LightLock: user identification system using light intensity readings on smartphones

Mohsen A. Alawami, William Aiken, and Hyoungshick Kim

Abstract—Sensor data on a user's mobile device can often be used to identify the user for improving the security of smartphones in indoor environments. In this paper, we present a novel continuous user identification system called LightLock that collects light sensor data from a user's smartphone and analyzes them to identify a specific user using a machine learning approach. We develop a multi-model system to extract four different feature vectors: (1) absolute time series (ATS); (2) auto-correlation function (ACF); (3) level crossing rate (LCR); and (4) peak readings detection (PRD). To show the feasibility of LightLock, we implemented an Android application and evaluated the performance of LightLock on the dataset collected during a period of 20 days. LightLock achieves over 98% accuracy in identifying a specific user. LightLock also provides an accurate and cost-less alternative solution to existing approaches that require explicit user-smartphone interaction or the high energy consumption of multiple sensors.

Index Terms—Light sensor, user identification, machine learning, smartphones, indoor environments

1 Introduction

Existing smartphones are equipped with all-or-nothing security mechanisms – meaning that once an attacker bypasses the initial unlock mechanism (e.g., PIN or pattern), the attacker is able to access everything on the target device [1]. This is because the device's unlock mechanisms are usually the only defense available on smartphones; there is no secondary security mechanism to further prevent unauthorized use of smartphones. In practice, the initial lock mechanism for smartphones is not perfectly secure. For smartphone password schemes, users tend to select easy-to-remember and quick-to-draw lock patterns [2] as well as PINs [3] that are typically easy to guess. Biometric-based security mechanisms such as fingerprints [4], [5] and facial recognition [6] can identify a specific user and distinguish legitimate from anomalous users using their registered biometrics. However, such mechanisms still suffer from security issues that emerge from spoofing attacks in which biometric information can be captured and reused. Therefore, sophisticated user identification and authentication techniques have been proposed in several previous studies. Buriro et al. [7] proposed an authentication solution, Touchstroke, by focusing on two mechanisms: hand movements while holding the device, and touch-typing time while entering a 4-digit PIN/password. Lee et al. [8] also developed a model that implicitly authenticates a user via the accelerometer and gyroscope sensors, and the model achieved 98.1% performance accuracy.

However, user identification and authentication using behavioral biometrics and/or location have two major drawbacks. First, touch-based smartphone approaches [9], [10], [11], [12] implicitly identify users' identity by continuously monitoring their behaviors via the touchscreen (e.g., finger movement, speed, pressure, typing patterns, etc.). Therefore, explicit user interactions on smartphone are required to collect multiple types of touch data. As a result, this may provide an opportunity for an attacker to steal a victim's sensitive data before detecting the attacker's intrusion or disable such a detection scheme itself. Second, sensor-based approaches [13], [14], [15], [16], [17] rely on data driven from a fusion of multiple sensors built into smartphones (e.g., GPS, Wi-Fi, accelerometer, gyroscope, magnetometer, etc.) to improve the performance of identification/authentication schemes. However, these schemes inherently suffer from high energy consumption due to the constant usage of multiple sensors.

In this paper, we aim to develop an anomaly detection system called LightLock that can automatically be trained on a user's behaviors on his or her smartphone and implicitly detect unauthorized use of the user's device. We particularly focus on the possibility of light sensor data collected from users' smartphones not only because high-end smartphones contain sensitive light sensors that are enabled by default and provide accurate discrete values but also because indoor environments already contain sufficient lighting infrastructures. We present a novel light-based indoor user identification system that can accurately distinguish one legitimate user's behaviors from the behaviors of the anomaly user who uses the same smartphone. Because the identification process can be converted into the problem of user behavior classification, we designed light-based features which continuously verify users' identities with the goal of improving the security of indoor environments. LightLock attempts to answer the question: can light

Manuscript received August 24, 2019; revised October 04, 2019; accepted October 25, 2019.
Mohsen Ali Alawami is with the Department of Electrical and Computer Engineering, College of Information and Communication Engineering, Sungkyunkwan University (SKKU), 16419, South Korea, (e-mail:mohsencomm@skku.edu).
William Aiken was with the Department of Electrical and Computer Engineering, College of Information and Communication Engineering, Sungkyunkwan University (SKKU), 16419, South Korea, (e-mail:billzo@skku.edu).
Hyoungshick Kim is with the Department of Software, College of Software, Sungkyunkwan University (SKKU), 16419, South Korea, (e-mail:hyoung@skku.edu, corresponding author).

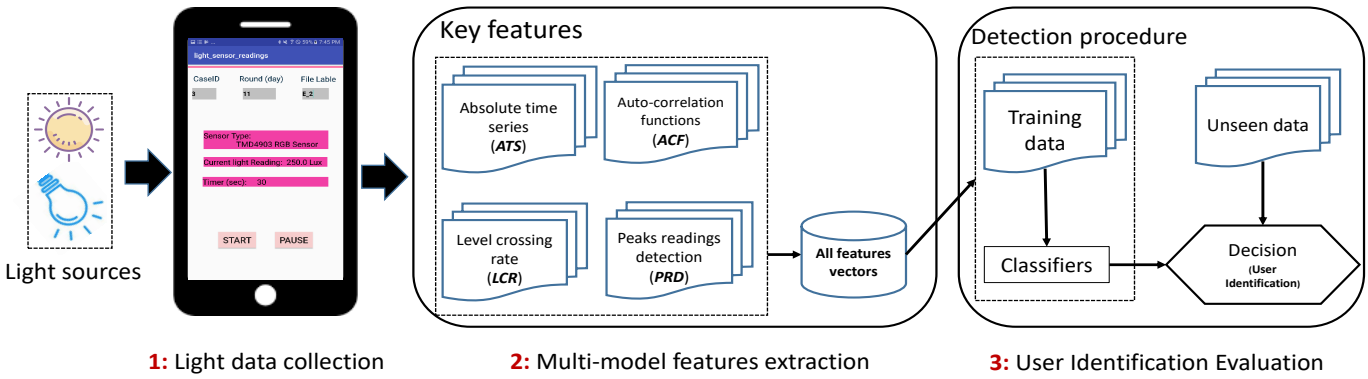


Figure 1: A high-level design of LightLock, showing the three key parts involved in user identification.

sensors on smartphones identify the identity of the user in indoor environments? Answering this question requires developing a detection system that is realistic, ubiquitous, does not require any user-smartphone interactions or additional devices, and provides high detection accuracy. Briefly, our contributions are summarized as follows:

1) We propose the idea of distinguishing users' behaviors in indoor environments by leveraging only light measurements and the users' smartphones. We further propose that such a system (which we refer to as LightLock) can be achieved via the fusion of four designed modules: (1) Absolute Time Series (ATS), (2) Auto-Correlation Function (ACF), (3) Level Cross Rate (LCR), and (4) Peak Readings Detection (PRD).
2) We performed an evaluation of LightLock on a real-world dataset representing 13 diverse behaviors collected by two users inside our university building over the course of 20 days at four different time profiles (Morning, Afternoon, Evening and *Allday*). Our evaluation included multiple combinations of these profiles under three different evaluation stages. In each evaluation stage, the extracted features from each module were used to train machine learning techniques (e.g., Support Vector Machines) to evaluate the system on various combinations of unseen data.

The rest of this paper is organized as follows. In Section 2, we briefly explain the LightLock system design, and in Section 3, we provide a detailed analysis of the four system modules. We evaluate and discuss the performance of our work in Section 4. In Section 5, we present the results of a separate lab study for analyzing the effects of environmental conditions. In Section 6, we discuss related work, and we conclude the paper in Section 7.

2 LightLock System Design

The proposed LightLock system is based on the exploitation of scattered visible lights in indoor environments, either natural or artificial lights sources. Fig. 1 shows the LightLock system architecture which consists of three cascaded parts. These parts include (1) the process of light data collection using our developed Android application by utilizing light sensors that are always running by default, (2) the analysis of extracted features using four different light modules, and (3) the detection procedure to evaluate the system using machine learning algorithms. Generally, all high-end smartphones contain built-in light sensors that provide discrete readings of the ambient environment's light intensity. In addition, all indoor environments are installed with an artificial lighting infrastructure and typically designed with windows and doors to allow for the entrance of natural light as well.

