions of the user at a certain interval. For each action, FAMOS uses the classifier $\mathcal{C}$ to predict the action label a. Then, FAMOS produces the representation vector $\vec{r}_{new}$ of the action and computes its cosine distance between the ground-truth vector $\vec{r}_{gt}^{\,a}$: $dis = 1 - \cos(\vec{r}_{new}, \vec{r}_{gt}^{\,a})$. At last, FAMOS uses the threshold t to determine whether the action passes ($dis < t$) the authentication or fails ($dis > t$). Note that this online stage is performed on the user's device and the data is never uploaded to the server. Thus the authentication phase does not leak the user's privacy.

5 Evaluation

In this section, we perform comprehensive experiments to evaluate the effectiveness of FAMOS. Our experiments aim to answer the following research questions:

- **RQ1**: How is FAMOS's performance compared with existing solutions?
- **RQ2**: How well can FAMOS mitigate background noises effectively?
- **RQ3**: Can contrastive learning cluster actions from different users?
- **RQ4**: Can federated learning improve the performance and facilitate training process?
- **RQ5**: How is the on-device efficiency of FAMOS?

5.1 Implementation

We use PyTorch 1.8 to implement FAMOS. We train the models (feature encoders, representation extractor, and classifier) in a decentralized manner. The encoders, representation extractor, and classifier are updated simultaneously. We trained the model using the contrastive and cross-entropy loss, which are provided by the library `pytorch.nn`.

We use the Grid Search [41] to find the optimal hyperparameter. Specifically, we set the optimizer to Adam, the learning rate to 1e-2, the batch size to 32, the feature length to 128, and the local training epoch to 5.

5.2 Ethical Disclaimer

We obtain the IRB approval from the ethics review committee in Alipay before we install Alipay-b and collect the data. Besides, all the volunteers are aware that the sensor readings are collected for research purposes. The data is not used for any other commercial purposes nor shared with any third party. We properly stored the data in an encrypted database.

5.3 Evaluation Protocol

To ensure the generalizability and soundness of our evaluation, we measure the performance of FAMOS with both real-world user data and carefully simulated in-lab data. We first use the real-world data to evaluate the overall performance of FAMOS in realistic settings. This experiment ensures the external validity of our evaluation. Then we take a well-controlled in-lab experiment to evaluate the performance of the internal components of FAMOS. This experiment ensures the internal validity of our experiments.

Our testbeds include a server and four smartphones from different brands. The server is used to evaluate the authentication accuracy, while the smartphone is used to measure the training and inference cost. The server is equipped with Ubuntu 20.04.6, 128 GB memory, an Intel Xeon CPU with 48 cores, and 4 NVIDIA GeForce GTX 2080Ti GPUs. We use four latest smartphones of different brands to measure the on-device performance and demonstrate the generalizability of FAMOS to different smartphone devices. The smartphones are Huawei Mate X3, Xiaomi 13 Pro, VIVO X100, and Honor Magic 6. The smartphones are equipped with TrustZone in which we can deploy our model, as discussed in the threat model (Section 3) and prior work [23] to prevent the model from being tampered with by attackers.

We use AuthentiSense [23] and KedyAuth [30] as the baselines because they are the latest state-of-the-art user-transparent authentication approach that was published on the top-tier conference. We rigorously implemented the baselines following the description in the paper as we did not find the source code of them. We use the same dataset as FAMOS to train and evaluate the baselines. Note that they do not fuse readings of different sensors. Instead, they directly sums the readings of the sensors as the input feature. Thus, existing baselines are not robust against background noises, as we will show in our evaluation.

We report both the averaged values and the specific values of each user model. To comprehensively evaluate FAMOS, we use six different metrics in the evaluation: False Acceptance Rate (FAR), False Rejection Rate (FRR), Equal Error Rate (EER), True Positive Rate (TPR), F1-Score, Area Under ROC Curve (AUC). These metrics are consistent with prior work [23].

5.3.1 Data Collection

To ensure that our experiments can faithfully simulate real-world user scenarios, we use a commercial mobile payment app, Alipay as the target app to evaluate FAMOS. We implemented Alipay-b by adding extra code for recording sensor readings in the callback functions for different user actions in the original version of Alipay. When the user takes an action (e.g., click or swiping), the corresponding callback will be invoked. Then, the added code will start a background thread that records the readings of sensors for *two* seconds and log the readings of the sensors along with the label of action. We call a tuple action-data tuple <*action, sensor_data*> as a data sample or sample for simplicity in this paper. Alipay-b records the five types of actions in Table 1.

sensor_data={Screen, IMU} contains the readings of touch screen sensors and IMU sensors. The touch screen sensor data *Screen* is a series of data points sampled as the rate of *twenty milliseconds*. Each point is a six-dimension vector that contains the following values:

- `start_x`: the X-coordinate of action's starting position.
- `start_y`: the Y-coordinate of action's starting position.
- `current_x`: the X-coordinate of current position.
- `current_y`: the Y-coordinate of current position.
- `duration`: the tap duration.
- `pressure`: the current tap pressure.

IMU={accelerometer, gyroscope, magnetometer} records the readings of three types of IMU sensors. Specifically, accelerometer measures the vibration or acceleration of the phone. Gyroscope measures the angular velocity of the phone

around the real-world coordinates. Magnetometer measures the Earth's magnetic field. These three IMU sensors all measure a vector in the three-dimensional real world. The data of accelerometer, gyroscope, and magnetometer are a series of data points with three-dimensional coordinates: `x`, `y`, and `z` for index at the X-, Y-, and Z- real-world coordination. The value of the coordinates is sampled every *ten milliseconds*.

5.3.2 Dataset Construction

We developed two datasets for our study: a real-world dataset for assessing FAMOS's overall performance in real-world environment and an in-lab dataset, which allows us to control background activities for a more precise evaluation of FAMOS's effectiveness in mitigating their impact.

Real-World Dataset. Our real-world dataset contains data from 70 realistic Alipay users. We delegated the developing team of Alipay to identify the candidates in our experiment and publish Alipay-b to users. We asked the experiment candidates to replace the original Alipay with Alipay-b and use the beta version in their daily lives. Note that we do not impose any restrictions on the background activities of the user actions. Hence, users are free to interact with phones in a natural manner under various background activities such as walking and lying. This makes the collected data more realistic and representative of the daily use cases.

