WearRF-CLA: Continuous Location Authentication with Wrist Wearables and UHF RFID

Ang Li
anglee@asu.edu
Arizona State University
Tempe, Arizona, USA

Jiawei Li
jwli@asu.edu
Arizona State University
Tempe, Arizona, USA

Dianqi Han
dqhan@asu.edu
Arizona State University
Tempe, Arizona, USA

Yan Zhang
yanzhangyz@asu.edu
Arizona State University
Tempe, Arizona, USA

Tao Li
tli6@iupui.edu
Indiana University–Purdue University Indianapolis
Indianapolis, Indiana, USA

Yanchao Zhang
yczhang@asu.edu
Arizona State University
Tempe, Arizona, USA

ABSTRACT

Continuous location authentication (**CLA**) seeks to *continuously* and *automatically* verify the physical presence of legitimate users in a protected indoor area. CLA can play an important role in contexts where access to electrical or physical resources must be limited to physically present legitimate users. In this paper, we present WearRF-CLA, a novel CLA scheme built upon increasingly popular wrist wearables and UHF RFID systems. WearRF-CLA explores the observation that human daily routines in a protected indoor area comprise a sequence of human-states (e.g., walking and sitting) that follow predictable state transitions. Each legitimate WearRF-CLA user registers his/her RFID tag and also wrist wearable during system enrollment. After the user enters a protected area, WearRF-CLA continuously collects and processes the gyroscope data of the wrist wearable and the phase data of the RFID tag signals to verify three factors to determine the user's physical presence/absence without explicit user involvement: (1) the tag ID as in a traditional RFID authentication system, (2) the validity of the human-state chain, and (3) the continuous coexistence of the paired wrist wearable and RFID tag with the user. The user passes CLA if and only if all three factors can be validated. Extensive user experiments on commodity smartwatches and UHF RFID devices confirm the very high security and low authentication latency of WearRF-CLA.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**; **Authentication**.

KEYWORDS

Continuous location authentication (CLA), wireless security, RFID, wrist wearables, deep learning

ASIA CCS '22, May 30–June 3, 2022, Nagasaki, Japan.

ACM ISBN 978-1-4503-9140-5/22/05…$15.00
https://doi.org/10.1145/3488932.3517426

ACM Reference Format:
Ang Li, Jiawei Li, Dianqi Han, Yan Zhang, Tao Li, and Yanchao Zhang. 2022. WearRF-CLA: Continuous Location Authentication with Wrist Wearables and UHF RFID. In *Proceedings of the 2022 ACM Asia Conference on Computer and Communications Security (ASIA CCS '22), May 30–June 3, 2022, Nagasaki, Japan.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3488932.3517426

1 INTRODUCTION

Continuous location authentication (**CLA**) seeks to *continuously* and *automatically* verify the physical presence of legitimate users in a protected indoor area. CLA can play an important role in contexts where access to electrical or physical resources must be limited to physically present legitimate users, e.g., in a mission-critical facility for authorized service members processing classified information, in hospitals for clinicians examining private patient medical records, and in companies for employees accessing confidential databases. In addition, CLA can eliminate the inconvenience for authorized users to keep manually providing identity credentials (e.g., inputting a password) if they have to stay for a long time in a protected site. Furthermore, CLA can automatically deauthenticate users once they leave the protected indoor venue.

CLA is most related to the following categories of works.

- **Presence-detection sensors**. This category relies on existing human-presence detection sensors such as passive IR detectors, ambient light sensors, and ultrasonic proximity sensors [4]. However, these sensors can neither verify user identities nor detect stationary users.
- **Distance bounding and location verification**. This category ties the location of a wireless device (called *prover*) with that of its registered owner. Distant bounding techniques like [28] allow a wireless device (called *verifier*) to verify that a prover is within a certain distance, but they cannot distinguish whether the prover is in the protected zone or an adjacent zone within the enforced distance limit. Location-verification schemes such as [33, 41] explore multiple verifiers to jointly estimate the location of a prover and can fulfill CLA, but the requirement for multiple verifiers may not be easily satisfied in practice.
- **Indoor localization**. This category explores wireless signals (e.g., WiFi, Bluetooth, or acoustic signals) and pre-deployed

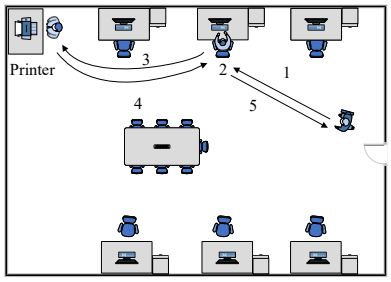


Figure 1: A snippet of user daily routines in a workplace.

devices such as WiFi routers to localize and track user-carried wireless devices in real time [19, 23–25, 40, 42] with up to decimeter-level accuracy. These techniques target benign users/devices, and their resilience to malicious users/devices has not been investigated.

- **Secure localization in sensor networks.** These techniques such as [20, 21] help a sensor node determine its location with the help of a few beacon nodes in the presence of wireless signal attacks. Targeting benign sensor nodes, they do not apply to CLA with possibly malicious users/devices.
- **Biometric authentication**. This category captures and validates a user's activity-related physiological and behavioral biometrics from his/her smartphone's IMU sensor data [30] or surrounding wireless signals [35] to continuously verify his/her physical presence in an indoor area. But these systems can only identify users when they are engaged in particular human activities like walking or sitting. Moreover, they require each user to participate in an intensive model-training process and are less applicable to multi-user scenarios such as in a workplace.

In this paper, we present **WearRF-CLA**, a novel CLA scheme built upon increasingly popular wrist wearables (e.g., smartwatches and fitness trackers) and ultra-high frequency radio frequency identification (UHF RFID) systems. In particular, Pew Research reported that about 21% of U.S. adults regularly use a smartwatch or fitness tracker in 2020, and this percentage keeps growing. In addition, UHF RFID tags have been widely used in many applications such as personnel identification, access control, and personal tracking. For example, the employees in many workplaces (e.g., companies, hospitals, and military bases) are required to wear uniforms or ID badges with embedded RFID tags. *WearRF-CLA is designed only for the contexts where each legitimate user carries both a wrist wearable and an RFID tag embedded into his/her ID badge or uniform.*

WearRF-CLA explores the observation that human daily routines in a protected indoor area comprise a sequence of human-states. In Fig. 1, we use Bob as an exemplary legitimate user to illustrate a snippet of his daily routines in a multi-user workplace. Bob enters the locked workplace after swiping his ID card or inputting the PIN on the door keypad. He then walks to his desk, sits down, and starts working (steps 1&2). During working hours, he may access some confidential databases through his computer by inputting additional passwords. Later, he sends some documents to the printer. So he stands up, walks to the printer, fetches the documents, goes back to his desk, sits down, and continues working (steps 3&4&2). If he decides to leave the workplace, he stands up and exits through the door (step 5). So Bob's activities can be seen as a natural human-state chain: walk→sit down→sit & work→stand up →walk→stand & fetch→walk→sit down→sit and work→stand up→walk.

WearRF-CLA explores the above observation for CLA in a multi-user indoor venue containing a commodity UHF RFID reader. Each legitimate user registers his/her RFID tag and also wrist wearable during system enrollment. Continue with the previous example. After Bob enters a protected indoor area, his RFID tag keeps answering the queries from the RFID reader while his wrist wearable (e.g., smartwatch) continuously records gyroscope data. A backend server retrieves the RFID signals' phase data from the reader and the gyroscope data from the wrist wearable. WearRF-CLA processes the received gyroscope and RFID-phase data to continuously and automatically verifies three factors to determine Bob's physical presence/absence without his explicit involvement: (1) the tag's Electronic Product Code (EPC) as in a traditional RFID authentication system, (2) the validity of the human-state chain, and (3) the continuous coexistence of the paired wrist wearable and RFID tag with Bob. Bob passes CLA if all three factors can be validated; otherwise, WearRF-CLA considers Bob absent and automatically shuts down all the physical/electronic resources opened to him. For example, if Bob walks away from his desk without returning, WearRF-CLA can recognize an illegitimate human-state chain and then consider Bob absent.