2.1 Light data collection

Here, we describe the details of the light data collection process conducted inside our university building. The collected data consists of discrete light readings measured by our developed Android application shown in Fig. 2 installed on the user's smartphone. The collected data values precisely vary in turn according to the conducted behavior type and the intensity of the lighting in the surrounding tested area. The collected data files were automatically stored on the smartphone for further processing.

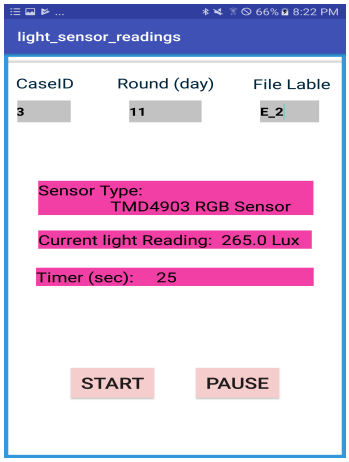


Figure 2: LightLock Android application.

Each data file includes the measurements of a single case (i.e., behavior) and contains the following data information: *Time:* the time the light signal was recorded, *Measurements:*

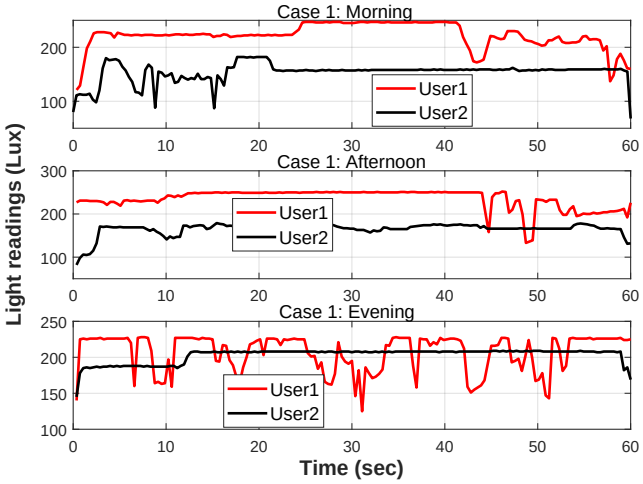


Figure 3: Illustration of the $case1$ light intensity signatures for both users at three time profiles.

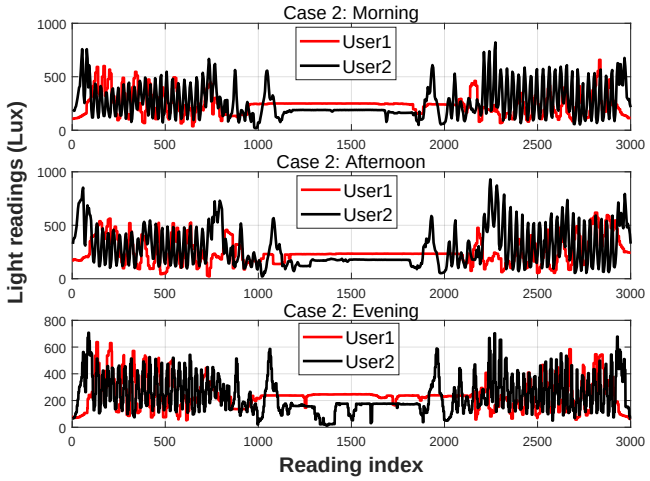


Figure 4: Illustration of the $case2$ light intensity signatures for both users at three time profiles.

the discrete values of measured light intensity, and *Label:* the value 1 or 2 that denotes the corresponding user.

2.2 Multi-model features extraction

During our experiments, we found that the discrete measurements of light intensity recorded by the smartphones are dramatically affected by indoor environment structures. Relying directly on these values results in highly inaccurate recordings over time. Unlike most previous sensor-based identification/authentication proposals (see Section 6) that rely on statistical time domain features (i.e., mean, SD, variance, Max, Min, ect.) and frequency domain features (i.e., Energy, Entropy, P1, ect.), LightLock relies on unique features extracted from four designed modules. In this second stage, we propose a multi-model system consisting of four modules that depend on extracting feature vectors (FVs) from light measurements recorded in the first stage. LightLock sequentially executes these four modules to compute light-based feature vectors: (1) Absolutes time series (ATS), (2) Auto-correlation functions (ACF), (3) Level crossing rate (LCR), and (4) Peak readings detection (PRD). After applying each module, all extracted feature vectors are stored in a database for further machine learning processing in the third stage. The first module has no data processing and relies simply on the original collected light measurements (i.e., raw data). However, the remaining three modules have their own algorithms that perform data transformations on the raw data to more adequately distinguish one user from another. All four modules will be described in detail in Section 3.

2.3 Identification using machine learning

In this stage, we construct machine learning classifiers from the feature sets of the four modules computed in the second stage to classify a user's behaviors indoors. In our work, we take advantage of a supervised machine learning technique, support vector machines (SVMs) with the RBF kernel function, to implement the identification process and compute classification accuracy. Our machine learning model is described as follows:

Features set. All extracted feature vectors (ATS, ACF, LCR, and PRD) are evaluated in three different stages according to two parameters: (1) the volume of the dataset used as input to the machine learning classifier and (2) the number of subset modules combined together in each stage. Moreover, all three stages are implemented under two evaluation setup scenarios.

SVM classifier. To implement a seamless and accurate identification system, we need to build SVM classifiers based on the four feature vectors that achieve high detection accuracy. We constructed hundreds of trained SVM classification models that cover all evaluation setup scenarios suggested in our system. Our recommended classification algorithm and details of the evaluation process are described in Section 4.

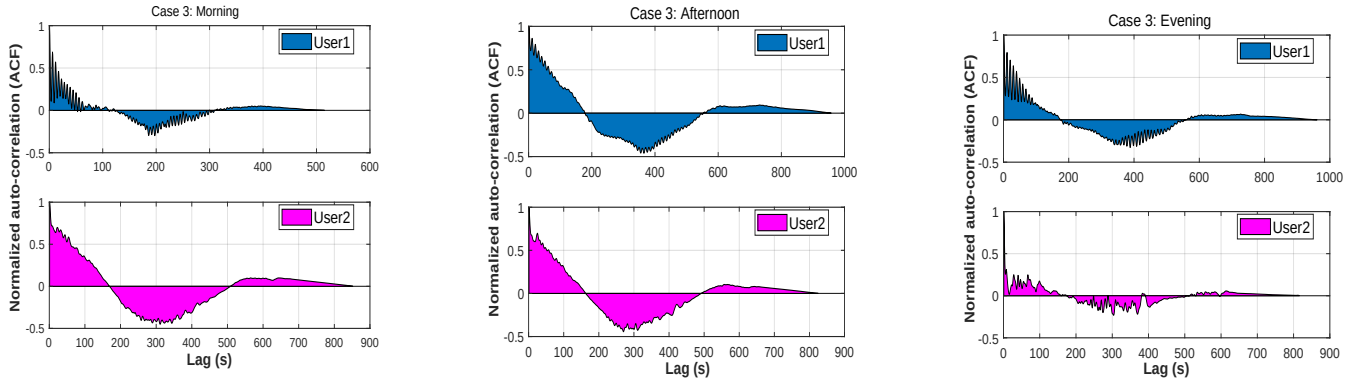
3 LightLock Multi-Model Features

In this section, we focus on analyzing and describing the four modules (ATS, ACF, LCR, PRD) in detail.

3.1 Absolute time series (ATS)

Through this module, our objective is to study the feasibility and effectiveness of relying on absolute light readings that are directly collected from the user's smartphone. We collect these absolute light readings over a series of time periods in indoor environments without using any additional optimization techniques. For use cases that involve walking, the light readings are periodic to the relative movement of various body parts as well as the indoor paths through which the subject passes.

The amplitude variations depend on several effects such as dynamic movement styles (moving and stopping frequencies), pedestrian pattern (speed, step length), the amount of shadowing the body provides while walking, and the geometry of indoor trajectories. These light signatures are typically unique from one person to another, and



(a) Morning. (b) Afternoon. (c) Evening.