We split the 70 users into two groups: 20 victims and 50 attackers. For each of the victims, we collect 500 data samples, 100 samples for each action. We then divided these samples by the ratio of 7:1:2 to form the training (350 samples), validation (50 samples), and test sets (100 samples), respectively. For the attackers, we collect 150 data samples (30 for each action) in total and divide the samples by 1:2 to form the validation (50 samples) and test sets (100 samples). For each victim user, the training set is used to train the authentication model. The validation and test sets are used to evaluate the model. For each victim user, the validation and test sets contain an equal number of victim samples (i.e., positive samples) and attack samples (i.e., negative samples). Note that the attacker's data is only used for evaluation the model in the validate and test sets. The attacker's data is not used for training FAMOS. This process simulates a typical attack scenario of user-transparent authentication as described in Section 3.

In-Lab Dataset. To build the in-lab dataset, we hired 24 volunteers from Ant Group, the mother company of Alipay, to collect data under specific background noises. The dataset construction process is approved by the IRB. Among the 24 volunteers, four act as victims and 20 as attackers. The 24 volunteers are divided into four groups, each group consists of one victim and five attackers. Each group uses one device to collect data from the victim and the attackers. The design of this volunteer setting is to simulate a realistic attack scenario that the victim and attackers use the same device. For background activities, we ask the volunteer to perform five activities: walking, lying, sitting, jogging, and climbing.

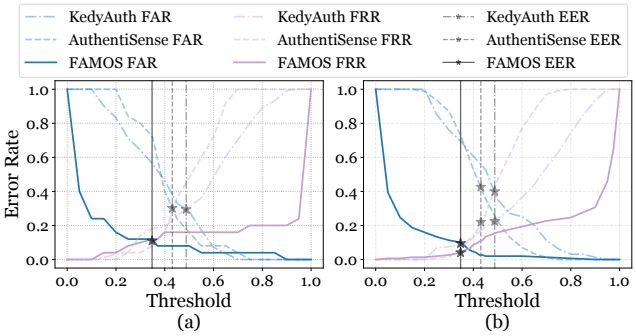


Figure 5: (a) FAR, FRR, and EER on the validation set. (b) FAR, FRR, and EER on the test set. EER is computed from the optimal threshold of the validation set.

For each victim users, we collected 1250 samples in total. Among 1250 samples, each action has 250 samples, 50 samples for each of the five background activities. All the samples of victims are divided into training, validation, and test sets by the ratio of 7:1:2. Thus each victim has 875 samples in the training set, 125 samples in the validation set, and 250 samples in the test set. For each of the attackers, we collected 75 samples. Each action has 15 samples, with three for each of the five background activity. These 75 samples are divided into validation and test sets by the ratio of 1:2. Thus each attacker has 25 samples in the validation set and 50 samples in the test set. For each victim user with five attackers, there are 125 attack samples in the validation set and 250 attack samples in the test set. This design ensures that the validation and test sets contain an equal number of victim samples (i.e., positive samples) and attack samples (i.e., negative samples).

5.4 RQ 1: Overall Effectiveness

In this section, we assess FAMOS's performance using a real-world dataset, comparing it with AuthentiSense across various metrics. Firstly, we determine the optimal threshold by computing the EER, FAR, and FRR on the validation set, which is then applied to the test set to ascertain the FAR and FRR. Second, we plot the ROC curve and compute the AUC score on the test set. Third, we evaluate F1-Score to report a fine-grained comparison between FAMOS and AuthentiSense.

FAR, FRR, and EER. The result of the first part is shown in Figure 5.The x-axis represents different thresholds and the y-axis represents the error rate. The blue line represents the FAR and the purple line represents the FRR. We mark the optimal threshold (chosen by the validation set) with a vertical dashed line. In each figure, FAMOS is represented by the solid lines and AuthentiSense is represented by the blurred lines. Figure 5 (a) shows the results on the validate set and Figure 5 (b) shows the results of the test set.

We use the validation set to find the optimal threshold that FAR equals to FRR (the intersection between the blue curve and the purple curve). The optimal threshold represents the setting that keeps both the FAR and FRR low. In Fig-

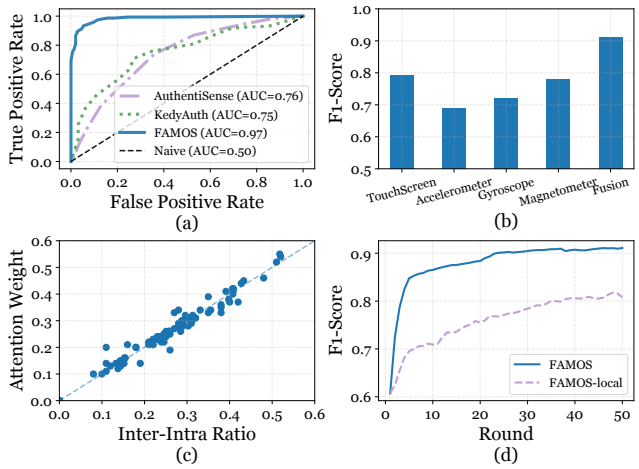


Figure 6: (a) Comparison of ROC curves. (b) F1-Score of different sensors and the sensor fusion. (c) The relationship of attention weight and the Inter-Intra Ratio. (d) Convergence curves of F1-Score of FAMOS and FAMOS-local.

ure 5 (a), we mark the optimal point with a black star (★). As the figure shows, the optimal threshold of FAMOS is 0.35, where the FAR=FRR=0.11. For AuthentiSense, the optimal threshold is 0.43, where FAR=FRR=0.30. For KedyAuth, the optimal threshold is 0.48, where FAR=FARR=0.29. As EER represents the optimal threshold to balance declined benign attempts (false positive) and accepted attack attempts (true negative), a lower EER of FAMOS represents a higher effectiveness for authentication.