The wrist wearable and RFID tag have complementary roles in WearRF-CLA. The former contributes gyroscope data for recognizing legitimate human-state chains, but it cannot directly prove Bob's physical presence because it can communicate with the remote server through WiFi/cellular links even after Bob leaves the protected area. In contrast, the communication range of a commodity UHF RFID reader is up to 12 m, so Bob's RFID tag continuously responding to the fixed reader can prove its close vicinity. Bob, however, may accidentally lose the tag or have it stolen, e.g., by a malicious coworker in the same area. In addition, RFID signals cannot help recognize human-state chains, so the RFID tag alone is insufficient as well. Therefore, WearRF-CLA combines the merits of the wrist wearble and RFID tag by verifying their continuous coexistence with Bob. For instance, if Bob leaves his desk while leaving his RFID tag there, the coexistence can no longer be detected, indicating Bob's absence from his desk to the system.

The design of WearRF-CLA faces three critical challenges.

• Challenge 1: *how to define human-states and legitimate human-state chains in a typical indoor venue*? Fine-grained daily user activities are difficult to recognize from noisy gyroscope data and also lead to privacy concerns. Therefore, we propose to only explore five common human-states in a typical indoor venue: walk, sit, stand, sit down, and stand up. In addition, a legitimate human-state chain must have two attributes: (1) human-states are permissible, and (2) human-state transitions are legitimate in a target area. So we define a human-state transition diagram to check whether a recognized human-state chain is legitimate.

• Challenge 2: *how to recognize human-states from noisy gyroscope data*? Due to differences in physiological (e.g., body shape, height, and weight) and behavioral characteristics (e.g., walking patterns), each human subject likely performs the same activity in different ways. Hence, there exists distribution discrepancy among gyroscope

Table 1: Categorization of common human-states.

Categories	Human-states
Dynamic state	Walk
Static state	sit, stand
Transition state	sit down, stand up

data collected from different users, resulting in a model trained on one set of users suffering from significant performance degradation when applied to other users. To tackle this challenge, we first utilize supervised contrastive learning [17] to train a feature encoder network that can extract user-independent feature representations from gyroscope data. Based on the extracted feature representations, we build a multiple layer perception (MLP) classifier to recognize human-states. To the best of our knowledge, this is the first work to explore self-supervised contrastive learning for gesture recognition based on the IMU sensor data of wrist wearables.

• Challenge 3: *how to check the coexistence of the paired RFID tag and wrist wearable with the same user*? To tackle this challenge, we explore the observation that the correlation between RFID-phase and gyroscope data rely on particular human-states because the RFID tag and wrist wearable are on different body positions. We classify such correlations into three classes: strong, weak, and uncorrelated. Since the gyroscope and RFID-phase data are of different types, we design a two-stream cross-modal deep neural network (DNN) to learn their correlations. By crosschecking recognized human-states and learned phase-gyroscope data correlations, we can verify if a user simultaneously carries the RFID tag and wrist wearable.

We prototype WearRF-CLA on commodity smartwatches and UHF RFID systems and evaluate its performance with 10 volunteers in two university offices. The over classification accuracy for the human-state classifier is about 98.5%, which significantly outperforms traditional machine learning models. In addition, the true and false acceptance rates with the cross-modal DNN for data correlation measurement are 92.25% and 0.1%, respectively. The results confirm that WearRF-CLA can effectively detect the physical presence or absence of legitimate users. In addition, WearRF-CLA can achieve an average authentication latency of less than 213 ms, so it can well satisfy the real-time CLA requirement.

The rest of this paper is organized as follows. §2 outlines the system model and workflow. §3 presents the adversary model.§4 details the WearRF-CLA design. §5 presents the experimental evaluation. §6 concludes this paper.

2 SYSTEM OVERVIEW

WearRF-CLA aims to continuously and automatically authenticate users for accessing physical/electrical resources after they enter a protected indoor area (e.g., workplace). From the hardware perspective, WearRF-CLA consists of a backend server, RFID readers, wrist wearables, and RFID tags. We outline the WearRF-CLA operations with Bob as an exemplary legitimate user in an multi-user workplace as shown in Fig. 1. We deploy one or more antennas connected to an RFID reader on the office wall or ceiling. Bob has an RFID tag which is worn on a lanyard or clipped to his clothes. The tag has been enrolled into the server and communicates with RFID readers by backscattering its signals. The RFID reader has a limited

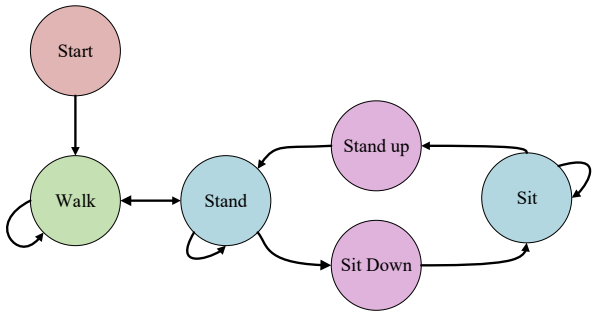


Figure 2: Human-state transition diagram.

communication range covering the indoor venue, while the system server may simultaneously server many indoor areas such as different rooms in a large facility. Bob also has a password-protected smartwatch (or fitness tracker) with a standard inertial gyroscope. He installs a WearRF-CLA app on it and creates the WearRF-CLA app username and password for the server to recognize him. The server associates the smartwatch with his tag.

We assume that Bob's smartwatch can always communicate with the server either through a direct WiFi/cellular channel or with a paired smartphone as a relay. The WearRF-CLA app handles all the communication messages between Bob's smartwatch and the server. So we assume a cryptographically secure end-to-end TLS-like channel between the WearRF-CLA app and server. Specifically, when the server receives authenticated messages from an WearRF-CLA app instance logged into under Bob's username and password, it trusts that the messages are indeed from Bob's smartwatch.

The WearRF-CLA workflow is as follows. (1) After passing one-time authentication such as swiping his ID card or inputting a PIN on the doorpad, Bob enters the protected workplace while wearing his smartwatch and RFID tag. (2) The RFID reader acquires the RFID EPC according to the standard EPC Gen2 protocol [**?**] and then sends it to the server for verification. (3) After validating the EPC associated with a legitimate user Bob, the server notifies the RFID reader to continuously report the phase information of backscattered RFID signals and also Bob's enrolled smartwatch to submit its gyroscope data at regular time interval (say, 15 s) without involving any Bob's effort. (4) The server keeps inferring a sequence of human states (e.g., stand, sit, and walk) from the noisy gyroscope data. Based on human-states, the server further checks the relationship between the RFID-phase and smartwatch-gyroscope data. Bob is considered still physically present if both the human-state chain and phase-gyroscope data relationship are deemed normal. Otherwise, he is considered no longer present in the target indoor area and automatically logged out (or deauthenticated).

3 ADVERSARY MODEL

For lack of cryptographic support, most commodity RFID systems are particularly vulnerable to tag cloning. In particular, a capable attacker can use a commodity RFID reader or software-defined radio to easily overhear unencrypted tag information transmitted

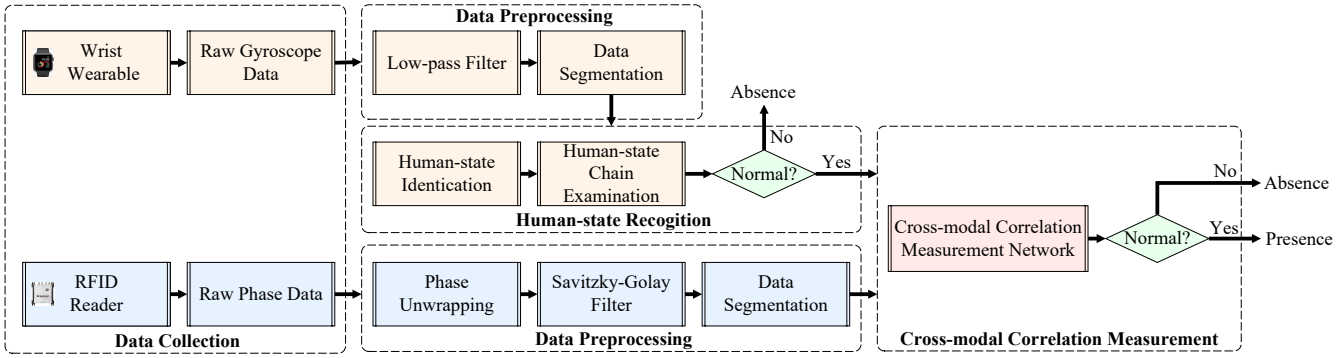


Figure 3: System modules.

between a legitimate RFID tag and the reader. Then he/she can create a cloned tag by writing the sniffed tag information onto a blank commodity RFID tag. Therefore, we consider a reasonable insider adversary (e.g., a malicious coworker), denoted by $\mathcal{A}$, that has a clone of Bob's RFID tag. $\mathcal{A}$ is inside the protected area and aims to impersonate Bob by faking Bob's physical presence with the cloned tag or Bob's genuine tag accidentally lost/stolen. Bob's smartwatch is password-protected. Each time Bob puts on his smartwatch, he must enter the unlocking password and launch the WearRF-CLA app therein which can run in the background until the smartwatch leaves Bob's wrist. Therefore, $\mathcal{A}$ still cannot bypass WearRF-CLA by stealing both Bob's smartwatch and RFID tag as long as the smartwatch's password mechanism is secure.