Figure 5: Illustrative example that represents the differences between the auto-correlation function (ACF) signatures while conducting the $case3$ behavior at the three time profiles.

they could be leveraged as a means for user identification. Fig. 3 and 4 show a case study example of collecting light measurements of both users while they performed the same behaviors described in $case1$ and $case2$ (see Table 1) during three time period profiles (i.e., Morning, Afternoon, Evening) in the same day. From this example, we observe that the $case1$ behavior produces enough light amplitude variation that is sufficient for user classification purposes. However, the readings from $case2$ exhibit much higher similarity, and as a result, relying solely on the ATS module is feasible only in classifying users while they conduct simple behaviors; it is less successful in instances consisting of complex behaviors.

3.2 Auto-correlation function (ACF)

The auto-correlation function (ACF) is used to measure how dependent and periodic the light measurements are over time. Measuring the degree of time dependency of the measured light readings vector $R(t)$ and the shifted (lagged) copies of itself as a function of the lag is a mathematical process that produces auto-correlation coefficients scaled between -1 and +1. Coefficients near zero indicate nearly all of the intensity observations are mostly random (i.e., no correlation between successive readings over time). However, coefficients close to -1 or +1 indicate nearly all observations are repeatable (i.e., high correlation between successive readings over time). In principle, the auto-correlation process is performed by first copying $R(t)$, shifting the copy one data point to the left, and finally multiplying the shifted copy with original light vector. It is often difficult to precisely determine if data in a vector $R(t)$ is random or periodic, but auto-correlation is able to provide insight into this. For each user behavior, the real discrete light readings vector ($R(t) \in \mathbb{R}^{m\times 1}$), where m denotes vector length, is used to calculate the auto-correlation function (ACF) as follows:

$$ACF(k) = \frac{Cov(R(t), R(t-k))}{Var(R(t))}, \tag{1}$$

$$Cov(R(t), R(t-k)) = \frac{1}{m-1}\sum_{t=k+1}^{m}(R(t)-\bar{R})(R(t-k)-\bar{R}), \tag{2}$$

$$Var(R(t)) = \frac{1}{m-1}\sum_{t=1}^{m}(R(t)-\bar{R})^2. \tag{3}$$

The parameter $ACF(k)$ denotes the normalized auto-correlation as a function of the k^{th} lag value, for $(k = 1, 2, 3, \dots)$. If we have a sample vector $R(t)$, $(t = 1, 2, 3, \dots, m)$, of the collected light readings, then the k^{th} order auto-correlation $ACF(k)$ can be estimated using Eq. 1 by first calculating the covariance and variance variables from Eq. 2 and Eq. 3 respectively. $\bar{R}$ is the mean of the $R(t)$ vector and Eq. 3 is the special case of Eq. 2 in cases where $k = 0$.

To show the effectiveness of the ACF module in distinguishing user behaviors, we showcase the results of $case3$ (Table 1) as an illustrative example. We plotted the ACF output signatures in Fig. 5 for morning, afternoon, and evening time profiles for both users. The collected measurements related to any behavior are limited to a specific vector length (i.e., do not go to infinity), and we observed that the shape function of the normalized auto-correlation function decays to zero. As expected, the $ACF(k)$ value when $k = 1$ is relatively large while the next values are progressively smaller for the subsequent k lags (approximately decaying to zero). Thus, the patterns of both users shown in Fig. 5a and Fig. 5c show large differences in the normalized ACF values such that they can be accurately distinguished from one another. Nevertheless, the patterns of user 1 and user 2 in Fig. 5b have some similarity that makes the classification process difficult in some situations. Ultimately, ACF could be used as a convenient mathematical process to distinguish user behavior signatures that provides a robust way to model light intensity data for user identification.

3.3 Level crossing rate (LCR)

This module is another popular technique used to quantify the periodicity of waveforms by measuring how often the

light measurements cross certain thresholds. Thus, LCR requires focusing on both light threshold levels as well as the frequency of crossing them. In this work, we exploit the LCR module to distinguish one person's behavior from another by first setting some specific thresholds and then computing the crossing rate of the collected light measurements corresponding to each user behavior.

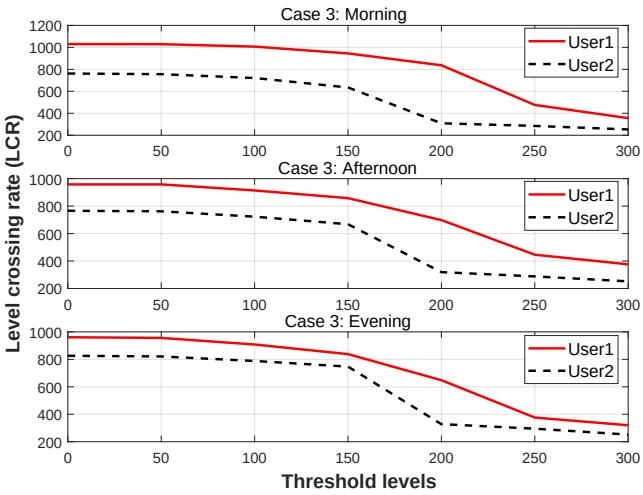


Figure 6: An example of the LCR model representation while conducting the $case3$ behavior by both users.

We implement the LCR module where its input is the light readings vector V_{light} that denotes the collected intensity measurements of a user behavior, and the output is the level cross rate vector V_{LCR} that represents the number of times the readings cross a certain threshold thr_i, $(i = 1, 2, 3, ..., q)$ of thresholds vector V_{thr}. After conducting some behavior, an LCR feature vector is extracted by comparing each reading value of the input vector V_{light} with the ith threshold value. If this reading is larger than the threshold, the crossing frequency counter (α) is incremented by one. This process is repeated to calculate the accumulative LCR value $\alpha(i)$. This process is again repeated for all chosen thresholds to obtain the final vectorized V_{LCR}. Fig. 6 shows an example of an LCR representation of $case3$ user behavior which was conducted at three different time periods. It illustrates that rates of crossing the pre-defined threshold values (50, 100, 150, 200, 250, 300) vary highly from one user to another. Thresholds are chosen empirically based on the range between the minimum and maximum values of the collected measurements by subjects. As expected, the LCR of both users are different due to the differences in the users' movement styles. Therefore, we conclude that the LCR could serve as an additional feature for differentiating users.

3.4 Peak readings detection (PRD)

The goal of the peak readings detection module is to free LightLock from relying on absolute intensity readings. PRD is employed as a means to focus solely on the peaks of light waveforms that aid in distinguishing subjects from one another as they move under or near light sources in indoor environments. The PRD module contains three main steps: (1) squaring, (2) smoothing, and (3) peak-finding; together these steps perform the whole operation of extracting peak values from the collected absolute intensities. The steps are shown in algorithm 1, and we describe each in detail below:

3.4.1 Squaring

After recording the absolute light readings, the data of each behavior is collected in a vector called V_{Light}. Then, each vector is squared to obtain $V_{Sqr} = (V_{Light})^2$. Squaring enhances large values (e.g., peak values) more than small values (e.g., noise values), thus making peak detection easier.