For the results on the test set in Figure 5 (b), we report the EER based on the optimal threshold from the validation set. For FAMOS on the optimal threshold (0.35), FAR is 0.09 and FRR is 0.04. For AuthentiSense on the optimal threshold (0.43), FAR is 0.43 and FRR is 0.22. For KedyAuth on the optimal threshold (0.48), FAR is 0.40 and FRR is 0.23. In the authentication service, FAR is more important than FRR, because a false rejection only introduces an additional authentication step, while a false acceptance leads to malicious access and may cause severe security accidents. The FAR of FAMOS is over 4.44× lower than the other two baselines, which means FAMOS provides a more reliable authentication service in real-world scenarios. Meanwhile, the FRR of FAMOS is over 5.51× lower than baselines, which means it can effectively reduce the probability of identifying the user as an attacker, thus enhancing the user experience.

ROC and AUC. Figure 6 (a) shows the ROC score to detect malicious access on the test dataset. Except for FAMOS and two baselines, we also plot a *Naive* baseline to represent the result of random guess (AUC is 0.50). As shown in the figure, the AUC of FAMOS, AuthentiSense, and KedyAuth are 0.97, 0.76, and 0.75, respectively. The AUC of FAMOS approaches the optimal value of 1.00 and is 27.63% higher than baselines, which means FAMOS can detect malicious access more precisely.

F1-Score. For FAMOS, the average F1-Score of FAMOS is 0.91, which demonstrates that FAMOS could provide a more reliable authentication service that can effectively distinguish victim users from malicious attackers. On the contrary, the performance of baselines is much lower than FAMOS. The F1-Score of AuthentiSense and KedyAuth are only 0.62 and 0.64, which are 31.87% and 29.68% lower than FAMOS.

Overall Effectiveness. As a summary, FAMOS can achieve over 4.44× lower FAR, 5.51× lower FRR, 27.63% higher AUC, and 42.19% higher F1-Score than the two state-of-the-art baselines. FAMOS is more effective and reliable than existing approaches in the user authentication service.

5.5 RQ 2: Mitigating Background Activities

In this section, we evaluate whether combining different sensors can effectively mitigate background noises. To this end, we first evaluate the improvement of using fused sensor data of the real-world dataset. Then, we evaluate how our attention identifies stable sensors and eliminates background noises.

Improvement on Accuracy. To evaluate how much can fusing multiple sensors improve the overall accuracy, we removed the sensor fusion module of FAMOS and evaluated its accuracy on real-world data with different sensors. We report the results in Figure 6 (b). Each bar represents the F1-score of using only one sensor. For example, TouchScreen means we only use the touch screen data to achieve user authentication. The bar of Fusion represents the result of FAMOS.

As shown in Figure 6 (b), fusing different sensors indeed improves the overall accuracy. The F1-Score for using touch screen, accelerometer, gyroscope, and magnetometer only is 0.79, 0.69, 0.72, and 0.78, respectively. These are all at least 0.12 lower than the F1-Score of FAMOS, which is 0.91.

Insight Verification. Our insight to mitigate the background noises is that different sensors have different stability under different background activities. Thus, by fusing different sensors, they can mitigate the background noise for each other. To evaluate the validity of this insight, we evaluate how the attention can improve the quality of the feature vectors learned by `Mod`① (See Figure 4) with the In-lab dataset.

Specifically, we evaluate the quality of the features vectors learned by `Mod`① and the attention with the stability score, following previous studies [12, 53]. The stability is defined as the ratio between the action similarities between *different* users (Inter-User Similarity, represented as "Inter" in Table 2). and the action similarities from the *same* user(Intra-User Similarity, represented as "Intra" in Table 2). We denote the action similarities of a set of samples as the average of Euclidean distances of the feature vectors of the samples [48]. The feature vectors are computed through our feature encoder module. A higher stability means the samples from the different users are well clustered: samples from the same users are close while samples from different users are far from each other [70].

We report the results in Table 2, the first four columns represent the Intra-User Similarity, Inter-User Similarity, and

Table 2: Stability (Inter-Intra Ratio) of data samples of each sensor and fused sensor. The Touch Screen sensor is represented as "-" as it produces no readings during the "Shake" action. “BG.” denotes the background activity, “Act.” denotes the action.

