ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

You Walk, We Authenticate: Lightweight Seamless Authentication Based on Gait in Wearable IoT Systems

[1]MR. M. RAJ KUMAR, [2] EDULAKANTI ANIL, [3] RAMAGIRI VARSHA, [4] ROHAN REDDY DOPATHI, [5] VOORADI SRINATH

[2,3,4,5] U.G. Scholor, Department of IOT, Sri Indu College of Engineering & Technology, Ibrahimpatnam, Hyderabad.

[1] Assistant Professor, Department of IOT, Sri Indu College of Engineering & Technology, Ibrahimpatnam, Hyderabad.

ABSTRACT With a plethora of wearable IoT devices available today, we can easily monitor human activities, many of which are unconscious or subconscious. Interestingly, some of these activities exhibit distinct patterns for each individual, which can provide an opportunity to extract useful features for user authentication. Among those activities, walking is one of the most rudimentary and mundane activity. Considering each individual's unique walking pattern, *gait*, which is the pattern of limb movements during locomotion, can be utilized as a biometric feature for user authentication. In this paper, we propose a lightweight seamless authentication framework based on gait (LiSA-G) that can authenticate and identify users on the widely available commercial smartwatches. Unlike the existing works, our proposed framework extracts not only the statistical features but also the human-action-related features from the collected sensor data in order to more accurately and efficiently reveal distinct patterns. Our experimental results show that our framework achieves a higher authentication accuracy (i.e., an average equal error rate (EER) of 8.2%) in comparison with the existing works while requiring fewer features and less amount of sensor data. This makes our framework more practical and rapidly deployable in the wearable IoT systems with limited computing power and energy capacity.

INDEX TERMS User authentication, gait, wearable device, Internet of Things, machine learning.

I. INTRODUCTION

The emergence of the Internet of Things (IoT) has revolutionized numerous systems and the way we interact with the computing and communication systems. On top of the predominant smartphones, there is a rapid increase in the number of wearable IoT devices such as smartwatches, smartglasses, and so on. According to Gartner Inc., we will reach the world-wide sales volume of 225 million wearable IoT devices in 2019 [1].

Although initial wearable devices were equipped with limited connectivity such as Bluetooth, the latest wearable devices are equipped with various communication modules such as WiFi and a variety of sensors. However, such multiple connectives on wearable devices can expose a variety of personal information and further increase the risk of security breaches [2] which necessitates robust security measures.

However, the attention to security aspects of wearable IoT devices has not kept pace with that to the quantitative growth. Compared to the precedent IoT devices, such as smartphones, wearable IoT devices are more prone to various security attacks due to the lack of security measures (e.g., insufficient user authentication) and limited resources (e.g., computing power and energy capacity). For example, in 2013, hackers were able to remotely infiltrate Google Glass systems to watch and hear everything wearers did [3]. In 2015, a study conducted by HP demonstrated that all smartwatches can be vulnerable to security attacks [4].

ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

Thus, to address the vulnerability of current wearable IoT security system, we consider user authentication which is one of the most principal security measures. In user authentication, one of the most commonly used methods is using passwords due to its simplicity. However, to procure a certain level of strong user authentication, users should maintain at least 19 different passwords on average for their various devices and services [5]. Thus, people often have difficulty in remembering a correct password. According to the survey conducted by Centrify at Infosecurity Europe 2015, 33% of participants in the study suffered from password rage [6].

Recently, Hanamsagar *et al.* [7] showed there is significant password re-use over multiple different online accounts. Besides, the PIN/Pattern-based authentication on smartwatches can be also prone to shoulder surfing and social engineering attacks [8]. Though regarded as more robust authentication system than passwords, conventional biometric-based authentication is also vulnerable to security breaches. Fingerprint-based authentication can be deceived by high quality fingerprint images or counterfeit fingers created using a 3D printer [9]. Even commercial iris-based authentication systems were breached by high-quality iris images [10].

To tackle the aforementioned security limitations in wearable IoT devices, we propose an authentication framework that addresses the following questions:

- How can we design an authentication framework that is both reliable and user-friendly so that users do not need to remember their passwords?
- How can we design an authentication framework that is laborious for attackers to hack or reproduce?
- How can we design an authentication framework that is easily deployable?

To address those questions, we utilize *subconscious* activities to authenticate wearable IoT device users. In interactions with users, wearable IoT devices generate various sensor data such as accelerometer or gyroscope data. And the pattern in such sensor data can be distinct as each user performs subconscious activities in their own way, which is not required to remember as well as difficult to hack or copy.

Among various subconscious activities, we consider *gait*, which is the pattern of limb movements during locomotion as it satisfies the goal of our authentication framework to provide a reliable and user-friendly authentication. Specifically, gait is shown to provide unique patterns even between people having similar physical attributes [11], and walking is one of the most rudimentary and mundane activities that requiring significant effort to copy or mimic.