3.4.2 Smoothing

One of the most famous techniques for smoothing a noisy signal is the $MovingAverage$ algorithm, which calculates the average of any subset of data elements. Here, we use the moving average as a low-pass filter to process collected light data by smoothing out noisy readings of short-term fluctuations. Algorithm 1 shows how we can compute and use the moving average technique in our work and return a vector which we label V_{MAV}. This vector has the same size as input vector, V_{Sqr}, and consists of smoothed mean values, where each mean is calculated over a sliding window of size $\mathcal{W}$ across neighboring elements of V_{Sqr}.

$$V_{MAV_{(j)}} = \frac{V_{sqr(j)} + V_{sqr(j+1)} + ... + V_{sqr(\mathcal{W}+j-1)}}{\mathcal{W}}. \quad (4)$$

$$V_{MAV_{(j+1)}} = V_{MAV_{(j)}} - \frac{V_{sqr(j)}}{\mathcal{W}} + \frac{V_{sqr(W+j)}}{\mathcal{W}}. \quad (5)$$

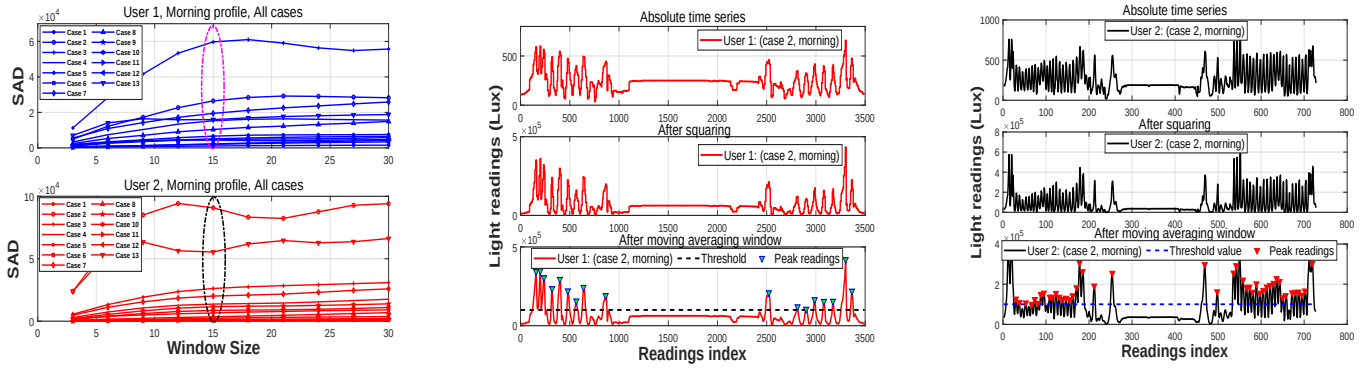
In our work, we compute the average over $\mathcal{W}$ samples of light readings by taking an equal number of samples on either side of a central value. Given a fixed size of V_{Sqr}, the first element of the moving average vector V_{MAV} is obtained by Eq. 4. Following this, the subset is modified by dropping the first number of the series and shifting the window forward to include the next value in the subset as in Eq. 5. Furthermore, in situations which lack sufficient data elements (less than $\mathcal{W}$) such as at the endpoints, the window size is automatically truncated and the average is taken over only the data elements that fill the window.

Dynamic window size: Instead of utilizing a fixed window size, we propose dynamic window size selection based on the collected dataset of each user behavior. The window size determination is based on what amount of smoothing is most appropriate. To do this, we use Eq. 6 to compute the summation of the absolute differences matrix (SAD) between the noisy squared data in V_{Sqr} and smoothed data in (V_{MAV}) at different window sizes $(\bar{w})$. SAD is computed as follows:

$$SAD_{ik} = \sum_{\forall j} \mid V_{MAV} - V_{sqr} \mid_{\mathcal{W}=\bar{w}}, \quad (6)$$

where i= (1,2,...,13) denotes the $case$ (behavior) index, k= (1 or 2) denotes user index, and j denotes the vector length of data.

An experimental example to select the best window size dynamically is illustrated in Fig. 7a where the estimated SAD values are plotted versus the range of window sizes



(a) Plot showing the effectiveness of SAD to dynamically select the best window size.

(b) Plot showing the peak-reading detection process for the $case2$ behavior of user 1 in morning.

(c) Plot showing the peak-reading detection process for the $case2$ behavior of user 2 in morning.

Figure 7: Illustrative example that represents the implementation of peak readings detection (PRD) module.

(from $0 \sim 30$) for all 13 behavior cases that were conducted by both users in the morning. The smallest window size (shown by dotted ellipses, $\mathcal{W} = 15$) where the SAD starts to flatten out is chosen. Large window sizes do not produce significantly more benefit (in smoothing) and require more intensive processing. Finally, we take the first-order derivative of V_{MAV} to determine the sign change that occurs at peak locations.

3.4.3 Peaks extraction:

Finally, the function ($FindPeaks$) takes the vector V_{MAV} as an input with three given parameters: (1) Peak threshold (P_{thr}), (2) Minimum Peak Height (MPH), and (3) Minimum Peak Distance (MPD). The peak threshold (P_{thr}) value will find peaks that are greater than both adjacent samples by the threshold ($P_{thr} \geq 0$). Minimum Peak Height (MPH) determines those peaks that are greater than the minimum peak height, (MPH ≥ 0). Minimum Peak Distance (MPD) detects peaks separated by more than the minimum peak distance, MPD. Outputs of algorithm 1 are (V_{PRD}) and ($PeaksLocations$) vectors that denote peak values and the location indices at which the peaks occur respectively. Fig. 7b and 7c show graphically the steps of implementing the PRD module and the ability to detect peaks on sample $case2$ behavior conducted by both users.

Algorithm 1: Peak Readings Detection (PRD)

Data: Light readings vector <V_{Light}>
Result: Peak readings vector <V_{PRD}>
Set: peak threshold (P_{thr}), Min peak height(MPH), Min peak distance(MPD);
Compute: <V_{sqr}> $\leftarrow$ squaring (<V_{Light}>);
Define: <V_{MAV}> moving averaging vector;
Use equations (4), (5), (6) to compute SAD matrix;
Get best dynamic window size, ($\mathcal{W}$);
for *jth value in* (V_{sqr}) *from j=1 to length* (V_{sqr}) **do**
    - Avg(j): compute moving average value using equation (4) and equation (5);
    - slide the window subset ($\mathcal{W}$) by one;
    - V_{MAV} = Vectorize [Avg(1),. . . , Avg(length(V_{sqr}))];
end
Apply the first order derivation on $< V_{MAV} >$;
<V_{PRD}> = Findpeaks($< V_{MAV} >$ | (P_{thr}, MPH, MPD));
Get [<V_{PRD}>, $PeaksLocations$].

4 LightLock Performance Evaluation

This section evaluates the performance of LightLock when implementing support vector machines (SVMs) trained on the feature vectors extracted from the four previously-described modules (ATS, ACF, LCR, PRD).

4.1 Experimental setup

We collect light-based features recorded while the users performed typical daily behaviors in their work cubicles and adjacent spaces while indoors.

Implementation: Data was collected by two participants using smartphones with developed Android application. Because different models of smarphones would yield minor variations in the collected light measurements, we deliberately use the same smartphone model (Galaxy S9 and Android version 8.0.0) for both participants. As a result, our system is only affected by user's behaviors, not hardware discrepancies. Nevertheless, LightLock can be installed on a variety of Android-based mobile phones. The participants were asked to conduct the designated 13 case behaviors within the designated testing points following identical paths and time periods. Table 1 contains 13 different daily behavior cases where each case includes a number of consecutive segments that represent specific activities (e.g. sitting, standing), phone placement and/or orientation (pocket, hand, desk), and phone facing direction (front, back). As the objective of this work is to enhance the security of indoor environments, we chose the third floor of two connected buildings of our university campus as a testing environment. The testing location points that we considered included participants' work cubicles, some commonly used public areas, and several other fixed locations as shown in Table 2. In addition, all the behaviors

Table 1: Details of the 13 cases that represent various user behaviors conducted inside our university's building.