BG.	Act.	Touch Screen			Accelerometer			Gyroscope			Magnetometer			Fusion		
		Intra↓	Inter↑	Ratio↑	Intra↓	Inter↑	Ratio↑	Intra↓	Inter↑	Ratio↑	Intra↓	Inter↑	Ratio↑	Intra↓	Inter↑	Ratio↑
Walking	Clk.	5.24	16.48	3.15	40.34	38.56	0.96	37.25	39.63	1.06	14.21	31.71	2.23	2.05	8.39	4.09
	Swp.	4.15	15.12	3.64	34.15	37.28	1.09	36.94	38.64	1.05	12.08	31.22	2.58	1.98	8.24	4.16
	Hld.	4.43	15.38	3.47	35.22	38.33	1.09	33.68	38.29	1.14	11.26	29.36	2.61	1.84	8.08	4.39
	Two.	4.36	15.08	3.46	36.13	36.71	1.02	32.49	37.28	1.15	10.07	28.54	2.83	1.88	8.30	4.41
	Shk.	-	-	-	37.82	38.14	1.01	36.71	39.26	1.07	13.09	31.88	2.44	2.03	8.11	4.00
Lying	Clk.	12.19	16.83	1.38	4.12	8.39	2.04	4.85	8.50	1.75	9.01	18.97	2.11	1.38	6.98	5.06
	Swp.	10.58	18.26	1.73	3.86	8.46	2.19	4.36	9.26	2.12	8.87	19.30	2.18	1.31	6.53	4.98
	Hld.	12.07	17.04	1.41	4.01	8.83	2.20	4.74	9.15	1.93	9.32	18.26	1.96	1.43	7.02	4.91
	Two.	11.21	17.27	1.54	3.86	8.45	2.19	4.26	8.76	2.06	8.69	20.28	2.33	1.36	6.72	4.94
	Shk.	-	-	-	4.02	8.37	2.08	4.31	8.95	2.08	9.14	19.73	2.16	1.42	6.80	4.79
Sitting	Clk.	4.49	10.29	2.29	3.52	8.11	2.30	4.91	11.38	2.32	9.18	20.76	2.26	1.28	6.41	5.01
	Swp.	4.52	11.03	2.44	3.97	9.52	2.40	5.02	12.18	2.43	9.41	21.48	2.28	1.32	6.55	4.96
	Hld.	4.44	10.68	2.41	3.69	9.36	2.54	4.72	11.58	2.45	8.59	22.12	2.58	1.49	6.64	4.46
	Two.	3.94	9.71	2.46	4.11	10.14	2.47	4.75	10.54	2.22	9.41	22.43	2.39	1.43	6.82	4.77
	Shk.	-	-	-	4.08	9.96	2.44	5.31	11.43	2.15	9.51	21.62	2.27	1.61	6.88	4.30
Jogging	Clk.	5.66	17.82	3.15	40.13	39.25	0.98	37.33	42.44	1.14	14.62	30.66	2.10	1.88	7.75	4.12
	Swp.	6.29	18.7	2.97	42.52	42.68	1.00	34.11	38.08	1.12	13.02	28.79	2.21	1.85	7.92	4.28
	Hld.	6.31	20.62	3.27	40.31	42.99	1.07	39.10	40.85	1.04	12.14	26.30	2.17	2.14	8.63	4.03
	Two.	6.76	19.10	2.83	32.64	33.93	1.04	33.41	35.16	1.05	11.48	26.50	2.31	1.84	7.93	4.31
	Shk.	-	-	-	40.80	41.92	1.03	36.08	35.83	0.99	13.66	29.69	2.17	1.99	8.02	4.03
Climbing	Clk.	6.37	19.25	3.02	32.75	37.30	1.14	33.23	38.50	1.16	14.82	33.64	2.27	2.14	9.55	4.46
	Swp.	6.02	18.46	3.07	36.16	39.77	1.10	30.29	31.23	1.03	14.69	33.87	2.31	2.06	8.57	4.16
	Hld.	6.53	20.15	3.09	29.26	33.72	1.15	32.05	35.03	1.09	15.13	33.90	2.24	2.08	8.31	4.00
	Two.	6.50	19.81	3.05	34.48	36.86	1.07	36.51	39.17	1.07	11.23	25.73	2.29	2.01	8.15	4.05
	Shk.	-	-	-	28.51	31.53	1.11	35.34	37.37	1.06	15.13	34.25	2.26	1.94	8.18	4.22

Table 3: Attention weight in various background activities. “BG.” denotes the background activity, “Act.” denotes the action. “Touch.” denotes the touch screen sensor.

BG.	Act.	Attention Weight			
		Touch.	Accelerometer	Gyroscope	Magnetometer
Walking	Clk.	0.36	0.13	0.20	0.31
	Swp.	0.37	0.10	0.20	0.33
	Hld.	0.35	0.11	0.21	0.33
	Two.	0.40	0.10	0.20	0.30
	Shk.	0.00	0.21	0.25	0.54
Lying	Clk.	0.14	0.34	0.19	0.33
	Swp.	0.21	0.28	0.22	0.29
	Hld.	0.20	0.28	0.23	0.29
	Two.	0.23	0.27	0.21	0.29
	Shk.	0.00	0.37	0.24	0.39
Sitting	Clk.	0.25	0.25	0.25	0.25
	Swp.	0.26	0.25	0.25	0.24
	Hld.	0.24	0.25	0.25	0.26
	Two.	0.26	0.26	0.24	0.24
	Shk.	0.00	0.34	0.32	0.34
Jogging	Clk.	0.44	0.14	0.16	0.26
	Swp.	0.42	0.14	0.16	0.28
	Hld.	0.45	0.13	0.12	0.30
	Two.	0.41	0.14	0.14	0.31
	Shk.	0.00	0.22	0.23	0.55
Climbing	Clk.	0.38	0.15	0.16	0.31
	Swp.	0.42	0.14	0.14	0.30
	Hld.	0.41	0.14	0.13	0.32
	Two.	0.40	0.15	0.14	0.31
	Shk.	0.00	0.26	0.22	0.52

Stability (Inter-Intra Ratio) for the four sensors we used in this paper (touch screen, accelerometer, gyroscopes, and magnetometer), respectively. The fusion column represents the results for FAMOS. From the results in Table 2, we can first observe that background noise can significantly affect the stability of sensor readings. For example, when walking, accelerometer and gyroscope have higher Intra-User Similarity and lower Inter-Intra Ratio than the touch screen and magnetometer. It means the accelerometers and gyroscopes are not stable under the background of walking. Similarly, we can observe that the touch screen and magnetometer are unstable under the background of lying. Therefore, we can conclude that background noise can significantly affect the stability of sensor readings and bring difficulties to user authentication. *This result confirms our insight that different sensors have different stability under different background activity.*

Second, according to the results in Table 2, we conclude that FAMOS improves the stability by fusing sensors. We can observe that the last column of Table 2 (column “Fusion”) has the highest Inter-Intra Ratio for all actions with both background activities. The Inter-Intra Ratio of fused sensors is 2.24× higher than a single sensor under two background activities (averagely 4.21 in walking, 4.93 in lying, 4.8 in sitting, 4.16 in jogging, and 4.18 in climbing).

Effectiveness of Attention. FAMOS uses the attention to identify stable sensors for authentication. To validate this insight, we report the average attention weight for each sensor under different background activities in Table 3. The sensor with higher stability (higher Inter-Intra Ratio) receives higher weights, while the sensor with lower stability receives lower weights. For example, when the background is walking, the touch screen sensor receives a relatively lower weight (averagely 0.19) as it exhibits weaker stability in this scenario.

Table 4: Compared of the averaged distance of fused features and representation vectors.

Action type	Features	Representations
Victim Samples (Different actions)	0.54	0.83
Victim Samples (Same action)	0.49	0.31
Victim & Attacker Samples (Same action)	0.52	0.50

Conversely, when the background is lying, the touch screen sensor receives a relatively higher weight (averagely 0.36).

Further, we display the correlation between the Inter-Intra Ratio and the attention weights in Figure 6 (c). To improve the clarity of the figure, we normalized the ratios of four sensors into the range [0,1] [2]. We can observe a positive linear correlation between the Inter-Intra Ratio and the attention weights, which are automatically learnt by FAMOS. This demonstrates that the attention module can effectively select stable sensors under different background noises.