In the literature, gait-based authentication frameworks can be classified into two categories based on the type of data used: 1) vision-based authentication, and 2) sensor-based authentication. The former can be spoofed by impersonating a person with a similar appearance and/or clothes [12]. Therefore, we focus on gait authentication utilizing sensors in commercial wearable devices which are easily deployable. To record the human gait sensor data, we can use various

ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

wearable devices: smartwatch, smartring, smartanklet, smartshoes, and smartbands. Among those various devices, we select a smartwatch as a wearable IoT device to record gait data since smartwatches are showing an exponential growth and expected to control nearly the half of the world market in wearable devices by 2022 [13].

In the literature, various research works [8], [14]–[19] have been proposed to authenticate wearable or hand-held device users, using subconscious activities. However, most of the prior research conducted their experiments on their own custom-made devices, where the sensors were elaborately and manually calibrated to meet their own convenience [14], [15]. In practice, however, such calibration is infeasible for most wearable IoT device users.

Even though there are some research projects conducted on commercial devices, they involve a long period of the authentication process mostly incurred by their walking detection algorithm (e.g., a user needs to walk more than 15 seconds on average for authentication) [16], [17], [19]. Consequently, they require a relatively large volume of data, which can be burdensome to wearable IoT devices with limited computing resources and energy capacity.

Therefore, to bridge the gaps in the existing works, we propose a lightweight seamless authentication framework based on gait (LiSA-G) that securely authenticates users on commercial smartwatch in a light manner. Unlike the existing works that extract only the statistical features from sensor data, LiSA-G additionally considers mechanical traits that are bound by the physical attribute of individuals. In addition, LiSA-G requires only one sampling period of sensor data which is much less than those of the existing works (i.e., 8-20 sampling periods) [14], [16], [17]. The experimental results show that our work achieves an Equal Error Rate (EER) of 8.2% on 51 participants, while using less number of features extracted from much shorter periods of sensor data. Our contributions are summarized as follows:

- Increased the authentication accuracy by using human-action-related features as well as statistical ones.
- Reduced the number of features used for authentication.
- Reduced the time required for gait authentication by eliminating the gait cycle detection.

The rest of this paper is organized as follows. In Section II, we review related works. In Section III, we propose a lightweight seamless authentication framework based on gait (LiSA-G). In Section IV, we compare our experimental results to the existing techniques. Additionally, we present the potential application scenario of our work in Section V, followed by some discussion and limitations in Section VI. We finally conclude our paper in Section VII.

II. RELATED WORK

A. GAIT AND GESTURE BASED AUTHENTICATION USING SMARTPHONES

Muaaz and Mayrhofer [17] presented a gait-based authentication method on a smartphone, where a user keeps his/her phone in his/her pocket. They developed an Android

ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

application to measure the accelerometer data from the smartphone. A pedometer sensor on a smartphone is used to detect if the person is walking or not. If the condition was true, then the accelerometer data would be recorded for 15 seconds. Both sensors were operated at 200 Hz. The collected data was further processed using linear interpolation and Savitzky-Golay filter. Gait cycles were detected using local maxima and DTW (Dynamic Time Wrapping). To check the actual performance of the system, impersonation attacks were performed. They achieved an EER of 13% with 35 volunteers.

Shahzad *et al.* [18] presented a gesture-based authentication. Here, users input their own user-specific gestures on the mobile touchscreen. They extracted some unique user features from the way passwords were inputted to devices. To record the data, they developed an Android application and Windows platform. They recorded the data of each touch point on the screen, accelerometer, and time-stamps. They found that various gestures result in different average classification accuracies. For classification, they used Support Vector Distribution Estimation (SVDE) with Radial Basis Function (RBF) kernel and achieved a false positive rate for each gesture under 5%.

Mahfouz *et al.* [19] presented a behavior based authentication for smartphones. They developed an Android application that can authenticate users based on the authentication scores which were calculated using gesture modality extracted from certain features. They recorded events related to unlocking touchscreen from 52 volunteers. For each touch, they recorded the time-stamp, coordinates, pressure, size, and action code. For evaluation purpose, they used a classification model which was trained on data from both a legitimate user and impostors and anomaly detection model which was trained on data from the legitimate user only.

B. GAIT AND GESTURE BASED AUTHENTICATION USING WEARABLES DEVICES

Johnston and Weiss [16] developed an Android application to collect the time-stamped accelerometer and gyroscope data from a commercial smartwatch. The data was recorded at 20 Hz, the process of authentication required minimum 10 seconds of data (i.e., 200 samples). They extracted the following features from the data set: average, standard deviation, average absolute difference, the time between peaks, binned distribution, and average resultant acceleration. WEKA [21] data mining suite is used for classification, and the highest identification and authentication EER achieved was 8.1%.

Cola *et al.* [14] presented a gait based authentication system that has only one user's gait pattern but no knowledge to patterns of other users, just like the commercial smartphone that stores only the fingerprints of its owner. For their manual implementation of the smartwatch, they used a Shimmer 3, an embedded TI MSP430 micro-controller (up to 24 MHz clock, 16 kB RAM, 256 kB flash), and an ST Micro LSM303DLHC accelerometer operating at 50 Hz. Walking detection was done by using a peak detection, and irregular patterns in the data were reduced by using

auto-correlation. Reduction of noise in the acceleration data along the 3 axes was done by using a Butterworth filter. For authentication, semi-supervised anomaly detection was used. For this, the Euclidean distance and the nearest neighbor analysis were used to determine the anomaly score, and the authentication accuracy achieved was 7.8% of EER.