We use one example to illustrate $\mathcal{A}$'s objectives. Assume that Bob forgets to lock his computer when leaving the workplace. $\mathcal{A}$ attempts to fake his continual presence to access Bob's computer or other physical/electronic resources opened to Bob. Bob's smartwatch can still respond to the system server over WiFi/cellular links even after leaving the workplace. $\mathcal{A}$ can respond to the RFID reader's query with the cloned tag or even Bob's genuine tag accidentally lost/stolen. $\mathcal{A}$ succeeds if the server detects legitimate human-state chains and normal phase-gyroscope data relationships from received smartwatch-gyroscope data and RFID-phase data. $\mathcal{A}$ is further classified into the following two types.

Type-1 adversary: random mimicking. $\mathcal{A}$ is assumed to know how WearRF-CLA works and wears the cloned tag to initiate a CLA session. But $\mathcal{A}$ cannot monitor Bob's real-time activities and just performs random human activities such as walking, sitting, and typing on a computer keyboard. Since the tag EPC is authentic and verifiable, the server pulls the phase and gyroscope data from the RFID reader and Bob's smartwatch, respectively. It then infers human-states from gyroscope data and checks the relationship between the phase and gyroscope data as usual. WearRF-CLA is designed to be highly autonomous, so Bob may be not aware that his smartwatch has uploaded gyroscope data that relate to his activities.

Type-2 adversary: synchronized mimicking $\mathcal{A}$ can observe Bob's real-time activities either in person (e.g., as a malicious bystander or coworker) or through a live feed from a spy camera. Then $\mathcal{A}$ attempts to synchronize his activities with Bob.

For both adversary types above, we assume that the server cannot use secure localization techniques to tell whether Bob's smartwatch and the RFID reader the adversary attempts to cheat are in the same target indoor area, as such techniques often rely on many assumptions which may not hold in practice. In addition, we do not consider denial-of-service attacks, in which the adversary seeks to induce wrong RFID phase measurements and thus authentication failures by signal interference.

4 SYSTEM DESIGN

In this section, we first describe human-states and the human-state transition diagram. Then we overview WearRF-CLA system modules. Finally, we detail the design of each module.

4.1 Human-states and Human-state Transitions

Although numerous human activity recognition techniques based on smartphone's IMU sensors have been proposed, fine-grained human daily activities are difficult to recognize from noisy gyroscope data due to many challenges such as distribution discrepancy and annotation scarcity [8]. In addition, since each WearRF-CLA user wears his[1] RFID tag on lanyard or clips it to his clothes, backscattered RFID signals' phase can relate to coarse-grained human-body states (e.g., walking or stationary). As mentioned in Section 1, we determine whether a user simultaneously wears his RFID tag and wearable by checking the relationship between the RFID-phase and gyroscope data. Hence, we do not need to identify fine-grained human activities from gyroscope data. Moreover, fine-grained human daily activity recognition can raise privacy concerns. Therefore, we explore common human states in typical indoor workplaces for CLA. As shown in Table 1, we consider five human-states: walk, sit, stand, sit down, and stand up. Based on their characteristics, we classify them into three categories: dynamic, static, and transition states. According to our observation, almost all human activities in a common indoor workplace are based on these human-states. For example, when a user sits at his desk, he may be talking to his colleagues, typing on a computer keyboard, or writing on paper. So we classify all such human activities into the sitting state.

Fig. 2 shows the human-state transition diagram. We consider some natural transitions between two human-states such as stand→sit down→sit. Additionally, we notice that users usually start walking (e.g., to their desks) after entering a typical indoor venue in most

[1]No gender implication.

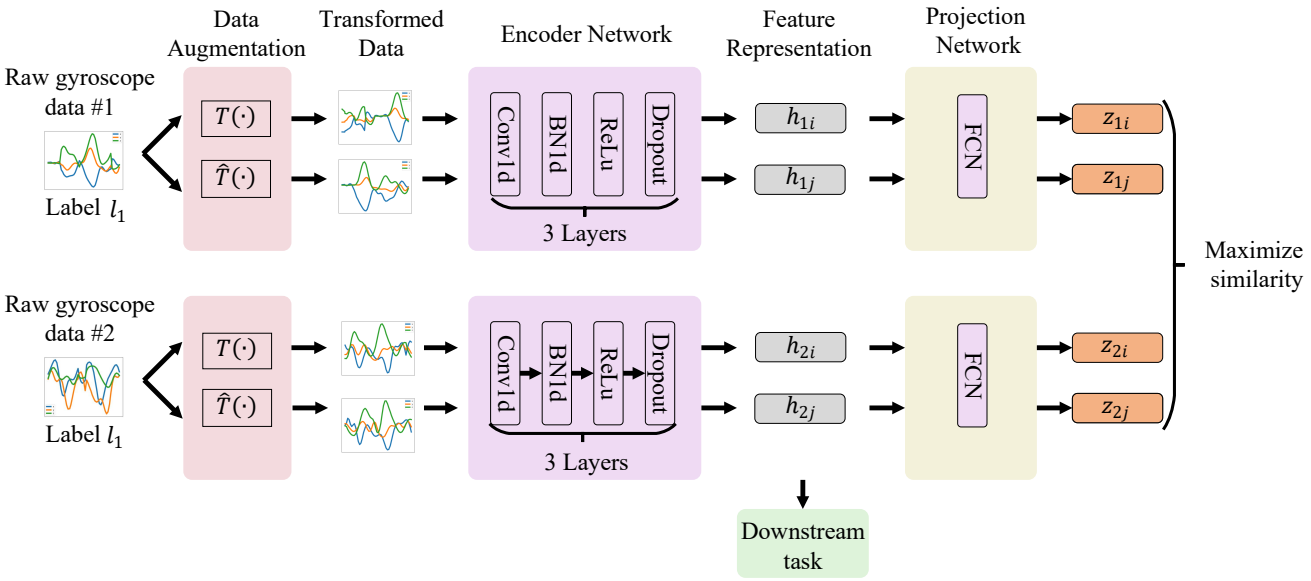


Figure 4: Supervised contrast learning framework for feature extraction .

cases. Therefore, we use the walking state as the initial human-state in a legitimate human-state chain.

4.2 System Modules

As shown in Fig. 3, WearRF-CLA consists of five main modules. In the Data Collection module, the server instructs the RFID reader and the enrolled wearables to continuously record and submit phase and gyroscope data at a regular time interval (say, 15 s), respectively. In the Data Preprocessing module, the server uses different techniques to remove the noise from raw phase and gyroscope data. After this, WearRF-CLA utilizes the fixed-size sliding window method to segment data. In the Human-state Recognition module, the server recognizes and extracts human-states from the processed gyroscope data. It also checks whether the human-states are permissible and also whether the human-state chain is legitimate in the target indoor area. If so, it feeds the processed gyroscope data to the last Cross-modal Correlation Measurement module; otherwise, the user is considered no longer present. In the last module, the system explores a two-stream DNN to check the relationship between the phase and gyroscope data. WearRF-CLA declares the physical absence/presence of the user in the target indoor area based on the output of the last module.