CaseID	Script
1	Put the phone (Desk-Horizontal-Front) on the user's desk (Center) and wait for 1 minute.
2	Start from the elevator (Hand-Random-Front), move to the user's seat and put the phone (Desk-Horizontal-Front) on the user's desk (Main), wait for 1 minute, move back to the elevator (Hand-Random-Front).
3	Put the phone (Desk-Horizontal-Front) on the user's desk (Left), wait for 1 minute, stand up and move to left lounge (Hand-Random-Front), sit down and put the phone (Desk-Horizontal-Front) on the desk (Main) for 1 minute, pick up the phone (Hand-Random-Front) and move back to the user's seat, put the phone on the user's desk (Left) for 1 minute (Desk-Horizontal-Front).
4	Put the phone (Desk-Horizontal-Front) on the user's desk (Main) for 30 seconds, stand up and pick up the phone (Hand-Random-Front), move clockwise around user's seat, come back to user's seat and put the phone (Desk-Horizontal-Front) on user's desk (Main) for 30 seconds.
5	Put the phone (Desk-Horizontal-Front) on the user's desk (Left) for 1 minute, stand up and pick up the phone (Hand-Tilt-Front), move to back seat, put the phone (Desk-Horizontal-Front) on the desk for 1 minute, stand and pick up the phone (Hand-Tilt-Front), move back to the user's seat, put the phone (Desk-Horizontal-Front) on the desk (Left) for 1 minute.
6	Put the phone (Desk-Horizontal-Front) on user left end of the desk (Main) for 1 minute, shift the phone on user center end of the desk(Main) for 1 minute, shift the phone on user right of the desk(Main) for 1 minute.
7	Put the phone on the Left Lounge desk (Hand on desk-Tilt-Front) for 1 minute, stand and put the phone in the pocket, move to location (A) in the corridor, stand still for 1 minute, take the phone out from the pocket, move to the men's restroom, put the phone in the storage closet for 1 minute, move to the user's seat (Hand-Horizontal-Front), stand still for 1 minute, sit for 1 minute, put the phone (Desk-Horizontal-Front) on the desk (Main), collect for 1 minute.
8	Start at elevator and move to the user's seat with phone in pocket, put the phone (Desk-Horizontal-Front) on the user's desk (Main), wait for 1 minute, stand and move back to elevator with phone in pocket.
9	Start at Right Lounge desk with phone (Hand-Horizontal-Front), move to user's seat, put the phone on the left side of the user's desk for 30 seconds.
10	Start at Left Lounge desk with phone (Hand-Horizontal-Front), move to user's seat, put the phone on the left side of the user's desk for 30 seconds.
11	Start at Right Lounge desk with phone in the pocket, move to user's seat, put the phone on the left side of the user's desk for 30 seconds.
12	Start at Left Lounge desk with phone in the pocket, move to user's seat, put the phone on the left side of the user's desk for 30 seconds.
13	Start from the elevator (Hand-Horizontal-Front), move (fast) to user's seat and put the phone on the user's desk (Main), wait for 1 minute, move (fast) back to the elevator (Hand-Horizontal-Front).
14	Start from point A (in-front of the lab door), move (Hand-Horizontal-Front) to rest room (point B) and put the phone on the desk, wait for 1 minute, move back (Hand-Horizontal-Front) to point A.

Table 2: Details of tested location points.

Location	Divisions
User seat 1: (in room # 27317)	Desk (Main), Desk (Left), Chair
User seat 2: (in room # 26316)	Desk (Main), Desk (Left), Chair
Elevator	3rd floor Elevator
Men's restroom	Outside / Inside of the door
Left lounge	Outside / Inside of the door
Right lounge	Outside / Inside of the door

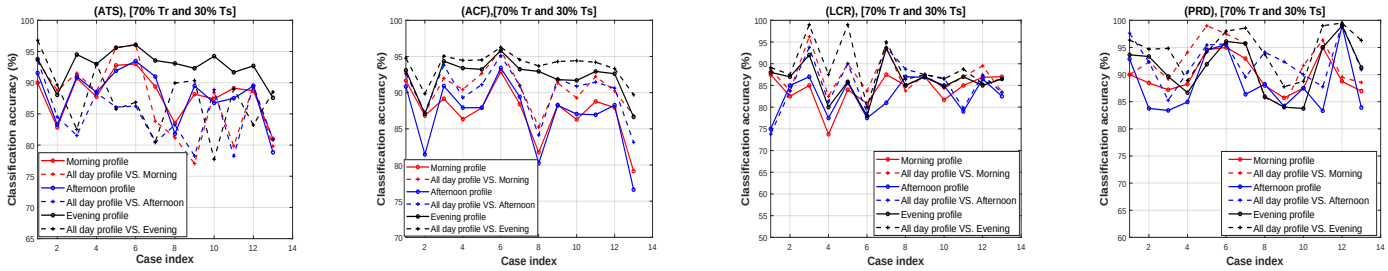
were conducted at three period profiles per day with chosen times as follows: [Morning: (9:00 $\sim$ 11:00), Afternoon: (13:00 $\sim$ 15:00), Evening: (17:00 $\sim$ 19:00)]. Moreover, to collect a dataset that is robust against light intensity fluctuations over multiple days, the participants conducted extensive experiments such that all the behaviors were repeated over 20 days (i.e., 20 rounds). All data files were stored locally on the smartphones and then exported to a computer for evaluation using support vector machines on MATLAB software. **Evaluation approach:** The evaluation process was conducted in two main scenarios: the first scenario consists of an evaluation of the three individual profiles (Morning, Afternoon, Evening) separately, and the second scenario consists of an evaluation of the $Allday$ profile versus each of the three individual profiles ($Allday$ vs Morning, $Allday$ vs Afternoon, $Allday$ vs Evening). Furthermore, we adopt both mentioned scenarios with each of the following three stages that vary according to the size of the dataset use. *First stage:* we evaluate the performance of each module individually based on the dataset that is collected during only one round (i.e., one day). *Second stage:* various subset combinations of the four modules (e.g., 2, 3, and 4) are evaluated together based on the same dataset that is collected during only one round (i.e., one day) as well. *Third stage:* all collected datasets of the 20 rounds (i.e., 20 days) are evaluated by dividing it into (20-$\mathbb{Z}$) days for training the classifiers with ($\mathbb{Z}$) remaining days used as an unseen dataset for testing the trained classifiers. We consider classification accuracy as a performance metric that indicates the ability to correctly distinguish one user's behavior from another by counting the number of test observations that are correctly classified. All of the evaluation stages are explained in detail in the following sections.

4.2 Individual modules evaluation

Here, we perform the first stage of the LightLock system evaluation process. In this stage we individually evaluate the feature vectors (FVs) that were extracted from each of the four modules (ATS, ACF, LCR, PRD) based on experiments conducted over the course of one day.

The objective of such an analysis is to investigate the effectiveness of each module under two scenarios that cover different time profiles per day. As a result, we decide which module among them produces features that are suitable for light-based indoor identification purposes. The process is started by separately collecting light measurements that correspond to the conducted user behaviors and then extracting the (FVs) by executing the modules' algorithms. Following this, the (FVs) are used to train SVMs for the classification process. Following the first scenario, each case behavior is evaluated for three time profiles: morning (M), afternoon (A), and evening (E) separately by training the SVM classifer on 70% of the feature vectors for the users.

Then, we evaluate the learned classifiers (SVM-M, SVM-A, SVM-E) with the remaining 30% of the data to compute the accuracy values. Similarly, when we follow the second scenario, each case behavior is evaluated again by training the (Allday-SVM) classifier on 70% of the feature vectors of the three profiles: morning (M), afternoon (A), and evening (E) together for the users. Then, we evaluate the learned classifiers for three time profiles $Allday$ versus morning (Allday-M), $Allday$ versus afternoon (Allday-A) and $Allday$ versus evening (Allday-E) with the remaining 30% data to compute accuracy values. We chose the dataset of round 10 (i.e., $\mathbb{Y}$ =10) to evaluate each case($\mathbb{X}$) where ($\mathbb{X}$ = 1 $\sim$ 13) under the two scenarios. During the evaluation of this stage, we created 312 SVM classifiers where each of the four modules has 78 SVM classifiers. These 78 SVMs correspond to



(a) Absolute time series (ATS) module.

(b) Auto-correlation functions (ACF) module.

(c) Level crossing rate (LCR) module.

(d) Peak-reading detection (PRD) module.

Figure 8: Performance results from a one-day dataset (round 10) for all 13 case behaviors under two scenarios for the four modules.

13 behaviors multiplied by 6 time profiles, (M, A, E, Allday-M, Allday-A, Allday-E). Our evaluation results of this first stage are as follows. Fig. 8a shows that the ATS module provides $Allday$ performance lower than individual profiles in most cases. However, the ACF module results shown in Fig. 8b provide classification accuracy that varies from (76.6 %) to (96.26 %), but $Allday$ has higher performance than the three individual profiles for all cases. The results of the LCR module shown in Fig. 8c are more stable and have a wider range from 73.75% to 99%, where the $Allday$ model provides higher performance than the individual profiles in most cases as well. Both the upper and lower boundaries of the classification accuracy are enhanced using the PRD module shown in Fig. 8d. The enhanced accuracy ranges from 83.33% to 99.5% with the $Allday$ profile achieving classification accuracy higher than individual profiles.