Zhao *et al.* [8] presented a touch-based authentication scheme on a smartwatch. A user can design his/her own user interface on the screen, and hence the user can interact and enter his/her own PIN pattern. They measured authentication accuracy, authentication speed, and security level. The authentication was measured by the average accuracy rate of the input password/pattern. The authentication speed was measured by the average completion of the input, and the security level was estimated by the rate of correctly replicated passwords/patterns by shoulder surfing. They concluded that the Square PIN was most secure.

Terada *et al.* [15] used Hitachi wireless-T (ankle wearable sensor) to record the accelerometer data and angular velocity data at 100 Hz. To evaluate the data, they used a scoring system that uses the sum of differences between registered data and input data for the rolling angular velocity in the swing phase. The score and threshold decided the legitimacy of the user. An EER of 20% was obtained by using this method. Slightly different from the aforementioned studies where the smartphone was kept in the waist pocket, this work involved more movements from the legs than arms, which can give more details of gait.

C. ALTERNATIVE AUTHENTICATION USING WEARABLES

Yoon *et al.* [22] utilized the ambient light sensor for users to type PIN code in user authentication. Based on the ambient light value measured, they defined two states of the sensor: 1) Non-Zero Lux (NZL), and 2) Zero Lux (ZL). By combining the states, this work identified the user actions such as single click (NZL → ZL), double click (NZL → ZL → NZL → ZL), and hold (ZL → ZL), and utilized such actions as PIN code for user authentication. However, the experiment was conducted in the indoor environment where lighting conditions are constant. Thus, in the outdoor environment where light changes drastically, the light-sensor based authentication may not work.

Vhaduri and Poellabauer [23] designed a continuous user authentication using commercially available smartbands, and smartwatches. Using their own developed Android and iOS applications, they collected physical activity data such as step counts and physiological activity data such as heart rate, calorie burn. In the experiment, they recruited 500 volunteers for 2 years. From the collected data, they extracted 45 features for user classification. Using the Support Vector Machine (SVM), they achieved the maximum user authentication accuracy of 93%. However, one of the major bottlenecks in this work is to continuously authenticate uses in the wearable devices that have limited computation power and battery. Moreover, only statistical features were considered in user authentication.

ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

TABLE 1. Summary of limitations of existing gait and gesture based authentication models.

Device Type	Authentication Model	Limitations
Smartphone	Gait Based [17]	Require WDA. Potential performance degradation due to device orientation and position change
Smartphone	Gesture Based [18]	Prone to SSA and memorization of gestures
Smartphone	Behavioral Based [19]	Prone to SSA and memorization of gestures
Smartwatch	Touch Based [8]	Prone to SSA and social engineering attack
Smartwatch	Gait Based [16]	Require a WDA. Low accuracy
Wrist Wearable	Gait Based [14]	Non commercial device. Require WDA. Low accuracy.
Ankle Wearable	Gait Based [15]	Non commercial device. Additionally, the swing phase of a leg can change with new shoes which can potentially degrade the authentication performance. [20].

WDA : Walk Detection Algorithm, SSA : Shoulder Surfing Attack

Enamamu *et al.* [24] utilized heart rate as a biometric measure for user authentication. To record the heart rate data from MioFuse, Fitbit and Microsoft band, they developed an Android application. To decompose the heart rate signals into sub-bands, discrete wavelet transform (DWT) was used. From 4 sub-bands, they extracted 10 features from each sub-band in the feature extraction process. Using the feed forward neural network, they achieved the maximum EER of 11%. However, despite the advantage of usability, the heart rate is directly dependent on mental and physical state, which can incur false alarms.

Voit and Schneegass [25] proposed FabricID, a user authentication system based on smart textile embedded in the sleeve. They developed their own fabric that can differentiate users' hand-prints by measuring the pressure distribution, and authenticated users using the identified hand-prints. From the 1024 pressure sensor values, they extracted 71 features, and fed them into a lightweight 5NN classifier, which achieved an identification accuracy of 82.5%. Even though FabricID increases the physical space for user interaction with the system, a hand can be easily duplicated with the advancement of 3D print. Moreover, the placement of the hand on the fabric might differ each time, which can raise false alarms.

D. LIMITATIONS OF EXISTING WORKS

The research works in [14], [16], and [17] require walking detection algorithm that incurs a long period of authentication process. Besides, the studies in [14] and [15] did not use any commercial devices as commercial devices are not precisely calibrated and the sampling frequency typically differs. The studies presented in [8], [18], and [19] are still vulnerable to shoulder surfing attack. The work presented by [8] is also prone to social engineering attack. Moreover, the existing works extracts only statistical features from the sensor data, missing the distinct behavioral attributes among individuals. The device type, authentication model, and limitations of the existing works are summarized in Table 1.