5.6 RQ 3: Effective of Contrastive Learning

The goal of our contrastive learning (Mod②) is to cluster samples from the same user while pushing the samples from different users far from each other. To evaluate the effectiveness of contrastive learning, we compare the cosine distance of fused features (fused by the attention mechanism after the feature encoder) and representation vectors (learned by the representation extractor). In Table 4, we compared the distance of victim samples in different actions, the distance of victim samples in the same action, and the distance between victim and attackers samples in the same action.

Table 4 shows that before applying our representation extractor, distinguishing between different users or actions is challenging due to the minimal average distance differences in their features. Specifically, the distance between different actions is 0.54, very close to the distance between users and attackers, and only slightly higher than within-user sample distances. However, employing a contrastive learning-based representation extractor significantly improves differentiation: the distance between different actions' representation vectors jumps to 0.83, a 1.63 times increase that enhances user authentication accuracy by effectively clustering same-user actions closer and distancing those of different users or attackers.

5.7 RQ4: Effectiveness of Federated Learning

Different from previous approaches, our framework utilizes FL paradigm to aggregate the user-agnostic knowledge to facilitate the training process. In this part, we study how much the FL paradigm can improve the performance of FAMOS. As for the baseline, we create a local training version of FAMOS, denoted as FAMOS-local, by only training FAMOS with the private data of each user (i.e., without aggregation). We compare the F1-Score and the convergence speed of FAMOS and FAMOS-local to evaluate the effectiveness of FL, as shown in Figure 6 (d). On average, the F1-Score of FAMOS-local is only 0.80, about 15.1% lower than FAMOS (averaged F1-Score is 0.91). The results demonstrate that applying FL can effectively improve authentication accuracy. All user models are well-trained and get convergence within 26.21 aggregation rounds on average. This aligns with the average cost for modern FL tasks [42, 43]. On contrary, the FAMOS-local spends more than 42.73 rounds to convergence, about 62.98% slower than FAMOS.

[2] The range of the coordinate axis in Figure 6 (c) is set to [0,0.6] because we found that all the values of normalized ratios and attention weights are smaller than 0.06.

5.8 RQ5: On-device Performance

In this section, we study the on-device performance of FAMOS in the TrustZone of four different smartphones (Huawei Mate X3, Xiaomi 13 Pro, VIVO X100, and Honor Magic 6). We use MNN [32] library to convert the model to the `.mnn` format. This format is a highly optimized format for ARM architecture [44]. Then we deploy the converted model in the TrustZone and measure the on-device performance and usability.

Overhead. We measure the overhead of FAMOS in terms of four aspects: memory usage, CPU usage, battery consumption, and running time for both training and inference phases. For the first three types of overhead, we compare the performance of Alipay with FAMOS running in the background and Alipay without FAMOS. We report the increased value when FAMOS is running. For the running time, we measure the end-to-end training time until convergence and the end-to-end authentication inference time of one input sample. We use Android `dumpsys` tool, sample information from the `/proc/stat` file, and invoke the built-in APIs of TrustZone to comprehensively measure the memory usage, CPU usage, and battery utilization. Each experiment is conducted five times and report the average value. We manually checked the variance of the results and found that it is less than 5%.

The results of the overhead measurement results are shown in Table 5. On average, the additional memory consumption of FAMOS is only 8.58 MB for training and 3.95 MB for inference. CPU utilization increased by 23.77% during training and 16.36% during inference. Besides, the increase of battery consumption per hour is 1.32% for training and 0.48% for inference. These results demonstrate that FAMOS has a negligible impact on the Alipay system. Furthermore, FAMOS completes training in averagely 26.9 minutes (i.e., 1618K milliseconds) and requires 135 milliseconds for inference. Such overhead is acceptable for applications using TrustZone [49]. Note that the training stage only needs to perform once and is a one-time cost. According to the findings of prior works, when the inference time is less than one second, it will not harm the user experience [67], and can support online real-time services [69]. Thus, the overhead measurement demonstrate that FAMOS is practical in real-world deployment.

Usability. We hire the 24 volunteers of the in-lab dataset to conduct the study. Volunteers were asked to engage in three independent Alipay usage sessions in a randomized

Table 5: The overhead measurement of four devices.

Device	Memory (MB)		CPU (%)		Battery (%/h)		Time (ms)	
	Train	Infer	Train	Infer	Train	Infer	Train	Infer
Huawei Mate X3	8.42	3.94	23.83	16.21	1.34	0.52	1590K	132
Xiaomi 13 Pro	8.51	3.83	21.28	14.72	1.18	0.45	1542K	109
VIVO X100	8.74	4.11	25.11	17.61	1.22	0.51	1644K	143
Honor Magic 6	8.63	3.90	24.85	16.90	1.33	0.42	1698K	155
Average	8.58	3.95	23.77	16.36	1.32	0.48	1618K	135

order: one session using Alipay without FAMOS, one session with the training phase of FAMOS, and one session with the authentication phase of FAMOS. All the three sessions lasted for 10 minutes. Volunteers were not informed about which session they were currently participating in, and they were asked to carry out their usual activities as much as possible and to cover the five actions listed in Table 1. The goal of this user study is to investigate if there exist any obvious delays, including lags, freezes, or slowdowns during display, keystrokes, and button interactions. After each session, the volunteers complete a brief survey to provide feedback on the presence of delays. The first question in the survey is "Did you perceive any delays?" If the answer is "Yes," we ask further questions as "During which actions did you experience delays with the display/keystrokes/buttons?" "How significant was the impact of the delay?" and "How often did you experience delays?" We also record the response time of interactions in the three sessions to provide quantitative results.

21 out of 24 volunteers (87.5%) reported no noticeable delays. Two volunteers experienced about five instances of acceptable display delays during the training phase. One volunteer reported an experience about 10 instances of minor button press delays during training phase, stating that these were acceptable. For quantitative results, the average response time of interactions without FAMOS is 1242.43 milliseconds with a standard deviation (SD) of 979.22. The average response time of interactions during the training phase is 1502.63 milliseconds (SD is 1117.38), and the average response time of interactions during the authentication phase is 1263.85 milliseconds (SD is 994.84). To rigorously evaluate the user experience, we use Welch's t-test to conduct significance testing on the response times of the training phase and the authentication phase compared to the response times with FAMOS disabled. The null hypothesis is that there is no significant difference between the two sets of response times. The p-values for training and inference are 0.91 and 0.97, respectively. The p-values can not reject the null hypothesis ($p > 0.05$). Overall, these results indicate that users only experienced minor delays during the training phase, with no difference during the authentication phase.