Table 3: Average classification accuracy of all 13 cases for the four modules at all time period profiles using only a one-day dataset (round 10).

Module	Time profile					
	Morning	Allday vs Morning	Afternoon	Allday vs Afternoon	Evening	Allday vs Evening
ATS	88.02	87.2	88.03	84.6	92.78	87.22
ACF	87.35	90.55	86.87	90.18	92.22	93.78
LCR	84.02	87.15	82.62	85.26	86.12	89.08
PRD	89.87	91.85	88.27	92.37	91.23	94.66

In order to show the ability of each module to generalize over all user behaviors, we compute the overall classification accuracy by averaging the accuracy values of all conducted 13 behaviors. This approach is more realistic because the identifying characteristics are based on observing a set of daily behaviors instead of a single behavior. Table 3 illustrates the comparative results of the overall classification accuracy provided by each module compared with other modules at each time profile. As expected, the figure shows that modules in the cases including all behaviors provide modest detection results that vary from 82.62% to 94.66% compared to the results of a single behavior which reached more than 99%, due to the different natures of the group behaviors. From the results in this section, we can draw several conclusions from our evaluation thus far: First, we found that all four modules still have the potential to provide good classification accuracy even under the condition of behavior diversity. Second, the $Allday$ profile performance is consistently higher than individual time profiles for the same given dataset (i.e., round 10). Third, the peak readings detection (PRD) module is the method that provides the highest classification results.

4.3 Evaluation of different combinations

In the *second stage*, we perform further evaluation but with different combinations of the four modules. The aim is to explore the impact of including different combinations of feature vectors (FVs) on user identification accuracy. The outputs of all possible combinations will determine which subsets are optimal in providing better performance for the security purposes. To do this, we evaluated all possible combinations of two, three and four module subsets using the same $10th$ day (i.e., round 10) raw dataset under the two scenarios. Then, we analyzed all behaviors and computed the overall classification accuracy values as done in the *first stage*.

The performance results of the *second stage* are listed in Table 4, which contains three sub-tables that record the results of the performances of two, three and four module subsets respectively. As shown in Table 4a, we evaluated six different combinations of the subsets that include two modules each by creating 468 SVM classifiers. Each modules subset shows the overall classification accuracy value aggregated from 13 behaviors under 6 time profiles. Similarly, as shown in Table 4b and 4c, we created 312 and 78 SVM classifiers for the combination of subsets that contain three and four modules respectively. We found that the two combination subsets (ACF+LCR) and (ACF+PRD) perform the best, providing the highest classification performance (above 99%) while other subsets that include the (ATS) module result in lower performance (e.g., ATS+LCR).

This is expected as the ATS module uses highly fluctuating data that directly depends on the original light readings without any processing. By evaluating combinations of three subsets consisting of three modules, the classification accuracy is further improved such as in the (ACF+LCR+PRD) subset which achieves the highest accuracy (above 99%) among this group. The evaluation of all four modules results in a drop in classification accuracy because adding the ATS module reduces the overall system performance for any combination. Finally, based on our experiment results, we decide that the (ACF+LCR), (ACF+PRD), (ACF+LCR+PRD) and (ATS+ACF+LCR+PRD) combinations are appropriate module subsets that can be used to evaluate LightLock against unseen data.

Table 4: List of overall classification accuracy results for combinations of different modules subset.

Module	Morning	Allday vs. Morning	Afternoon	Allday vs. Afternoon	Evening	Allday vs. Evening
ATS+ACF	95.98	95.41	95.99	94.93	95.52	94.56
LCR+PRD	92.62	90.63	92.72	89.88	90.68	89.39
ATS+LCR	78.78	87.90	79.57	84.06	83.65	88.38
ATS+PRD	91.87	90.55	92.51	89.79	91.84	90.27
ACF+LCR	99.78	99.84	99.74	99.85	99.82	99.89
ACF+PRD	99.83	99.95	99.84	99.94	99.89	99.95

(a) Combinations of two modules subset.

Module	Morning	Allday vs. Morning	Afternoon	Allday vs. Afternoon	Evening	Allday vs. Evening
ATS+ACF+LCR	95.80	95.23	96.16	94.97	95.75	94.75
ATS+ACF+PRD	95.75	95.22	95.72	93.86	95.27	94.28
ATS+LCR+PRD	92.26	90.62	92.38	89.80	91.10	89.38
ACF+LCR+PRD	99.43	99.79	99.48	99.81	99.61	99.86

(b) Combinations of three modules subset.

Module	Morning	Allday vs. Morning	Afternoon	Allday vs. Afternoon	Evening	Allday vs. Evening
ATS+ACF+LCR+PRD	95.80	95.23	96.16	94.97	95.75	94.75

(c) All modules.

4.4 Unseen data evaluation

In the previous two stages, we evaluated our system by dividing the dataset of the $10th$ day into 70% for training and 30% for testing. However, because our system depends on light measurements that change from one day to another and are affected by different weather conditions, our final evaluation on unseen datasets consists of readings collected over several days. Specifically, in this stage we evaluate our system on data collected over the course of 20 days using the four highest-performing module subsets obtained from the results of the *second stage*. We trained SVM classifiers on (20-$\mathbb{Z}$) days, and we used the remaining ($\mathbb{Z}$) days as unseen data for testing.

Table 5: Phase 1 performance results on unseen data when ($\mathbb{Z}$=3).

Module	Time profile					
	Morning	Allday vs Morning	Afternoon	Allday vs Afternoon	Evening	Allday vs Evening
ACF+PRD	99.79	99.81	99.84	99.86	99.56	99.78
ACF+LCR	99.14	99.15	99.32	99.37	99.32	99.45
ACF+LCR+PRD	98.91	98.92	98.45	99.2	99.08	99.2
ATS+ACF+LCR+PRD	98.02	97.98	89.78	90.67	88.66	89.57

We present two phases for evaluating the final performance of our LightLock system. In the first phase, we choose ($\mathbb{Z}$=3) where the data collected over 17 days are used to train the SVM classifiers, and the datasets of the remaining three days (day 1, day 8, day 15) are used to test the learned classifiers. In the second phase, we choose ($\mathbb{Z}$=10) where the datasets collected across 10 even-numbered days (day 2, day 4,. . . , day 20) are used to train the SVM classifiers, and the datasets of the remaining odd 10 days (day 1, day 3,. . . , day 19) are used for testing. We deliberately selected the dataset of divergent days to evaluate our system efficiency under the worst conditions.

Table 6: Phase 2 performance results on unseen data when ($\mathbb{Z}$=10).

Module	Time profile					
	Morning	Allday vs Morning	Afternoon	Allday vs Afternoon	Evening	Allday vs Evening
ACF+PDR	99.69	99.7	99.83	99.76	99.67	99.85
ACF+LCR	99.33	99.36	99.51	99.47	99.43	99.46
ACF+LCR+PRD	98.93	98.85	99.12	99.07	99.04	99.19
ATS+ACF+LCR+PRD	85.02	86.47	89.3	89.42	85.84	88.18

This evaluation approach is conducted in the same fashion as the previous two stages, and the performance results of the computed overall classification accuracies under the two phases are provided in Table 5 and Table 6, respectively. By comparing the results, we observe that the performance with $Allday$ classification accuracy is still slightly higher than individual time profiles for both phases. Finally, we found that the performance of (ACF+LCR), (ACF+PRD) and (ACF+LCR+PRD) subsets is sufficient to detect user behavior and provide high classification results (above 98%) even on the unseen dataset.

5 Lab Study for the Effects of Environmental Conditions

As shown in Table 5 and Table 6, the classification accuracy of LightLock was between 85% $\sim$ 99% using combinations of module subsets on unseen data. Those results showed that LightLock could be used for identifying some specific users with high accuracy using the smartphone's light sensor measurements in indoor environments.