III. LiSA-G

In this section, we present a light-weight seamless authentication framework based on gait (LiSA-G) for wearable IoT devices.

ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

A. SYSTEM OVERVIEW

In the conventional authentication system, a user needs to authenticate himself/herself by typing a password or provid- ing biometric measures. In other words, the existing authen- tication system requires *direct* user interaction, which may hinder seamless authentication. However, when enabled to automatically analyze users' sensor data collected in real- time via wearable IoT devices, we can authenticate users without requiring their direct or active interaction.

Aiming such seamless authentication, we propose a gait- based authentication framework for wearable IoT devices that automatically authenticates users by analyzing their sensor data. As shown in Figure 1, the workflow of our framework mainly consists of three steps: 1) data collection,
2) data preprocessing, and 3) authentication. In the data collection step, implemented in the smartwatch, a client-side LiSA-G collects and transmits the sensor data stream to the server. Then, a server-side LiSA-G runs the data preprocessing that removes undesirable glitches and noises in the received data stream as well as getting a consistent data sampling rate by applying linear interpolation. In the authentication step, the server-side LiSA-G extracts not only statistical features (e.g., mean, standard deviation) but also physical or mechanical features of arm movement (e.g., pitch, yaw, roll) from the data. Then, based on the extracted features, the server-side LiSA-G authenticates users via machine learn- ing techniques.

B. DATA COLLECTION

To collect the behavioral data from users, we developed two Android applications that operate on smartwatches and smartphones, respectively. As the smartwatches used in our experiment are only equipped with Bluetooth connectivity, we pair a smartphone and a smartwatch with each other so that the smartphone acts as a access point to relay sensor data stream collected in the smartwatch to the remote server.

For registration of users, users are asked to submit the following basic information using the developed smart-phone application: 1) name, 2) date of birth, 3) height, and
4) weight. After the submission, the smartwatch application shows a button to start recording the sensor data. On touching the button, the smartwatch starts recording data from two types of sensors available on the smartwatch: 1) accelerom- eter, and 2) gyroscope, both of which operate at 100 Hz.

ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

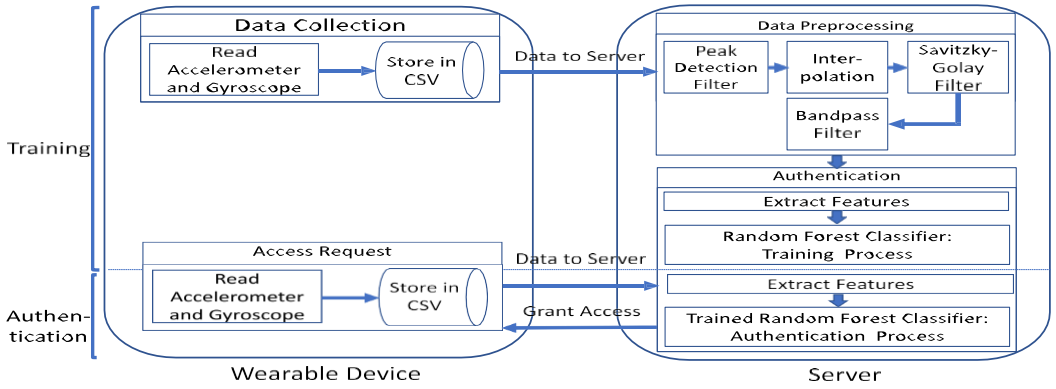
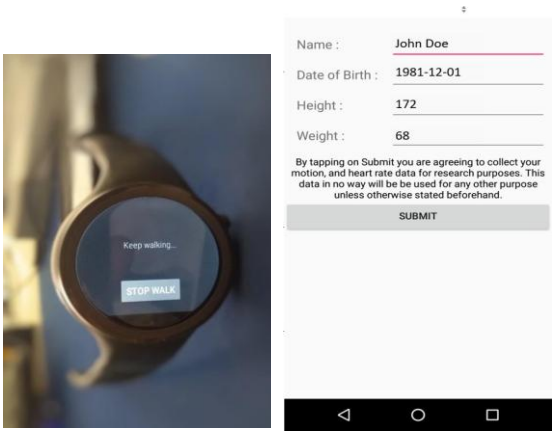


FIGURE 1. System overview.



(a) (b)

FIGURE 2. Android applications for data collection. (a) Smartwatch application. (b) Smartphone application.

Note that our system does not conduct any additional calibration beforehand to reflect the practical situation where users do not manually calibrate their smartphone or smartwatches. For data recording, each user walks approximately 500 meters in his/her normal gait. The sensor data is stored in the smartwatch in comma separated values (CSV) files and forwarded to its paired smartphone using Bluetooth. Then, the data is transported from the smartphone to the authentication server. Figure 2 shows the graphical user interfaces (GUI) of the Android application used for data collection.