6 Related Work

In mobile payment, user authentication is crucial for downstream services. While several approaches have been suggested, including passwords [55], SMS verification [54], and authenticators [46]. However, they often lack user-friendliness [23, 65] and are susceptible to sophisticated attacks [38, 64].

Researchers have studied various behavior patterns to authenticate users, such as touch gestures [13,24,68,74], motion sensor [3, 22, 27, 37, 40, 63], or correlated multi-modal feature [6, 9, 19]. AuthentiSense [23] and KedyAuth [30] are the latest works that leverage standard built-in sensors for continuous user authentication. They have two key limitations. First, they are not robust to actions under various background activities. Second, they rely on centralized training to learn user's behavioral biometrics, which violate users' privacy.

7 Discussion

Device-dependency implications. The four sensors used by FAMOS are commonly deployed in modern mobile devices and widely utilized in research related to user action patterns [30]. The data format generated by these sensors is generally consistent across different devices. Although there exist some differences in sensor data for different devices, such variation can be mitigated by the training process of FAMOS. Specifically, FAMOS employs local model training to enhance the accuracy and reliability of user authentication. The training process not only takes into account device-related variations but also aligns with the heterogeneity of sensor data in practice. Thus FAMOS is robust and effective across a variety of mobile devices.

Generalization. FAMOS is a general solution that can be integrated into other payment app due to two reasons. First, the four sensors used in FAMOS are commonly deployed in modern mobile devices and widely utilized in the research about user action patterns. Besides, the payment apps (e.g., AliPay and WeChat) have similar authentication requirements and interaction designs [71], allowing for the consistent deployment process of data collection, model training, and authentication.

8 Conclusion

In this paper, we introduce FAMOS to achieve robust privacy-preserving authentication. FAMOS integrates an attention mechanism to identify and utilize stable sensors, an action-aware model for clustering action-specific representations, and a residual DNN for effectively projecting representations. By incorporating the FL framework, FAMOS mitigates privacy concerns. Our comprehensive evaluation demonstrates FAMOS's effectiveness, significantly outperforming the SOTA with a 42.19% enhancement in F1-score and a 27.63% increase in AUC.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable feedback of this paper. Ding Li is the corresponding author. Yifeng Cai and Ziqi Zhang contributed equally to this work. This work was partly supported by the National Science and Technology Major Project of China (2022ZD0119103), the Natural Science Foundation of Shanghai (23ZR1429600), and the CCF-AFSG Research Fund (RF20220006).

References

[1] Suzan Ali, Mounir Elgharabawy, Quentin Duchaussoy, Mohammad Mannan, and Amr Youssef. Parental controls: safer internet solutions or new pitfalls? *IEEE Security & Privacy*, 19(6):36–46, 2021.

[2] Zeyuan Allen-Zhu and Yuanzhi Li. What can resnet learn efficiently, going beyond kernels? *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Sara Amini, Vahid Noroozi, Amit Pande, Satyajit Gupte, Philip S Yu, and Chris Kanich. Deepauth: A framework for continuous user re-authentication in mobile apps. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 2027–2035, 2018.

[4] Simone Aonzo, Alessio Merlo, Giulio Tavella, and Yanick Fratantonio. Phishing attacks on modern android. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1788–1801, 2018.

[5] Rosanna Bellini, Kevin Lee, Megan A Brown, Jeremy Shaffer, Rasika Bhalerao, and Thomas Ristenpart. The {Digital-Safety} risks of financial technologies for survivors of intimate partner violence. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 87–104, 2023.

[6] Cheng Bo, Lan Zhang, Xiang-Yang Li, Qiuyuan Huang, and Yu Wang. Silentsense: silent user identification via touch and movement behavioral biometrics. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 187–190, 2013.

[7] Malik Boudiaf, Jérôme Rony, Imtiaz Masud Ziko, Eric Granger, Marco Pedersoli, Pablo Piantanida, and Ismail Ben Ayed. A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI*, pages 548–564. Springer, 2020.

[8] Andrew Bud. Facing the future: The impact of apple faceid. *Biometric technology today*, 2018(1):5–7, 2018.

[9] Attaullah Buriro, Bruno Crispo, Filippo Delfrari, and Konrad Wrona. Hold and sign: A novel behavioral biometrics for smartphone user authentication. In *2016 IEEE security and privacy workshops (SPW)*, pages 276–285. IEEE, 2016.

[10] Daniel Buschek, Benjamin Bisinger, and Florian Alt. Researchime: A mobile keyboard application for studying free typing behaviour in the wild. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2018.

[11] Michele Campobasso and Luca Allodi. Impersonation-as-a-service: Characterizing the emerging criminal infrastructure for user impersonation at scale. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1665–1680, 2020.

[12] Bryan Bo Cao, Abrar Alali, Hansi Liu, Nicholas Meegan, Marco Gruteser, Kristin Dana, Ashwin Ashok, and Shubham Jain. Vifit: Reconstructing vision trajectories from imu and wi-fi fine time measurements. In *Proceedings of the 3rd ACM MobiCom Workshop on Integrated Sensing and Communications Systems*, pages 13–18, 2023.

[13] Huijie Chen, Fan Li, Wan Du, Song Yang, Matthew Conn, and Yu Wang. Listen to your fingers: User authentication based on geometry biometrics of touch gesture. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):1–23, 2020.

[14] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Computing Surveys (CSUR)*, 54(4):1–40, 2021.

[15] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5659–5667, 2017.

[16] Rogier Creemers and Graham Webster. Translation: Personal information protection law of the people's republic of china. *DigiChina Project*, 20, 2021.

[17] L Minh Dang, Kyungbok Min, Hanxiang Wang, Md Jalil Piran, Cheol Hee Lee, and Hyeonjoon Moon. Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition*, 108:107561, 2020.