However, because there are many environmental conditions (e.g., user's height and new location) in real life, it may be difficult to make a strong claim about the generality based solely on the results of the experiments in Section 4. The dataset of the experiments was collected without considering height conditions; additionally, all participants' behaviors were evaluated within pre-trained indoor locations.

To address the limitations of those experiments, we conducted a separate lab study to collect a new dataset with two additional environmental conditions: 1) recruiting participants with different heights; and 2) light measurements of previously unseen indoor locations are collected and evaluated by LightLock.

From this case study, we determined that the height of a user (i.e., the distance from the lights) as well as non-trained locations play a role in the final identification accuracy. To determine the effects of a user's height, we selected one tall male with a height of 184 cm and one short female with a height of 152.4 cm. We collected the smartphone readings when these two users were conducting case 3 and case 13 from Table 1 for the three time profiles (Morning, Afternoon, and Evening) over the course of one day. We followed the same evaluation approach explained in Section 4 under the same two scenarios (i.e., the three individual profiles and the other three $Allday$ profiles) as well as in same tested areas and location points shown in Table 2.

Similarly, after collecting the new dataset, we applied our four proposed modules' algorithms to extract the corresponding feature vectors (FVs) in order to train SVMs for the classification process. To show the ability of each module individually to cover the six different time profiles under two scenarios, we trained 48 SVM classifiers on 70%

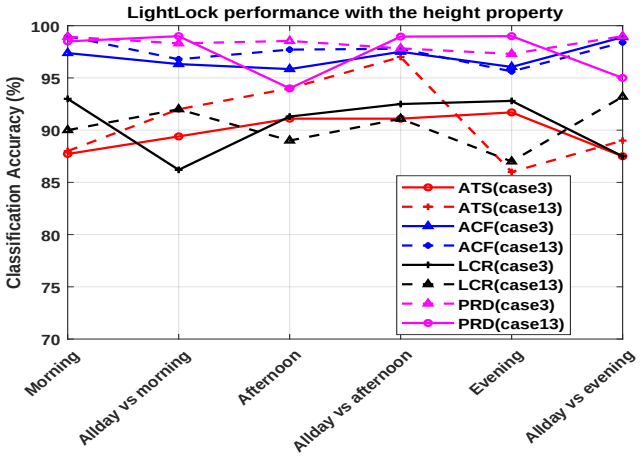


Figure 9: Illustration of the LightLock performance with including participant's height property.

of the feature vectors for the users while evaluating on the remaining 30% of the data to compute the accuracy.

Figure 9 shows the estimated classification accuracies for identifying a user when considering the height effects under case 3 and case 13 behaviors. The results range from 86% (using ATS module) to 99% (using PRD module) showing improvement in the final estimation, particularly in increasing the lower bounds of accuracy from approximately 76% to 86%, over previous results that did not consider differences in height. We also observed that the PRD module, which depends on the peak readings of the light intensity, provides the best results among the other modules. This is likely because the height of the user produces peak values that are highly affected by the distance to the lights and hence provides sufficient difference in measurements that could contribute to higher accuracy in distinguishing users.

Additionally, we aimed to investigate how the results of our approach would change if a user moved to a new location which LightLock had not previously encountered. To evaluate this, we asked participants to conduct one more behavior, case 14 in Table 1, and collected the recordings of locations that were not visited before (i.e., none of the locations listed in Table 2). We then used this dataset to test the performance of LightLock modules to identify users in these new locations. Again, we followed the same evaluation fashion as previous stages through generating another 48 SVM classifiers for the four modules.

Table 7: LightLock performance results on dataset of unseen locations.

Module	Time profile					
	Morning	Allday vs Morning	Afternoon	Allday vs Afternoon	Evening	Allday vs Evening
ATS	86.49	88.24	85.14	86.75	87.19	90.48
ACF	94.83	96.55	88.64	89.38	86	92.11
LCR	89	92.49	90.63	92.38	92.10	95.24
PRD	92.19	93.74	91.23	94.37	96.49	97

Table 7 shows the results of classification accuracy, which are computed without any pre-training process on the modules. The accuracy of LightLock in new locations is naturally slightly lower compared with previous results. In more detail, ATS module provides the lowest accuracy in such scenarios in the range of 85% $\sim$ 90%. However, the other modules such as PRD still perform well and provide accuracy up to 97%. Ultimately, we believe that our proposed LightLock system can contribute in optimizing user identification technology based on smartphones' light recordings even in previously unseen locations.

6 Related Work

Behavioral biometrics based methods belong to the "what you are" authentication/identification category [18] that aim to continuously monitor the most stable human behavior patterns during various daily activities on smartphones. In this section we categorize the related work into three types according to the technology used.

6.1 Biometrics-based identification

After a brief review of the related work of user identification on smartphones, we found that biometrics-based identification can be divided into methods based on user behaviors, touch/key, and gestures. Zheng et al. [19] leveraged the fusion of sensors such as the touch screen sensors, the gyroscope, and the accelerometer on smartphones to identify users by exploiting four types of features for capturing a user's tapping behaviors and achieved an averaged equal error rate of 3.65%. Kwapisz et al. [20] presented a behavioral biometric identification approach based on a user's movement signatures via the smartphone's accelerometer data collected from 23 users while performing daily activities. This work investigated both identifying an individual as well as authenticating a specific user. Izuta et al. [21]proposed a screen unlock method based on data collected from the accelerometer and pressure sensors. The work identified a user based on behavioral features when smartphones are taken out from the pocket, and the pressure distribution of the hand-held device during this process.

Shrestha et al. [22] presented a wave-to-access method via a hand-waving gesture recognition approach using the light sensor and then integrated the gesture with a dialing service in order to analyze user behaviors. Yang et al. [23] proposed a technique called Opensesame which addresses screen locking/unlocking for smartphones based on four features of the hand-waving to verify users. They utilized SVMs for user identification and finally achieved 15% FPR and 8% FNR. Zeng et al. [17] proposed a Wi-Fi based user identification framework called WiWho that uses channel state information (CSI) captured by Wi-Fi endpoints to identify a user's steps and walking gait. WiWho achieved an average accuracy of 80% to 92% for distinguishing a specific user from a small group of users (2$\sim$6) in a device-free manner.

6.2 Biometric authentication on smartphones

This kind of continuous authentication has been widely investigated on smartphones in two broad approaches: touchscreen-based user authentication and built-in sensor-based authentication. Xu et al. [24] addressed the issue that the smartphones cannot authenticate users during run time by developing a model that exploits four types of user touch

operations on the touchscreen and achieved classification accuracy higher than 80%. Feng et al. [25] proposed a user authentication method of continuously analyzing touch screen gestures in the context of a running application. Zou et al. [26] developed a touch-based authentication system by training a novel one-class classification algorithm import vector domain description (IVDD). Shen et al. [27] investigated the reliability and applicability of using motion-sensors to measure user's behaviors based on different gestures, habits, and angle preferences of touch actions.

Mohamed et al. [28] demonstrated an RSSI-based gait authentication process using on-body devices (e.g., smartwatches and smartphones) by extracting three channel features used to train four different classification techniques. Li et al. [29] proposed a system which authenticates a user based on continuously monitoring behavior patterns by leveraging the data of the built-in accelerometer, gyroscope, and magnetometer sensors. Li et al. [30] proposed a continuous authentication method that exploited permutation, cropping, sampling, jittering, and scaling as data augmentation approaches in addition to the accelerometer and gyroscope sensor data and achieved an error rate of 4.66%.

6.3 Light-based tracking and localization

Little work has focused on leveraging unmodified and pre-installed indoor lighting infrastructure. Zhao et al. [31] exploited generic light sources and off-the-shelf smartphones to develop an indoor localization and navigation framework to achieve sub-meter localization accuracy. Xu et al. [32] proposed an indoor localization system by leveraging data of motion sensors and light intensity values of photo-diode sensors on smartphones along walks and achieved mean location errors of 0.38, 0.42, and 0.74 meters in three buildings respectively. Ali et al. [33] developed a multi-model framework that consists of four different modules including tracking by lights to determine whether a user is in indoor or outdoor environments.