C. DATA PREPROCESSING

Since raw (uncalibrated) sensor data collected from smartwatches may contain undesired noise, the raw sensor data is cleaned and refined in the data preprocessing step. Before

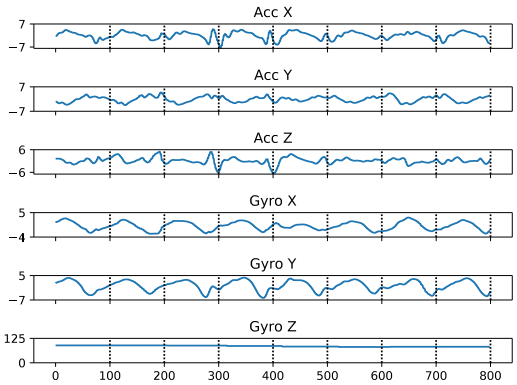


FIGURE 3. Sensor data collected from an user.

running the preprocessing, we first visualize the raw sensor data to gain an insight to analyze the data. As shown in Figure 3, we observed that the data patterns repeat in approximately 100 samples which can be interpreted as an arm swing. However, some of the data sequences show abnormal values. Thus, we removed or mitigated such fallacious values as presented below.

1) SPIKE REMOVAL

As we kept the sensors uncalibrated, there were some unexpected values skyrocketing in the raw sensor data as shown in Figure 4a. We call these values *spikes* which could degrade the authentication performance. Note that the spike values in the figure range from $-25\text{km}/s^2$ to $5\text{km}/s^2$ which are unrealistic. Thus, to eliminate such undesirable spikes, we use a spike removal algorithm that runs as follow.

ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

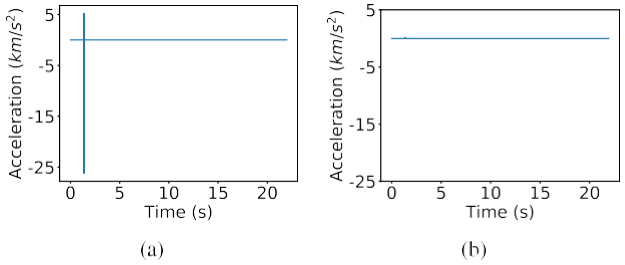


FIGURE 4. Effect of spike removal. (a) Before spike removal. (b) After spike removal.

1) Detect all the peaks in a given data sequence.
2) Compute the average of the peaks and remove the peaks that are λ times higher than the average.

Though the idea of this algorithm is quite simple, it detects and removes unusual spikes well. As shown in Figure 4b, spikes are effectively removed when $\lambda = 5$. Depending on the rigidity of the spike removal process, λ can be adjusted.

2) INTERPOLATION

As mentioned in the data collection process, the sensors in the smartwatch operate at 100 Hz. However, the Android API does not output sensor values at fixed time intervals. This is due to the mechanism of *OnSensorChanged* method in Android systems that outputs a sensor value only when the value differs from its previous value [26]. As a result, even at 100 Hz sampling rate, the sensor may not generate 100 samples in one second. That is, the data from the sensor may not be at equidistant time-intervals. In order to remedy such inconsistency, we apply linear interpolation to approximate the correlation between a time sequence and its corresponding sensor values as

$$A^r = A_0 + \frac{(A_1 - A_0)(t^r - t_0)}{(t_1 - t_0)}, \quad (1)$$

where A_0 and A_1 denote previous and current sensor data values, respectively. t_0 and t_1 are the time values for A_0 and A_1, respectively. t^r denotes the time value or series between t_0 and t_1.

3) NOISE REMOVAL

In addition to spikes, sensor noises are inevitably generated during the recording of sensor data. The main sources of noise is the electronic noise generated from the circuitry that converts the motion into an electric signal and the mechanical noise from the sensor itself. Thus, to remove such random noises, we apply the Savitzky-Golay smoothing filter (also known as the least square smoothing filter) [27]. Specifically, it applies the least square fit to a high degree polynomial

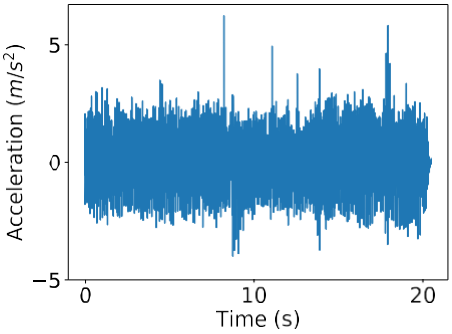


FIGURE 5. Accelerometer data after noise removal.

TABLE 2. Notations and symbols.

Notation	Description
Acc_x	accelerometer data sequence on x axis
Acc_y	accelerometer data sequence on y axis
Acc_z	accelerometer data sequence on z axis
G_x	gyroscope data sequence on x axis
G_y	gyroscope data sequence on y axis
G_z	gyroscope data sequence on z axis
$\bar{x}$	mean of data sequence x
$\tilde{x}$	median of data sequence x
σ_x	standard deviation of data sequence x
γ_x	skewness of data sequence x
K_x	kurtosis of data sequence x
$\rho_{(a,b)}$	correlation between two data sequences, a and b

concentrated their gaze on the smartwatch while walking in the beginning. After some brief moment, they started walking normally. Similarly, they consciously gazed on the smartwatch at the end of their walk. When analyzing the sensor data sequences, we observed that most of such noises are distributed at both ends of the data sequence. Hence, we apply a band-pass filter to the data in order to remove the noises at both ends. Figure 5 shows the refined data after the noise with an odd size window while the data point is at the center. We choose the least square smoothing filter as it preserves the original features of a waveform (e.g., relative maxima, minima, and width), while effectively reducing the noise. Moreover, we remove the noise incurred from the human factor. During our experiments, some volunteers consciously

ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

removal process.