[18] Anupam Das, Nikita Borisov, and Matthew Caesar. Tracking mobile web users through motion sensors: Attacks and defenses. In *NDSS*, 2016.

[19] Debayan Deb, Arun Ross, Anil K Jain, Kwaku Prakah-Asante, and K Venkatesh Prasad. Actions speak louder than (pass) words: Passive authentication of smartphone* users via deep temporal features. In *2019 international conference on biometrics (ICB)*, pages 1–8. IEEE, 2019.

[20] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[21] Yongheng Deng, Feng Lyu, Ju Ren, Huaqing Wu, Yuezhi Zhou, Yaoxue Zhang, and Xuemin Shen. Auction: Automated and quality-aware client selection framework for efficient federated learning. *IEEE Transactions on Parallel and Distributed Systems*, 33(8):1996–2009, 2021.

[22] Muhammad Ehatisham-ul Haq, Muhammad Awais Azam, Usman Naeem, Yasar Amin, and Jonathan Loo. Continuous authentication of smartphone users based on activity pattern recognition using passive mobile sensing. *Journal of Network and Computer Applications*, 109:24–35, 2018.

[23] Hossein Fereidooni, Jan König, Phillip Rieger, Marco Chilese, Bora Gökbakan, Moritz Finke, Alexandra Dmitrienko, and Ahmad-Reza Sadeghi. Authentisense: A scalable behavioral biometrics authentication scheme using few-shot learning for mobile platforms. *NDSS*, 2023.

[24] Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE transactions on information forensics and security*, 8(1):136–148, 2012.

[25] Mayank Goel, Leah Findlater, and Jacob Wobbrock. Walktype: using accelerometer data to accomodate situational impairments in mobile touch screen text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2687–2696, 2012.

[26] Xueluan Gong, Yanjiao Chen, Wang Yang, Qian Wang, Yuzhe Gu, Huayang Huang, and Chao Shen. Redeem myself: Purifying backdoors in deep learning models using self attention distillation. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 755–772. IEEE Computer Society, 2023.

[27] Sandeep Gupta, Rajesh Kumar, Mouna Kacimi, and Bruno Crispo. Ideauth: A novel behavioral biometric-based implicit deauthentication scheme for smartphones. *Pattern Recognition Letters*, 157:8–15, 2022.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[29] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[30] Jun Ho Huh, Sungsu Kwag, Iljoo Kim, Alexandr Popov, Younghan Park, Geumhwan Cho, Juwon Lee, Hyoungshick Kim, and Choong-Hoon Lee. On the long-term effects of continuous keystroke authentication: Keeping user frustration low through behavior adaptation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7(2):1–32, 2023.

[31] Md Mofijul Islam and Tariq Iqbal. Mumu: Cooperative multitask learning-based guided multimodal fusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1043–1051, 2022.

[32] Xiaotang Jiang, Huan Wang, Yiliu Chen, Ziqi Wu, Lichuan Wang, Bin Zou, Yafeng Yang, Zongyang Cui, Yu Cai, Tianhang Yu, et al. Mnn: A universal and efficient inference engine. *Proceedings of Machine Learning and Systems*, 2:1–13, 2020.

[33] Fatima Khalid and Ammar Masood. Vulnerability analysis of qualcomm secure execution environment (qsee). *Computers & Security*, page 102628, 2022.

[34] Radhesh Krishnan Konoth, Björn Fischer, Wan Fokkink, Elias Athanasopoulos, Kaveh Razavi, and Herbert Bos. Securepay: Strengthening two-factor authentication for arbitrary transactions. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 569–586. IEEE, 2020.

[35] Yann LeCun et al. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, 20(5):14, 2015.

[36] Homin K Lee, Tal Malkin, and Erich Nahum. Cryptographic strength of ssl/tls servers: Current and recent practices. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 83–92, 2007.

[37] Wei-Han Lee and Ruby B Lee. Implicit smartphone user authentication with sensors and contextual machine learning. In *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 297–308. IEEE, 2017.

[38] Zeyu Lei, Yuhong Nan, Yanick Fratantonio, and Antonio Bianchi. On the insecurity of sms one-time password messages against local attackers in modern mobile devices. In *Network and Distributed Systems Security (NDSS) Symposium 2021*, 2021.

[39] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021.

[40] Yantao Li, Hailong Hu, Zhangqian Zhu, and Gang Zhou. Scanet: sensor-based continuous authentication with two-stream convolutional neural networks. *ACM Transactions on Sensor Networks (TOSN)*, 16(3):1–27, 2020.

[41] Petro Liashchynskyi and Pavlo Liashchynskyi. Grid search, random search, genetic algorithm: a big comparison for nas. *arXiv preprint arXiv:1912.06059*, 2019.

[42] Bingyan Liu, Yifeng Cai, Hongzhe Bi, Ziqi Zhang, Ding Li, Yao Guo, and Xiangqun Chen. Beyond fine-tuning: Efficient and effective fed-tuning for mobile/web users. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 2863–2873, 2023.

[43] Bingyan Liu, Yifeng Cai, Ziqi Zhang, Yuanchun Li, Leye Wang, Ding Li, Yao Guo, and Xiangqun Chen. Distfl: Distribution-aware federated learning for mobile scenarios. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 5(4), dec 2022.

[44] Chengfei Lv, Chaoyue Niu, Renjie Gu, Xiaotang Jiang, Zhaode Wang, Bin Liu, Ziqi Wu, Qiulin Yao, Congyu Huang, Panos Huang, Tao Huang, Hui Shu, Jinde Song, Bin Zou, Peng Lan, Guohuan Xu, Fei Wu, Shaojie Tang, Fan Wu, and Guihai Chen. Walle: An End-to-End, General-Purpose, and Large-Scale production system for Device-Cloud collaborative machine learning. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pages 249–265, July 2022.

[45] Haojie Ma, Wenzhong Li, Xiao Zhang, Songcheng Gao, and Sanglu Lu. Attnsense: Multi-level attention mechanism for multimodal human activity recognition. In *IJCAI*, pages 3109–3115, 2019.