4.3 Data Collection

WearRF-CLA starts detecting the physical presence of a legitimate user after the RFID reader validates his tag EPC. In particular, the RFID reader continuously transmits a continuous wave (CW) to activate tags within its read range. The RFID tag automatically responds to the reader's query by backscattering its unique EPC. The reader-tag communications follow a standard RFID protocol such as EPC Gen2. [?]. After receiving the tag response, the reader decodes it and then sends the tag EPC to the backend server. The server then queries its database. If a matching user (i.e., Bob in our example) is found, the server sends a "Start Sensing" notification to both the reader and Bob's enrolled wearables. The overall latency from the RFID tag responding until the "Start Sensing" command reaches the reader and wearable is usually very short and well below 0.5 s per the EPC Gen2 standard and our experiments. Once receiving "Start Sensing", the RFID reader and wrist wearable begin to record and submit the phase and acceleration data in a regular authentication interval (say 15 s), respectively, which is predetermined according to the CLA time-granularity requirement. The server issues a "Stop Sensing" command to the reader and wearable if the RFID reader cannot detect the tag or the user is considered not present.

4.4 Data Preprocessing

The server feeds raw RFID-phase and gyroscope data into the Data Preprocessing module. Since the data are collected from two different modalities, they are processed in separate ways.

As shown in Fig. 3, the raw phase data go through phase unwrapping, Savitzky-Golay filtering, and data segmentation in sequence. In particular, the raw phase values are wrapped within $[0, 2\pi]$ and result in range ambiguity. So we correct each phase value in radians by adding or subtracting 2π if absolute differences between two consecutive phase values are greater than or equal to π. Then we use a Savitzky-Golay smoothing filter [34] to smooth the unwrapped phase data and also remove random noise. This filter uses least-squares fitting to perform a local polynomial regression in a subset of neighboring points. It can preserve some crucial distribution features like relative maxima and minima, making it more suitable than other filtering methods for our case.

For the gyroscope data, we adopt a low-pass filter with a cutoff frequency of 3 Hz to eliminate noise and interference. The frequency of people walking is lower than 3 Hz [15]. Additionally, according

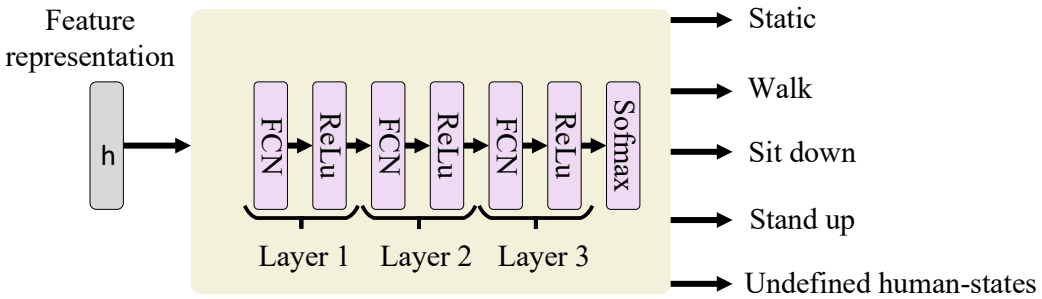


Figure 5: Downstream task: MLP classifier for human-state recognition.

to our experiments, the frequency of transition states varies around 0.5 Hz to 1.25 Hz. So we set up the cutoff frequency as 3 Hz to all three axes of gyroscope data to remove the high-frequency noise and interference.

After removing noise and interference, we use a fixed-length sliding window method with 50% overlap to segment the phase and gyroscope data, respectively. Previous studies show that the size for fixed windows ranges from 2 s to 5 s for a frequency of 20 Hz to 50 Hz [6, 36]. In addition, the average duration of a single dynamic or transition state (e.g., a gait step or sitting down) is less than 2 s. We thus segment the phase and gyroscope data into frames of 2 s with 50% overlap.

4.5 Human-state Recognition

Fig. 3 shows that this module has two components: Human-state Identification and Human-state Chain Examination. The former aims to recognize predefined human-states from a sequence of gyroscope data segments and output a human-state chain. The latter then seeks to check the validity of the human-state chain. The following steps are performed in sequence.

4.5.1 Human-state identification. Prior attempts adopt traditional machine learning or deep learning approaches to recognize human activities from IMU sensor data such as accelerometer and gyroscope data [12–14, 22, 32]. However, since different users perform the same activity in different ways, these pre-trained models are highly specific to users. Recent works explore transfer learning to improve cross-user activity recognition performance [5, 7, 9, 31]. However, transfer learning requires extra training efforts for each user. A natural question that arises is that whether we can train a "one-size-fits-all" model that is able to be trained once but used anywhere. The key challenge is how to extract user-independent feature representations from sensor data for activity recognition. In this work, we explore supervised contrastive learning [17] to tackle this challenge. This learning technique aims to learn latent feature representations from raw input data by contrasting positive pairs (samples from the same class) against negative pairs (samples from different classes). So the latent feature representations from the same class should stay close to each other, while the feature representations of different classes far apart. Fig. 4 shows the framework of supervised contrastive learning, which comprises the following four components.

• **Data augmentation**. For each batch of raw input data, we transform it into two augmentations, each of which represents a different view of the data. In this paper, we utilize a set of data transformation functions in [39] and [32].

1. Jittering. To imitate noise induced by hardware, software, and environment, random Gaussian noise with zero mean and standard deviation of 0.05 are added to a data sample.
2. Scaling. The magnitude of a data sample is scaled by a random factor that is sampled from a Gaussian distribution of mean being 1 and standard deviation being 0.1.
3. Rotation. A rotation matrix for a random rotation angle around a random axis is generated for each data sample. Both the rotation angle and axis are drawn with a uniform distribution. The generated rotation matrix is then applied to the data sample.
4. Permutation. The data sample is first divided into N (e.g., 4) segments that have the same length. Then we randomly permute the segments to generate a new data sample .
5. Time-warping. This transformation function stretches or compresses a data sample by smoothly varying the time intervals of the original data samples.
6. Magnitude-warping. This transformation changes the magnitude of a data sample by multiplying a smooth curve varying around one (e.g., cubic spline curve).
7. Channel Shuffling. The axial dimensions of a data sample are shuffled at random.

• **Encoder network** $f(\cdot)$. This component aims to extract high-level feature representations from augmented data samples. As shown in Fig. 4, we instantiate Encoder network as a three-layer convolutional neural network. Each layer consists of a 1D convolutional layer, a batch normalization Layer, a Rectified Linear Unit (ReLU), and a dropout layer.

• **Projection network** $g(\cdot)$. It maps the high-level feature representations into another space for contrastive learning. We utilize a multilayer perceptron with one hidden layer of size 1024 to convert the high-level feature representations to a one-dimensional vector.

• **Supervised contrastive loss function**. A contrastive loss function defines the learning object for supervised contrastive learning. Consider a dataset of N pairs of data samples and labels, denoted as $\{x_k, y_k\}_{k=1,2,\ldots,N}$. For each data sample in the dataset, we perform two augmentations, resulting in $2N$ pairs of

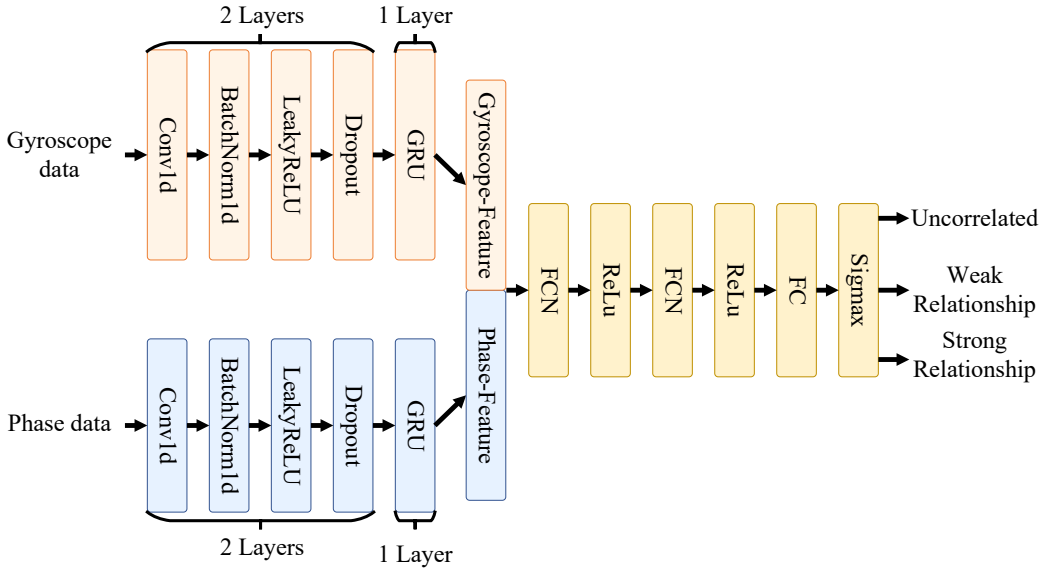


Figure 6: Cross-modal correlation measurement.