D. USER AUTHENTICATION

In the user authentication process, LiSA-G first extracts a set of features from the preprocessed data. In the existing research, various statistical features have been considered in the feature extraction process. Cola *et al.* [14] considered interquartile range (IQR), kurtosis, max, min, mean, mean crossing rate (MCR), median, median absolute deviation (MAD), peak to peak amplitude, root mean square (RMS), skewness, and standard deviation. On the other hand, Johnston and Weiss [16] used average absolute difference (AAD), binned distribution, and average resultant acceleration (ARA) on top of mean, standard deviation, peak to peak amplitude as features.

However, with an increase in the number of participants, the accuracy of user authentication can degrade when only the statistical features are taken into account. Therefore, we additionally consider physical or mechanical features which can better represent each user's unique attributes in physical

ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

movement. Concurrently aiming to authenticate users in a lightweight manner, we empirically select the following fea- tures that well represent characteristics of each user's walking motion: mean, standard deviation, skewness, kurtosis, corre- lation, pitch, roll, yaw, and force. To extract the aforemen- tioned features, we divide each user's whole data sequence into smaller data sequences with equal sample window size. We quantify one smaller data sequence as a period of the whole data sequence. As shown in Figure 3, we can observe the data sequence periodically repeats approximately every

100 data samples. In the following, we present each feature and briefly explain each definition and interpretation.

1) MEAN ($\bar{x}$)

We calculate mean values for both accelerometer and gyro- scope on each axis (x, y, and z), e.g., $\bar{Acc}_x$ and $\bar{G}_x$. Those mean values represent each user's general arm swing accel-
eration and wrist rotation.

2) STANDARD DEVIATION (σ_X)

We calculate standard deviation values for both accelerometer and gyroscope on each axis (x, y, and z), e.g., σ_{AccX} and σ_{GX}. The standard deviation represents the consistency in arm swing. For instance, a smaller σ_x can be interpreted as a more constant arm swing pattern.

3) SKEWNESS (γ_X)

In mathematics, skewness measures the asymmetry of data distribution about its mean. With a negative skewness, the mass of the data distribution is concentrated on the right side of the data distribution. On the other hand, a positive
skewness means that the mass of the data distribution is
concentrated on the left of the data distribution. In our system,

it indicates when the main acceleration and rotation occur in each period of the data (one arm swing) between early

5) CORRELATION ($\rho(a,b)$)

As mentioned earlier, we use two kinds of sensors to analyze users' gait: 1) accelerometer, and 2) gyroscope. As both sen- sors are coupled in terms of the sensor dynamics, we calculate the correlation between both sensors. Concretely, there can be correlations between the sensors about their x, y, and z axes. Hence, we extract 9 correlation features in total. The correlation between two data sequences, namely a and b, is calculated as follows:

$$\rho(a,b) = \frac{\Sigma(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\Sigma(a_i - \bar{a})^2}\sqrt{\Sigma(b_i - \bar{b})^2}} \quad (4)$$

where $a \in \{Acc_x, Acc_y, Acc_z\}$ and $b \in \{G_x, G_y, G_z\}$.

6) PITCH, ROLL, AND YAW

In mathematics, the Euler angles describe the orientation of a rigid body in 3D space using 3 elemental rotation axes:
1) yaw (vertical), 2) pitch (transverse), and 3) roll (longi- tudinal). To understand the mechanism of the Euler angles, imagine an aircraft in flight. The yaw rotation is the rotational movement of the aircraft nose to left or right about the yaw axis running up and down. The pitch rotation is the rotational movement of the aircraft nose to up or down about the pitch axis running from a wing to the other wing. The roll rotation is the rotational movement about the roll axis running from nose to tail. Similarly, we can consider a smartwatch worn on each user's wrist as an aircraft controlled by the arm swinging motion and the wrist rotation. By including the Euler angles, we can better capture each user's behavioral characteristics. We calculated each user's pitch, roll, and yaw values as follows:

$$yaw = \frac{180}{\pi} \times \arctan\left(\frac{Acc_z}{\sqrt{(Acc_x)^2 + (Acc_z)^2}}\right), \quad (5)$$

$$pitch = \frac{180}{\pi} \times \arctan\left(\frac{Acc_y}{\sqrt{(Acc_x)^2 + (Acc_z)^2}}\right), \quad (6)$$

ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

phase and latter phase. For instance, one period of data with a negative skewness indicates that its main activity happens in the early stage. The skewness is calculated as follows:

$$\gamma_x = \frac{3(\bar{x} - \tilde{x})}{\sigma_x}, \quad (2$$

where $\tilde{x}$ is median of the data sequence.