7 Conclusion

This paper presented LightLock, a machine learning approach for user identification by utilizing indoor light intensity measurements. Compared with existing methods that require user interaction with the smartphone's touchscreen or continuous sensing using a fusion of multiple energy-consuming sensors, LightLock needs only the ubiquitous and energy efficient light sensor data, and does not require additional hardware. Our experiments were conducted on 13 various daily behaviors at three time profiles over the course of 20 days inside our university buildings. A multi-model system consisting of four different modules was also proposed to extract feature vectors to evaluate hundreds of SVM classifiers implemented under three different stages of evaluation. To show the effectiveness of the proposed identification approach, LightLock was evaluated against unseen data and achieved above 98% classification accuracy to distinguish user identities by using a fusion of modules.

Acknowledgments

This work was supported in part by the NRF of Korea (NRF-2019R1C1C1007118), the ITRC program (IITP-2019-2015-0-00403), and the ICT Consilience Creative program (IITP-2019-2015-0-00742).

References

[1] Eiji Hayashi, Sauvik Das, Shahriyar Amini, Jason Hong, and Ian Oakley. Casa: context-aware scalable authentication. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, page 3. ACM, 2013.

[2] Sebastian Uellenbeck, Markus Dürmuth, Christopher Wolf, and Thorsten Holz. Quantifying the security of graphical passwords: The case of android unlock patterns. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pages 161–172, New York, NY, USA, 2013. ACM.

[3] Hyoungshick Kim and Jun Ho Huh. PIN selection policies: Are they really effective? *Computers & Security*, 31(4), 2012.

[4] Kai Cao and Anil K Jain. Learning fingerprint reconstruction: From minutiae to image. *IEEE Transactions on information forensics and security*, 10(1):104–117, 2015.

[5] Hoyeon Lee, Seungyeon Kim, and Taekyoung Kwon. Here is your fingerprint!: Actual risk versus user perception of latent fingerprints and smudges remaining on smartphones. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, pages 512–527. ACM, 2017.

[6] Abdenour Hadid, Nicholas Evans, Sébastien Marcel, and Julian Fierrez. Biometrics systems under spoofing attack: an evaluation methodology and lessons learned. *IEEE Signal Processing Magazine*, 32(5):20–30, 2015.

[7] Attaullah Buriro, Bruno Crispo, Filippo Del Frari, and Konrad Wrona. Touchstroke: Smartphone user authentication based on touch-typing biometrics. In Vittorio Murino, Enrico Puppo, Diego Sona, Marco Cristani, and Carlo Sansone, editors, *New Trends in Image Analysis and Processing – ICIAP 2015 Workshops*, pages 27–34, Cham, 2015. Springer International Publishing.

[8] 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.

[9] Lingjun Li, Xinxin Zhao, and Guoliang Xue. Unobservable re-authentication for smartphones. In *NDSS*, volume 56, pages 57–59, 2013.

[10] 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, 2013.

[11] 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. ACM, 2013.

[12] Kálmán Tornai, Terrance Boult, Niko Sunderhauf, Ethan Rudd, Lalit P Jain, Walter J Scheirer, Terrance Boult, Ajita Rattani, Walter J Scheirer, Arun Ross, et al. Gesture-based user identity verification as an open set problem for smartphones. In *IAPR International Conference On Biometrics*, volume 35, 2019.

[13] W. Lee and R. B. Lee. Multi-sensor authentication to improve smartphone security. In *2015 International Conference on Information Systems Security and Privacy (ICISSP)*, pages 1–11, Feb 2015.

[14] Z. Sitová, J. Šeděnka, Q. Yang, G. Peng, G. Zhou, P. Gasti, and K. S. Balagani. Hmog: New behavioral biometric features for continuous authentication of smartphone users. *IEEE Transactions on Information Forensics and Security*, 11(5):877–892, May 2016.

[15] Chao Shen, Tianwen Yu, Sheng Yuan, Yunpeng Li, and Xiaohong Guan. Performance analysis of motion-sensor behavior for user authentication on smartphones. *Sensors*, 16(3):345, 2016.

[16] Z. Qin, L. Hu, N. Zhang, D. Chen, K. Zhang, Z. Qin, and K. R. Choo. Learning-aided user identification using smartphone sensors for smart homes. *IEEE Internet of Things Journal*, 6(5):7760–7772, Oct 2019.

[17] Y. Zeng, P. H. Pathak, and P. Mohapatra. Wiwho: Wifi-based person identification in smart spaces. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–12, April 2016.

[18] A. Alzubaidi and J. Kalita. Authentication of smartphone users using behavioral biometrics. *IEEE Communications Surveys Tutorials*, 18(3):1998–2026, thirdquarter 2016.

[19] N. Zheng, K. Bai, H. Huang, and H. Wang. You are how you touch: User verification on smartphones via tapping behaviors. In *2014 IEEE 22nd International Conference on Network Protocols*, pages 221–232, Oct 2014.

[20] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Cell phone-based biometric identification. In *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–7, Sep. 2010.

[21] Ryo Izuta, Kazuya Murao, Tsutomu Terada, Toshiki Iso, Hiroshi Inamura, and Masahiko Tsukamoto. Screen unlocking method using behavioral characteristics when taking mobile phone from pocket. In *Proceedings of the 14th International Conference on Advances in Mobile Computing and Multi Media*, MoMM '16, pages 110–114, New York, NY, USA, 2016. ACM.

[22] Babins Shrestha, Nitesh Saxena, and Justin Harrison. Wave-to-access: Protecting sensitive mobile device services via a hand waving gesture. In Michel Abdalla, Cristina Nita-Rotaru, and Ricardo Dahab, editors, *Cryptology and Network Security*, pages 199–217, Cham, 2013. Springer International Publishing.

[23] L. Yang, Y. Guo, X. Ding, J. Han, Y. Liu, C. Wang, and C. Hu. Unlocking smart phone through handwaving biometrics. *IEEE Transactions on Mobile Computing*, 14(5):1044–1055, May 2015.

[24] Hui Xu, Yangfan Zhou, and Michael R. Lyu. Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones. In *10th Symposium On Usable Privacy and Security (SOUPS 2014)*, pages 187–198, Menlo Park, CA, 2014. USENIX Association.

[25] Tao Feng, Jun Yang, Zhixian Yan, Emmanuel Munguia Tapia, and Weidong Shi. Tips: Context-aware implicit user identification using touch screen in uncontrolled environments. In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, page 9. ACM, 2014.

[26] Bin Zou and Yantao Li. Touch-based smartphone authentication using import vector domain description. In *2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, pages 1–4. IEEE, 2018.

[27] Chao Shen, Yuanxun Li, Yufei Chen, Xiaohong Guan, and Roy A Maxion. Performance analysis of multi-motion sensor behavior for active smartphone authentication. *IEEE Transactions on Information Forensics and Security*, 13(1):48–62, 2018.

[28] M. Mohamed and M. Cheffena. Received signal strength based gait authentication. *IEEE Sensors Journal*, 18(16):6727–6734, Aug 2018.

[29] Y. Li, H. Hu, G. Zhou, and S. Deng. Sensor-based continuous authentication using cost-effective kernel ridge regression. *IEEE Access*, 6:32554–32565, 2018.

[30] Y. Li, H. Hu, and G. Zhou. Using data augmentation in continuous authentication on smartphones. *IEEE Internet of Things Journal*, 6(1):628–640, Feb 2019.

[31] Zenghua Zhao, Jiankun Wang, Xingya Zhao, Chunyi Peng, Qian Guo, and Bin Wu. Navilight: Indoor localization and navigation under arbitrary lights. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE, 2017.

[32] Qiang Xu, Rong Zheng, and Steve Hranilovic. Idyll: Indoor localization using inertial and light sensors on smartphones. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 307–318. ACM, 2015.

[33] M. Ali, T. ElBatt, and M. Youssef. Senseio: Realistic ubiquitous indoor outdoor detection system using smartphones. *IEEE Sensors Journal*, 18(9):3684–3693, May 2018.