augmented samples and labels, denoted as $\{\tilde{x}_l, \tilde{y}_l\}_{l=1,2,...,2N}$, where $\tilde{x}_{2k}$ and $\tilde{x}_{2k-1}$ are two augmentations of $x_k (k = 1, 2, ..., N)$ and $\tilde{y}_{2k} = \tilde{y}_{2k-1} = y_k$. Let $I \equiv \{1, 2, ..., 2N\}$ be the index set of augmented samples. According to [17], the supervised contrastive loss function, called SupCon, can be written as:

$$\mathcal{L} = -\sum_{i \in I} \frac{1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(z_i \cdot z_p / \tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a / \tau)}. \quad (1)$$

where $A(i) \equiv I \setminus \{i\}$ denotes the index set of augmented samples that does not include i, the index i is called the anchor, $z = g(f(\tilde{x}))$, $P(i) \equiv \{p \in A(i) : \tilde{y}_p = \tilde{y}_i\}$ represents the index set whose labels are the same as the label of ith data sample, $|P(i)|$ is the cardinality of $P(i)$, $\tau \in \mathbb{R}^+$ is a scalar temperature parameter, and the $\cdot$ symbol is the inner product. According to [17], lower temperature benefits training more than higher ones, but extremely low temperatures are harder to train due to numerical instability.

Downstream task. After completion of training, we discard the projection head $g(\cdot)$ and use encoder $f(\cdot)$ to extract high-level representations h for the downstream task, which refers to human-state recognition in this paper. As shown in Fig. 5, we build a multilayer perception (MLP) classifier with three hidden layers for human-state recognition. It takes as input feature representations extracted by the encoder $f(\cdot)$ and outputs human-state labels.

4.5.2 Human-state chain examination. After obtaining a human-state chain, WearRF-CLA starts verifying its validity. Specifically, based on the predefined human-states in Table 1, it first checks if the human-state chain contains some undefined human-states in a normal typical indoor workplace. If so, WearRF-CLA consider the legitimate user absent to be on the safe side. Otherwise, the system further checks if the transitions between every two consecutive human-states is normal based on the human-state transition diagram shown in Fig. 2. If so, it feeds the processed phase data, gyroscope data, and the human-state chain to the Cross-modal Correlation Measurement submodule.

4.6 Cross-modal Correlation Measurement

Finally, we build a *cross-modal correlation measurement network* to check the relationship between the processed phase and acceleration data by exploring deep learning techniques. Fig. 6 shows the architecture of the proposed DNN architecture. It takes the processed phase and gyroscope data as inputs. We pad zeros in the end (if necessary) to ensure that all input vectors/matrices are of the same length. The network has two streams that can extract features from the phase and gyroscope data, respectively. Both network streams consist of two one-dimensional (1D) convolutional layers and a Gated Recurrent Units (GRU) layer. Each 1D convolutional layer is followed by a batch normalization Layer, a Leaky Rectified Linear Unit (LeakyReLU) activation layer, and a dropout layer. The two 1D convolutional layers extract features from the input vector/matrix, and the GRU layer encodes the sequence of features to a vector representation. At the end of the two branches, we concatenate the feature map of each stream in the phase and gyroscope domain. We then feed the concatenated feature map to three fully connected (FC) layers with a softmax activation function to produce a likelihood vector that represents the probability distribution of a list of potential relationship between the two data. After this, we select the one that has largest values as the relationship type of the input phase and gyroscope data. Finally, we determine if the relationship is normal or not based on the human-state chain. Specifically, there are three cases: (1) if the user is in the static state (e.g., sitting and standing), the relationship between the phase and gyroscope data should be weak; (2) if the user is in the dynamic or transition state, the relationship between them should be strong; and (3) if we consider the aforementioned adversaries in Section 3,

the relationship between them should be uncorrelated. If a relationship type is consistent with the corresponding human-state in the human-state chain, the legitimate user is considered still present in the typical indoor area and otherwise absent.

5 PERFORMANCE EVALUATION

5.1 Experimental Setup

We evaluate WearRF-CLA in this section. In what follows, we first describe our experimental setup. Then we evaluate the performance of WearRF-CLA in two university offices as shown in Fig. 7.

Hardware implementation. We implemented a prototype of WearRF-CLA with commodity UHF RFID devices and smartwatches. Specifically, we used an Impinj Speedway R420 RFID reader with two circularly polarized RFID antennas. We connected the reader to a Dell Precision laptop that serves as the backend server and mounted antennas on two tripods. As shown in Fig. 7, the two antennas were deployed in two university offices with the sizes of 22 ft × 16 ft and 13 ft × 10 ft, respectively. In addition, each user was required to wear a smartwatch and a Zebra UHF RFID tag on his or her lanyards in our experiments. But our system is applicable to any other commodity UHF RFID tags such as Aline ALN-9640 RFID tags and Omni-ID's Adept 650P RFID cards.

Software implementation. To capture human states, we used three different types of smartwatches, including Huawei Watch 2 that runs Android Wear 2.1, Samsung Galaxy Watch that runs Tizen 5.0, and Apple Watch 2 that runs WatchOS 7. To collect gyroscope data, we implemented Android, Tizen, and WatchOS applications on Huawei Watch 2, Galaxy Watch, and Apple Watch 2, respectively. To save energy, the gyroscope sampling rate was set to 20 Hz. We also implemented a Java application based on Octane SDK [3] together with the reader to obtain the RFID phase data. The application on the smartwatches and the phase-recording application are both part of the Data Collection submodule in WearRF-CLA. In addition, we implemented Data Preprocessing, Human-state Recognition, and Cross-modal Correlation Measurement modules using Python on the laptop. Finally, we constructed the proposed DNN architecture in PyTorch 1.8 [26] and trained it on Dell 7920 Tower with Quadro RTX 5000 16GB GPU.

5.2 Data Collection

With the Institutional Review Board (IRB) approval from our institution, we recruited 10 participants for the experiments, including 4 females and 6 males aged between 20 and 35. We consider eight common daily activities in an indoor workplace: (1) walk, (2) sit and type, (3) sit and write, (4) sit and talk, (5) stand and type, (6) stand and talk, (7) sit down, and (8) stand up. As mentioned in Section 4.1, we labeled (2)-(4) and (5)-(6) activities as sitting and standing states, respectively. In addition, we used three publicly available datasets to cover a wide variety of human subjects and daily activities for human-state recognition. Finally, We collected the following two training datasets (Dataset I and II) and two test datasets (Datasets III and Datasets IV).

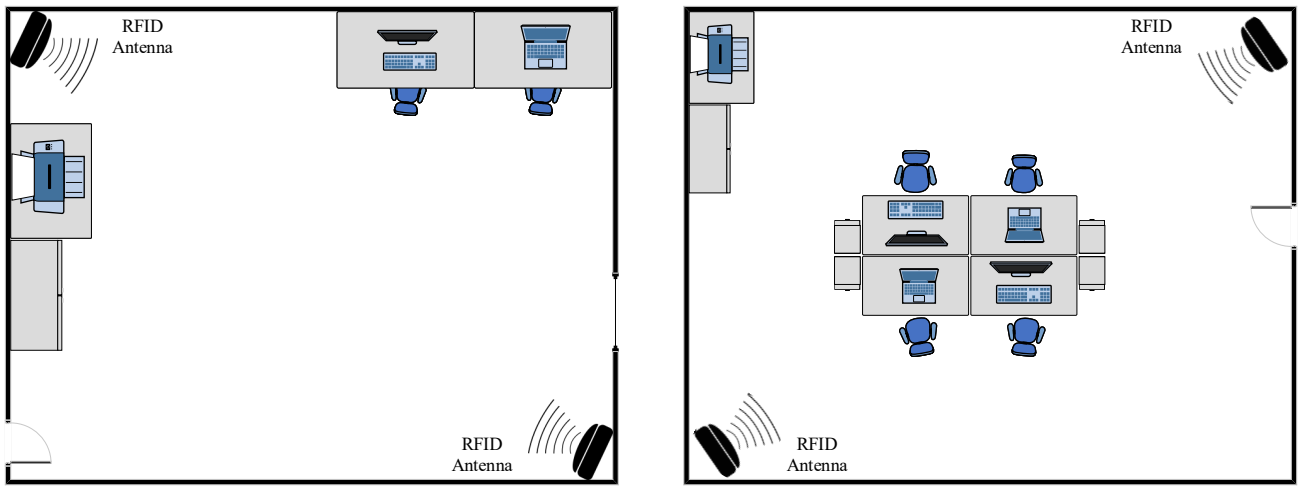
Dataset-I for human-state recognition. We asked a participant $\mathcal{P}$ to perform the aforementioned human daily activities in office A. Specifically, he performed "sit down" and "stand up" 3,800 times, respectively. He was also asked to perform the other daily activities. Finally, we collected about 24 hours of phase and gyroscope data for the walking state and 16 hours in total for sitting and standing states . We fed the raw phase and gyroscope data into the Data Preprocessing module to obtain the processed phase and gyroscope data segments. We also considered the following three public datasets to incorporate more daily activities and human subjects.