4) KURTOSIS (K_X)

In mathematics, kurtosis measures the sharpness of the peak in data distribution. In other words, it measures how densely data is concentrated on the center of the data distribution. In general, data distribution with a higher kurtosis shows a sharper and taller peak. In our system, kurtosis indicates the intensity and swiftness of each user's arm swing (accelera- tion or rotation): the higher kurtosis, the more intense and swift an arm swing is. We calculate kurtosis as follows:

$$K_x = \frac{\sum_n (x_i - \bar{x})^4}{\sigma_x^4}, \quad (3)$$

where n is the number of data samples.
from

$$roll = \frac{\pi}{180} \times \arctan\left(\frac{Acc_x}{\sqrt{(Acc_y)^2 + (Acc_z)^2}}\right), \quad (7)$$

where Acc_x, Acc_y, and Acc_z represent the accelerometer data along x, y, z axis, respectively.

7) FORCE (N)

As shown in the GUI of our smartphone application, we col- lected each user's basic information including weight during the data collection. Depending on each user's mass, the force applied in arm swing while walking can differ. The force is calculated as follows:

$$Force(N) = \sqrt{Acc_x^2 + Acc_y^2 + Acc_z^2} \times m, \quad (8)$$

where m is the mass of a user. Note that we assume that each user's arm weight (mass) would be positively correlated with his/her overall body weight.

As a result of the feature extraction process, we have 24 statistical features by calculating mean, standard deviation, skewness, and kurtosis on x, y, z-axis respectively

ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

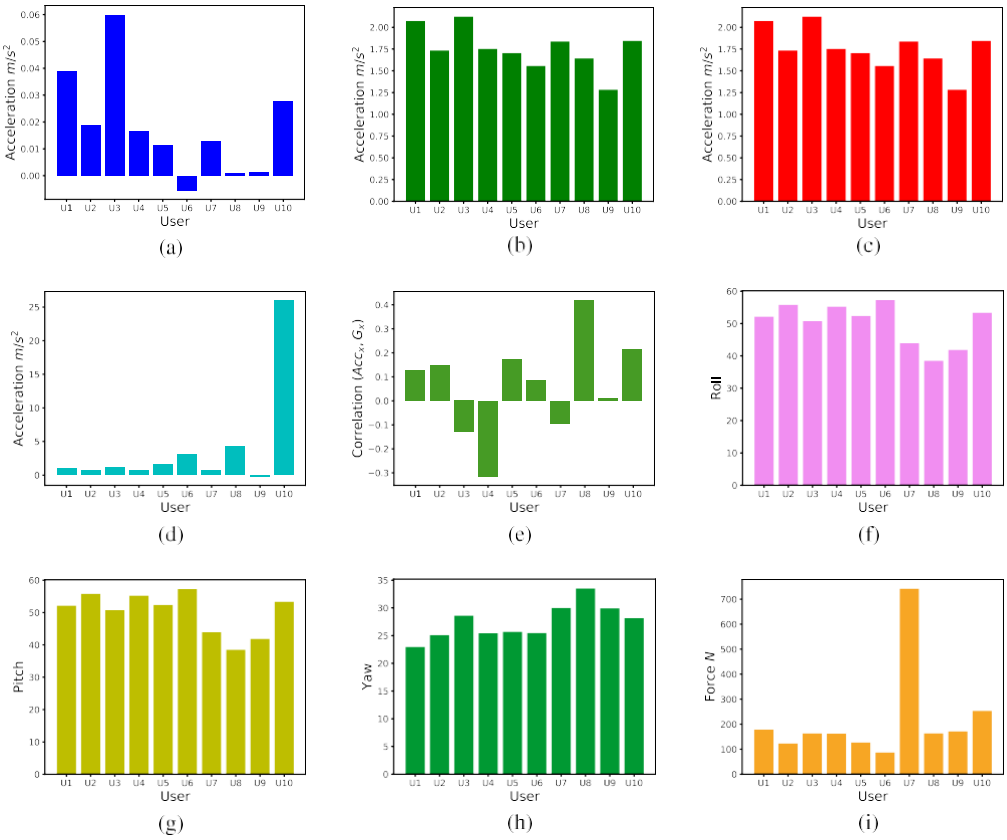


FIGURE 6. Features extracted from the data. (a) Mean ($\overline{Acc_X}$). (b) Standard Deviation (σ_{Acc_X}). (c) Skew (γ_{Acc_X}). (d) Kurtosis (K_{Acc_X}). (e) Correlation ($\rho_{(Acc_X,G_X)}$). (f) Roll. (g) Pitch. (h) Yaw. (i) Force.

the accelerometer and the gyroscope. Additionally, we have 13 behavioral features: 9 correlations, yaw, pitch, roll, and force. In total, we extract 37 features from one period of data sequence for each user. Consequently, a feature matrix with 37 columns is constructed. Here, the number of rows in the feature matrix varies depending on the number of periods in each user's data sequence. Accordingly, a class label vector is constructed whose size is the same as the number of rows in the feature matrix. In the vector, all the elements indicate an id of the user corresponding to the feature matrix. The distribu- tions of features are shown in Figure 6. Note that we show the distribution of 9 selected features for the first 10 users (i.e., U1-U10) due to the readability: 1) mean, standard deviation, skewness, kurtosis of accelerometer on x-axis, 2) roll, pitch, yaw, force, and 3) correlation between accelerometer and gyroscope on x-axis. In the figure, the difference among users are evidently shown.