[46] Claudio Marforio, Nikolaos Karapanos, Claudio Soriente, Kari Kostiainen, and Srdjan Capkun. Smartphones as practical and secure location verification tokens for payments. In *NDSS*, volume 14, pages 23–26, 2014.

[47] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[48] Mark Michael and Wen-Chun Lin. Experimental study of information measure and inter-intra class distance ratios on feature selection and orderings. *IEEE Transactions on Systems, Man, and Cybernetics*, (2):172–181, 1973.

[49] Fan Mo, Hamed Haddadi, Kleomenis Katevas, Eduard Marin, Diego Perino, and Nicolas Kourtellis. Ppfl: privacy-preserving federated learning with trusted execution environments. In *Proceedings of the 19th annual international conference on mobile systems, applications, and services*, pages 94–108, 2021.

[50] Xiaomin Ouyang, Xian Shuai, Jiayu Zhou, Ivy Wang Shi, Zhiyuan Xie, Guoliang Xing, and Jianwei Huang. Cosmo: Contrastive fusion learning with small data for multimodal human activity recognition. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, MobiCom '22, page 324–337, 2022.

[51] Liangying Peng, Ling Chen, Zhenan Ye, and Yi Zhang. Aroma: A deep multi-task learning based simple and complex human activity recognition method using wearable sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(2):1–16, 2018.

[52] Sandro Pinto and Nuno Santos. Demystifying arm trustzone: A comprehensive survey. *ACM computing surveys*, 51(6):1–36, 2019.

[53] Bozhao Qi and Suman Banerjee. Goniosense: a wearable-based range of motion sensing and measurement system for body joints: poster. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 441–442, 2016.

[54] Bradley Reaves, Nolen Scaife, Dave Tian, Logan Blue, Patrick Traynor, and Kevin RB Butler. Sending out an sms: Characterizing the security of the sms ecosystem with public gateways. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 339–356. IEEE, 2016.

[55] Blake Ross, Collin Jackson, Nick Miyake, Dan Boneh, and John C Mitchell. Stronger password authentication using browser extensions. In *USENIX Security Symposium*, volume 17, page 32, 2005.

[56] Lizhi Sun, Shuocheng Wang, Hao Wu, Yuhang Gong, Fengyuan Xu, Yunxin Liu, Hao Han, and Sheng Zhong. Leap: Trustzone based developer-friendly tee for intelligent mobile apps. *IEEE Transactions on Mobile Computing*, 2022.

[57] Zehua Sun, Qiuhong Ke, Hossein Rahmani, Mohammed Bennamoun, Gang Wang, and Jun Liu. Human action recognition from various data modalities: A review. *IEEE transactions on pattern analysis and machine intelligence*, 2022.

[58] Kurt Thomas, Frank Li, Ali Zand, Jacob Barrett, Juri Ranieri, Luca Invernizzi, Yarik Markov, Oxana Comanescu, Vijay Eranti, Angelika Moscicki, et al. Data

breaches, phishing, or malware? understanding the risks of stolen credentials. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 1421–1434, 2017.

[59] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839, 2020.

[60] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[62] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[63] Cong Wang, Yanru Xiao, Xing Gao, Li Li, and Jun Wang. A framework for behavioral biometric authentication using deep metric learning on mobile devices. *IEEE Transactions on Mobile Computing*, 22(1):19–36, 2021.

[64] Ding Wang, Ping Wang, Debiao He, and Yuan Tian. Birthday, name and bifacial-security: Understanding passwords of chinese web users. In *USENIX Security Symposium*, pages 1537–1555, 2019.

[65] Daphna Weinshall. Cognitive authentication schemes safe against spyware. In *2006 IEEE Symposium on Security and Privacy (S&P'06)*, pages 295–300. IEEE, 2006.

[66] Leon Y Xiao. People's republic of china legal update: Supreme people's court's guiding opinion on refund requests relating to unauthorized online video gaming transactions paid for by minors. *Gaming law review*, 24(7):476–479, 2020.

[67] Daliang Xu, Mengwei Xu, Qipeng Wang, Shangguang Wang, Yun Ma, Kang Huang, Gang Huang, Xin Jin, and Xuanzhe Liu. Mandheling: Mixed-precision on-device dnn training with dsp offloading. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, pages 214–227, 2022.

[68] Hui Xu, Yangfan Zhou, and Michael R Lyu. Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones. In *Symposium on usable privacy and security, SOUPS*, volume 14, pages 187–198, 2014.

[69] Mu Yuan, Lan Zhang, Fengxiang He, Xueting Tong, and Xiang-Yang Li. Infi: End-to-end learnable input filter for resource-efficient mobile-centric inference. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, pages 228–241, 2022.

[70] Xiaoyuan Zhang, Jianzhong Zhou, Changqin Wang, Chaoshun Li, and Lixiang Song. Multi-class support vector machine optimized by inter-cluster distance and self-adaptive deferential evolution. *Applied Mathematics and Computation*, 218(9):4973–4987, 2012.

[71] Yue Zhang, Yuqing Yang, and Zhiqiang Lin. Don't leak your keys: Understanding, measuring, and exploiting the appsecret leaks in mini-programs. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 2411–2425, 2023.

[72] Ziqi Zhang, Chen Gong, Yifeng Cai, Yuanyuan Yuan, Bingyan Liu, Ding Li, Yao Guo, and Xiangqun Chen. No privacy left outside: On the (in-) security of tee-shielded dnn partition for on-device ml. In *2024 IEEE Symposium on Security and Privacy (SP)*, 2024.

[73] Ziqi Zhang, Yuanchun Li, Bingyan Liu, Yifeng Cai, Ding Li, Yao Guo, and Xiangqun Chen. Fedslice: Protecting federated learning models from malicious participants with model slicing. In *2023 IEEE/ACM 46th International Conference on Software Engineering*. IEEE, 2023.

[74] Xi Zhao, Tao Feng, Weidong Shi, and Ioannis A Kakadiaris. Mobile user authentication using statistical touch dynamics images. *IEEE Transactions on Information Forensics and Security*, 9(11):1780–1789, 2014.

[75] Chaoshun Zuo, Zhiqiang Lin, and Yinqian Zhang. Why does your data leak? uncovering the data leakage in cloud from mobile apps. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1296–1310. IEEE, 2019.