- Heterogeneity human activity recognition (HHAR) dataset [38]. There are six human activities in this dataset: sitting still, standing still, walking, biking, stair up, stair down. Nine subjects were required to wear Samsung and LG smartwatches on each arm and conduct five minute of each activity. We labeled the last three activities as undefined activities in the two offices. The sampling rates of Samsung and LG smartwatches are 200 Hz and 100 Hz, respectively.
- PAMAP2 [29]. This dataset was obtained from a group of 9 participants with a wrist-worn Colibri wireless IMUs from Trivisio [2]. The gyroscope signals are sampled at 100 Hz when participants performed the following 21 activities: sitting, standing, walking, lying, running, cycling, Nordic walking, watching TV, computer work, car driving, ascending stairs, descending stairs, vacuum cleaning, ironing, folding laundry, house cleaning, playing soccer, rope jumping. Most activities were performed over three minutes. Except for the first three activities, we classified all others as undefined activities in our experiments.
- UT-Complex [36, 37]. This dataset was collected from 10 participants with a smartphone on their wrist to emulate smartwatches or wrist-worn devices. The data were recorded for 13 activities: sitting, standing, walking, sitting and eating, sitting and writing, sitting and typing, sitting and drinking coffee, sitting and giving a talk, standing and smoking, jogging, biking, upstairs, downstairs. The last four were labeled undefined activities. All these activities were perform about three to six minutes by each participant.

Table 2 summarizes some details of the above three datasets. Since the data sampling rate in the three dataset ranges from 50 Hz to 200 Hz, we downsampled them at 20 Hz. Then we used the Data Preprocessing module to remove noise and divide denoised data into a series of segments. Finally, we combined them with $\mathcal{P}$'s processed gyroscope data to build Dataset-I.

Dataset-II for training cross-modal correlation measurement network. We used processed phase and gyroscope data of $\mathcal{P}$'s activities for training the Cross-modal Correlation Measurement Network. In particular, we labeled phase-gyroscope data pairs under dynamic and transition states as 2 and those under static states as 1. We further asked another participant $\mathcal{P}'$ to perform daily activities as $\mathcal{P}$ did in office B to collect phase and gyroscope data samples. A phase data sample from $\mathcal{P}$ ($\mathcal{P}'$) and a gyroscope data sample from $\mathcal{P}'$ ($\mathcal{P}$) constitute an uncorrelated phase-gyroscope data pair, which are labeled as 0.

Dataset-III for evaluating human-state identification and cross-modal correlation measurement. 10 participants were asked to perform the aforementioned daily activities at least about 1.5 hours in the two offices. Nine of them performed the activities in



(a) office A (b) Office B

Figure 7: Experimental Setup.

Table 2: Summary of public datasets

Dataset	No. of users	No. of Activities	Devices
HHAR	9	6	Samsung smartwatch and LG smartwatch
PAMAP2	9	19	Colibri wireless IMUs
UT-Complex	10	13	Samsung Galaxy S2

office A, while two carried out the activities in Office B. Specifically, to imitate multi-user workplaces, we asked two groups of three participants to perform the activities in office A at the same time slot. To imitate sing-user workplaces, three other participants to perform the activities at different time slots. Additionally, in office B, two participants were asked to perform the activities in the same time slot. After obtaining the raw data, we used the Data Preprocessing module to remove noise and segment the preprocessed data. Similar to Dataset-II, we labeled the phase-gyroscope data pairs from each participant under dynamic states and transition states as 2 and those under static states as 1.

Dataset-IV for user and attacker emulation. Five participants acted as either legitimate users or attackers to generate this dataset. For each volunteer who served as an attacker, the other 5 were regarded as his/her victims. In particular, we first asked them to act as Type-1 attackers. So they just randomly performed some daily activities. Then each participant served as a Type-2 attacker. Specifically, each attacker was physically co-located with the victim and could clearly observe the victim's hand movement. They observed and practiced victims' daily activities (e.g., gait patterns) until he/she was familiar with them. Afterwards, we required them to mimic victims' daily activities in real time. For both cases, we asked all participants to perform the activities for 15 minutes. Therefore, the collected paired phase-gyroscope samples were assigned as 0.

5.3 Performance Metrics

We use the confusion matrix to evaluate human-state recognition. A confusion matrix is a $N \times N$ table that summarize prediction results on a classification problem. N is the number of target classes. Each row of the table represents the instances in an actual class, and each column represents the instances in a predicted class.

We adopt True Acceptance Rate (TAR) and False Acceptance Rate (FAR) as the main performance metrics to evaluate the cross-modal correlation measurement network. TAR is the ratio between correctly classified positive (legitimate) instances and all positive ones in a test dataset. A higher TAR means that the system is more likely to admit legitimate users. FAR is the ratio between wrongly classified negative (adversarial) instances and all negative ones in a test dataset. A lower FAR means that the system can more effectively prevent adversaries from being physical present in a protected indoor area.

5.4 Model Setting

We used Dataset-I to train a feature encoder network and a MLP classifier for human-state recognition. Additionally, we trained the proposed cross-modal correlation measurement network on Dataset-II. The parameter settings are summarized as follows.

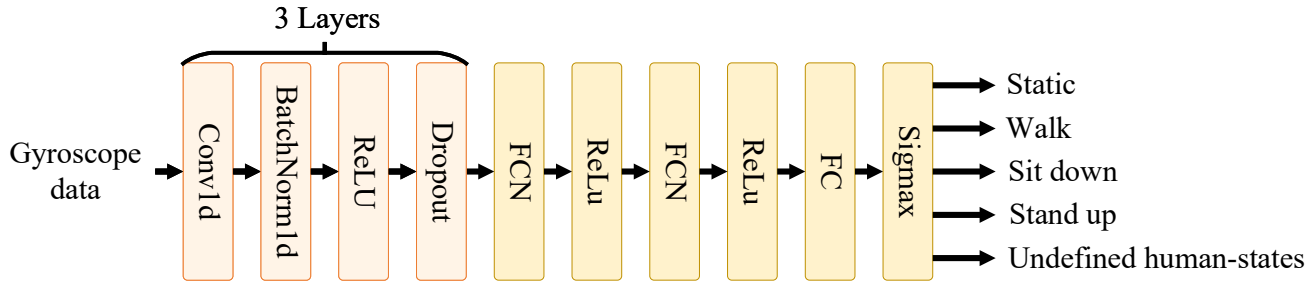


Figure 8: CNN-base classifier.

Table 3: List of features.

Domain	#	Feature Name	Description
Time-domain	1	Mean	The arithmetic mean along each axis.
	2	Standard deviation	Standard deviation along each axis.
	3	Mean absolute deviation	Median absolute deviation of the data along each axis.
	4	Skewness	Measure the asymmetry around its mean value along each axis
	5	Kurtosis	Measure the combined weight of a distribution's tails relative to the center of the distribution
	6	Root-Mean-Square (RMS)	The square root of the mean square
	7	Max	Maximum value along each axis
	8	Min	Minimum value along each axis
	9	Zero-crossing rate (ZCR)	The rate at which a signal changes from positive to zero to negative or from negative to zero to positive
	10	Absolute Energy	Summation over the squared values along each axis
	11	Variance Coefficient	Measure dispersion of a distribution, which is defined as the ratio of the standard deviation to the mean.
	12	Autocorrelation	The correlation of a signal with a delayed copy of itself as a function of delay.
	13	Pearson correlation coefficient	Measure linear correlation between two sets of data.
Frequency-domain	14	Spectral centroid	The spectral center of gravity.
	15	Spectral variance	The amount of variation of the spectrum over time
	16	Spectral skewness	Measure the asymmetry of the spectrum around its mean value.
	17	Spectral Kurtosis	Measure the flatness of the spectrum around its mean value.
	18	Spectral entropy	Measure spectral power distribution of a signal.
	19	Spectral spread	The spread of the spectrum around its mean value.
	20	Spectral Rolloff	The frequency below which 95% of the signal energy is contained.
	21	Spectral Decrease	The amount of decreasing of the spectra amplitude.
	22	Spectral slope	Measure how quickly the spectrum of a signal tails off towards the high frequencies.

Feature encoder network and projection network. As shown in Fig. 4, the feature encoder network comprises three-layer 1D CNN, and the projection network has one FC layer. In this paper, the number of hidden units in each 1D convolutional layer was set to 128. The kernel size of 1D convolutional layer was 7 with stride of 1. The dropout rate was set to 0.2 for all dropout layers. In addition, the number of neuron in the projection network was set to 1,024. In addition, the temperature τ is set to 0.11.

MLP classifier for human-state identification. After training the feature encoder network, we frozen all three layers of the feature encoder network and combined it with the MLP classifier. For the MLP classier, the number of hidden units in the first, second, and third hidden layers were set to 1,024, 512, and 256, respectively. The model is fine-tuned with Adam optimizer and a learning rate of 0.01 for 100 epochs

Cross-modal correlation measurement network. We used dataset-III to train the proposed DNN architecture. The number of hidden units in each 1D convolutional layer and GRU layer was set to 128. The kernel size of 1D convolutional layer was 3 with stride of 1. The dropout rate was set to 0.5 for all dropout layers. The number of neuron in the first and second FC layer was 120 and 80, respectively. We trained the network by minimizing binary cross entropy between the actual label and the output using the Adam optimizer [18].

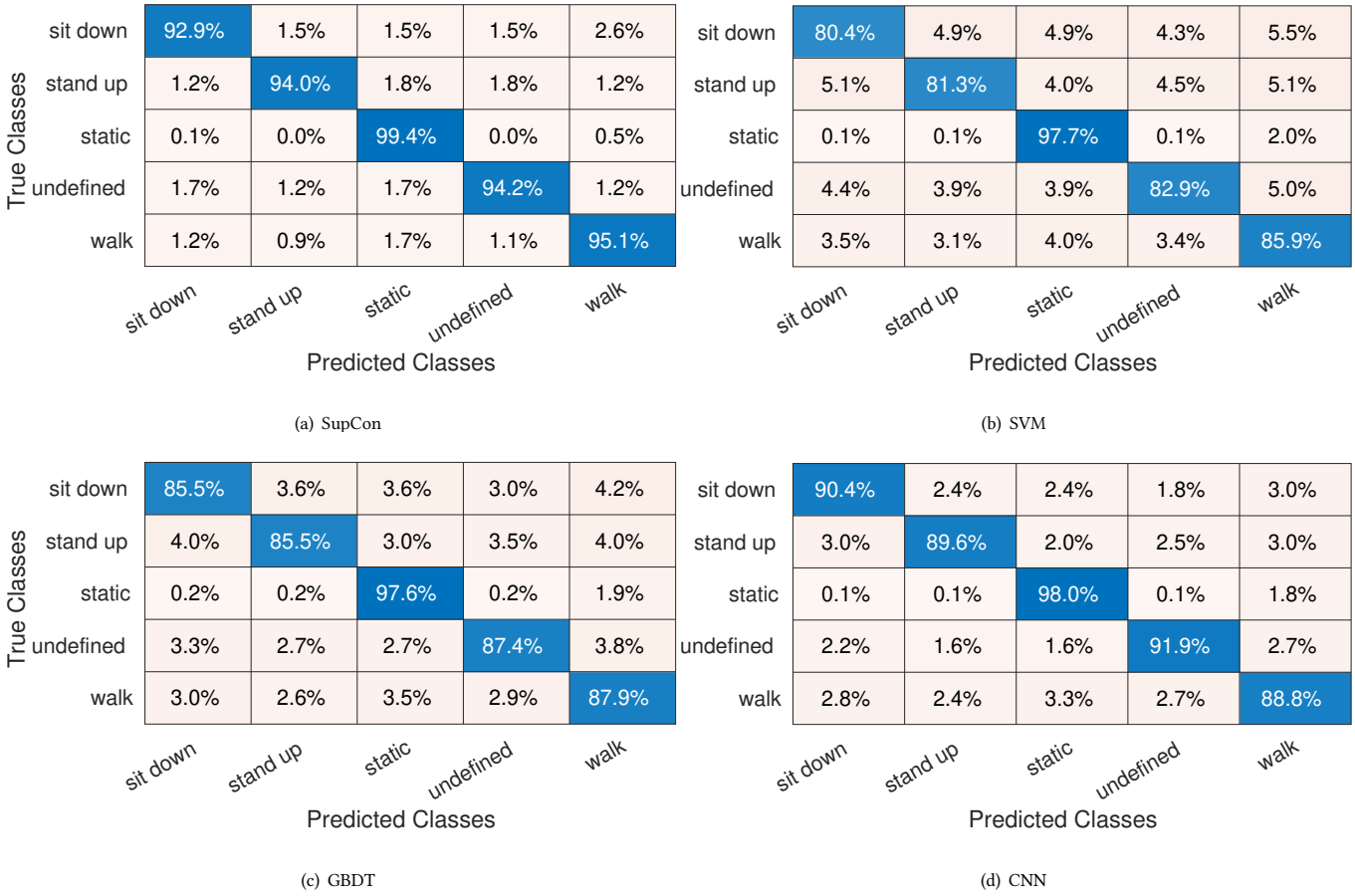


Figure 9: Confusion matrix for classifiers.

5.5 Performance Evaluation of Human-state Identification

5.5.1 Baselines. To fairly evaluate the performance of the feature encoder network and MLP classifier, we use the following models as baselines.

Support Vector Machine (SVM) [10]. SVM is one of the most widely used supervised learning algorithm. The goal of the SVM algorithm is to construct a hyperplane or a set of hyperplane in a high-dimensional or infinite-dimensional space that distinctly classifies the data points. In this paper, we adopted the one-vs-one scheme for our multi-classification problem.

Gradient boosting decision tree (GBDT) [11]. GBDT is an ensemble model of decision trees, which are trained in sequence. In this work, we trained this model by utilizing LightGBM [16], which is a fast, distributed, high-performance gradient boosting framework based on the decision tree algorithm and can be used for ranking, classification and many other machine learning tasks.

CNN-based model. We built a three-layer CNN model to identify human-states from raw gyroscope data. Fig. 8 shows the CNN architecture, in which the number of hidden units in each 1D convolutional layer was set to 128. The kernel size of 1D convolutional layer was 5 with stride of 1, and the dropout rate was set to 0.2 for all dropout layers. The number of hidden units in the first, second, and third hidden layer were set to 1,024, 512, and 256, respectively.

5.5.2 Model evaluation. We first used the gyroscope data in Dataset-I to train the aforementioned models. Then we tested the trained models on the gyroscope data in Dataset-III and Dataset-IV. In particular, to train the SVM and GBDT models, we extracted time-domain and frequency-domain features in Table 3 to represent the gyroscope segment [27]. To evaluate the impact of using different transformation for the feature encoder network and MLP classifier, we used the weighted F1 score as the evaluate metric. Based on our experiments, the highest performing models were trained by combining magnitude-warping and permutation, with an average F1 score of 0.983. So we selected this model as the human-state classifier, which is denoted by "SupCon" hereafter. Fig. 9 shows the confusion matrix of the four models in the two offices. Compared with the other three models, the SupCon model achieves a much higher overall classification accuracy. Specifically, the recognition

accuracy for walking activities are as high as 95.1% in the two offices, indicating that the SupCon model generalizes better than the other three models. The reason is that different users have different walking patterns. This also demonstrates that the SupCon model can extract user-independent features from raw gyroscope data.

5.6 Performance Evaluation of Cross-modal Correlation Measurement Network

We evaluated the cross-modal correlation measurement network on Dataset-III and Dataset-III. As mentioned in Section 4.6, we directly input the processed phase and gyroscope data segments into the trained cross-modal correlation measurement network to check their relationship. Our results show that the TAR of the deep correlation network is 92.25%. In addition, the FARs under Type-1 and Type-2 adversaries are around 0.1% and 0.2%, respectively. For Type-2 adversaries, we observed that the walking, standing-up, and sitting-down patterns of a user are very difficult to imitate, so it is almost impossible for attackers to bypass our system. Therefore, these results demonstrate that our system can detect Type-1 and Type-2 attackers with overwhelming probability.

5.7 Authentication Latency

We also studied the authentication latency of WearRF-CLA, which can be broken into two parts: the network delay to transmit gyroscope data to the server and the response time that the system needs to make a decision. In our experiment, the smartwatches uploaded gyroscope data every 15 s. The average network delay for transferring the gyroscope data is about 52 ms. In addition, the average response time is about 0.161 s. Hence, WearRF-CLA can achieve an average authentication latency of less than 213 ms.

6 CONCLUSION

This paper presented the design and evaluation of WearRF-CLA, a deep learning-based system that explores commodity UHF RFID tags and wrist wearables to continuously and automatically verify the physical presence of legitimate users in a protected indoor area without user involvement. Comprehensive experiments confirmed the high security and low authentication latency of WearRF-CLA.

ACKNOWLEDGEMENT

This work was supported in part by US National Science Foundation under grants NSF CNS-1824355/1933069/2055751. We also would like to thank anonymous reviewers for their constructive comments.

REFERENCES

[1]]UHFG218 [n. d.]. EPC UHF Gen2 Air Interface Protocol. https://www.gs1.org/standards/epc-rfid/uhf-air-interface-protocol
[2] 2012. Trivisio. https://www.trivisio.com/
[3] 2018. Octane SDK-Impinj Support Portal. https://support.impinj.com/hc/en-us/ARTICLEs/202755268-Octane-SDK
[4] 2020. Powering Human Presence Detection with Sensors. https://www.evaluationengineering.com/industries/article/21151815/powering-human-presence-detection-with-sensors
[5] Lei Bai, Lina Yao, Xianzhi Wang, Salil Kanhere, Bin Guo, and Zhiwen Yu. 2020. Adversarial multi-view networks for activity recognition. *Ubicomp* (2020), 1–22.
[6] Oresti Banos, Juan-Manuel Galvez, Miguel Damas, Hector Pomares, and Ignacio Rojas. 2014. Window size impact in human activity recognition. *Sensors* 14, 4 (2014), 6474–6499.
[7] Kaixuan Chen, Lina Yao, Dalin Zhang, Xiaojun Chang, Guodong Long, and Sen Wang. 2019. Distributionally robust semi-supervised learning for people-centric sensing. In *AAAI*. 3321–3328.
[8] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. 2021. Deep Learning for Sensor-based Human Activity Recognition: Overview, Challenges, and Opportunities. *Comput. Surveys* 54, 4 (2021), 1–40.
[9] Ling Chen, Yi Zhang, and Liangying Peng. 2020. Metier: A deep multi-task learning based activity and user recognition model using wearable sensors. *Ubicomp* (2020), 1–18.
[10] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.
[11] Jerome Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.
[12] Yu Guan and Thomas Plötz. 2017. Ensembles of deep lstm learners for activity recognition using wearables. *Ubicomp* (2017), 1–28.
[13] Sojeong Ha and Seungjin Choi. 2016. Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors. *IJCNN* (2016), 381–388.
[14] Nils Hammerla, Shane Halloran, and Thomas Plötz. 2016. Deep convolutional and recurrent models for human activity recognition using wearables. *ArXiv* (2016).
[15] Tianjian Ji and Aikaterini Pachi. 2005. Frequency and velocity of people walking. *Structural Engineer* 84, 3 (2005), 36–40.
[16] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *NIPS*. Curran Associates, Inc.
[17] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *ArXiv* (2020).
[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[19] kun Qian, Chenshu Wu, Yi Zhang, Guidong Zhang, Zheng Yang, and Yunhao Liu. 2018. Widar2.0: Passive human tracking with a single Wi-Fi link. In *MobiSys*.
[20] L. Lazos and R. Poovendran. 2004. SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks. In *ACM WiSe*. Philadelphia, PA, 21–30.
[21] Loukas Lazos, Radha Poovendran, and Srdjan Capkun. 2005. ROPE: robust position estimation in wireless sensor networks. In *IEEE IPSN*. Los Angeles, CA, 324–331.
[22] Song-Mi Lee, Sang Min Yoon, and Heeryon Cho. 2017. Human activity recognition from accelerometer data using Convolutional Neural Network. *Bigcomp* (2017), 131–134.
[23] Tao Li, Yimin Chen, Rui Zhang, Yanchao Zhang, and Terri Hedgpeth. 2018. Secure crowdsourced indoor positioning systems. In *INFOCOM*. Honolulu, HI.
[24] Tao Li, Dianqi Han, Yimin Chen, Rui Zhang, Yanchao Zhang, and Terri Hedgpeth. 2020. IndoorWaze: a crowdsourcing-based context-aware indoor navigation system. *IEEE Transactions on Wireless Communications* 19, 8 (May 2020), 5461–5472.
[25] Kaikai Liu, Xinxin Liu, and Xiaolin Li. 2013. Guoguo: Enabling fine-grained indoor localization via smartphone. In *ACM MobiSys*. Taipei, Taiwan.
[26] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).
[27] Geoffroy Peeters, Bruno Giordano, Patrick Susini, Nicolas Misdariis, and Stephen McAdams. 2011. The timbre toolbox: Extracting audio descriptors from musical signals. *The Journal of the Acoustical Society of America* 130, 5 (2011), 2902–2916.
[28] Kasper Bonne Rasmussen and Srdjan Čapkun. 2010. Realization of RF Distance Bounding. In *USENIX Security*. Washington, DC, 25–25.
[29] Attila Reiss and Didier Stricker. 2012. Introducing a new benchmarked dataset for activity monitoring. In *ISWC*. IEEE.
[30] Yanzhi Ren, Yingying Chen, Mooi Choo Chuah, and Jie Yang. 2014. User verification leveraging gait recognition for smartphone enabled mobile healthcare systems. *IEEE Transactions on Mobile Computing* 14, 9 (2014), 1961–1974.
[31] Seyed Rokni, Marjan Nourollahi, and Hassan Ghasemzadeh. 2018. Personalized human activity recognition using convolutional neural networks. *AAAT* (2018).
[32] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. 2019. Multi-task self-supervised learning for human activity detection. *Ubicomp* (2019), 1–30.
[33] N. Sastry, U. Shankar, and D. Wagner. 2003. Secure Verification of Location Claims. In *ACM WiSe*. San Diego, CA, 1–10.
[34] Abraham Savitzky and Marcel Golay. 1964. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry* 36, 8 (1964), 1627–1639.
[35] Cong Shi, Jian Liu, Hongbo Liu, and Yingying Chen. 2017. Smart user authentication through actuation of daily activities leveraging WiFi-enabled IoT. In *MobiHoc*.
[36] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul JM Havinga. 2016. Complex human activity recognition using smartphone and wrist-worn motion sensors. *Sensors* 16, 4 (2016), 426.

[37] Muhammad Shoaib, Stephan Bosch, Hans Scholten, Paul Havinga, and Ozlem Incel. 2015. Towards detection of bad habits by fusing smartphone and smartwatch sensors. In *PerCom Workshops*. IEEE, 591–596.
[38] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Prentow, Mikkel Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. 2015. Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition. In *SenSys*.
[39] Terry Um, Franz Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. 2017. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *ICMI*. ACM.
[40] Deepak Vasisht, Swarun Kumar, and Dina Katabi. 2016. Decimeter-Level localization with a single WiFi access point. In *USENIX NSDI*. Santa Clara, CA.
[41] S. Čapkun and Jean-Pierre Hubaux. 2005. Secure Positioning of wireless devices with application to sensor networks. In *IEEE INFOCOM*. Miami, FL, 1917–1928.
[42] Lei Yang, Yekui Chen, Xiang-Yang Li, Chaowei Xiao, Mo Li, and Yunhao Liu. 2014. Tagoram: Real-Time Tracking of Mobile RFID Tags to High Precision Using COTS Devices. In *ACM Mobicom*. Maui, Hawaii.