After the feature extraction process, LiSA-G authenticates users by applying *supervised* machine learning algorithm to the extracted feature dataset. Note that the authentica- tion system in LiSA-G is not a simple binary classification (i.e., whether a user is one of the legitimate users or not), but a *multi-class* classification (i.e., whether a user is the corresponding user or not). Given the feature matrix and the label vector, we train the supervised machine learning

algorithm to minimize the classification error. Once the train- ing is completed, we can predict the class of test feature matrix.

IV. EXPERIMENT

In this section, we provide the experiment results and evaluate the performance of LiSA-G.

A. EXPERIMENT SETTING

In the experiment, we used Motorola 360 Sport 2nd Gen (smartwatch) and Motorola G4 plus (smartphone) to collect the sensor data from 51 volunteers. Considering the number of volunteers, the number of classes for

ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

classification is 51. Since the choice of classifier influences the classification result, we first tested three well-known machine learning classifiers: 1) random forest classifier, 2) K-nearest neighbors classifier, 3) and multi-layer perceptron. Based on our empir- ical study where the random forest classifier worked best for our dataset, we chose to use the random forest classifier. In addition, our empirical study showed that the classifier showed higher weights to correlation, yaw, pitch, roll features which characterize the actual physical traits of each user, as well as standard deviation.

Our experiment consists of two phases: 1) training phase and 2) testing phase. Before creating two separate phases,

TABLE 3. Confusion matrix.

	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10
U1	**61**	0	0	0	1	0	0	0	0	0
U2	0	**39**	0	0	0	0	0	0	0	0
U3	0	0	**29**	0	0	0	0	0	0	0
U4	0	0	1	**54**	0	1	0	0	1	0
U5	0	0	0	0	**42**	0	0	0	0	0
U6	0	0	1	0	1	**41**	0	0	0	0
U7	0	0	0	1	1	0	**50**	0	0	0
U8								**35**		
U9	0	0	0	1	0	0	0	0	**50**	0
U10	0	0	0	0	0	0	0	0	0	**48**

we first combine all the feature matrices from 51 volunteers into a feature matrix by concatenating them along the row axis maintaining the number of columns as 37. Then, the feature matrix is divided into a training feature matrix (by randomly shuffling and slicing 70% of the entire feature matrix) and a test feature matrix (by selecting the remaining 30% of the entire feature matrix), while dividing the class label vector into a training label vector and a test label vector accordingly. In the training phase, LiSA-G trains the classifier with the training feature matrix and the training label vector. In the testing phase, LiSA-G predicts the classes of the test feature matrix using the trained classifier.

B. PERFORMANCE EVALUATION

With the test label vector and the prediction result, a confusion matrix of order (m, n) is created, where $m = n$ and m is the number of classes. For better readability, the confusion matrix shown in Table 3 contains only the first 10 users. Using the confusion matrix, we can calculate the values for True Positive (TP_i), True Negative (TN_i), False Positive (FP_i), and False Negative (FN_i) for user i. In the confusion matrix, TP_i is where both row and column index are i in the diagonal elements marked in boldface. TN_i can be calculated by summing all the diagonal elements in bold font except TP for user i. FP_i can be calculated by adding all the elements in i-th column except for the diagonal element in the column. FN_i is calculated by adding all the elements for i-th row except for the diagonal element in the row. Based on these values, the authentication accuracy for user i is calculated as

$$accuracy_i = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}. \quad (9)$$

For example, TP_6, TN_6, FP_6, and FN_6 in Table 3 can be calculated as follows: $TP_6 = 41$, $TN_6 = 61 + 39 + 29 + 54 + 42 + 50 + 35 + 50 + 48 = 408$, $FP_6 = 1$, $FN_6 = 2$. As a result, the authentication accuracy for user 6 in Table 3 is calculated to be 99.34%.

Considering all the participants in the experiment, an EER of 8.2% is achieved on average. In Table 4, we compare our work to the existing works [14], [16] in terms of the average EER, the number of features used, and the number of participants in the experiment. Note that in this comparison, each work is evaluated on its own dataset. Considering the

ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

TABLE 4. Performance comparison on own dataset.

Approach	# of Features	EER	# of Participants
GAIT [14]	54	2.9%	15
Smartwatch [16]	40	16%	59
LiSA-G (Proposed)	37	8.2%	51

TABLE 5. Performance comparison on our dataset.

Approach	# of Features	EER
GAIT [14]	54	38.23%
Smartwatch [16]	40	14.81%
LiSA-G (Proposed)	37	8.2%

TABLE 6. Effect of period size on EER.

Period Size	EER
100	8.2%
200	9.2%
300	8.7%
400	9.05%
500	11.02%
1000	13.9%

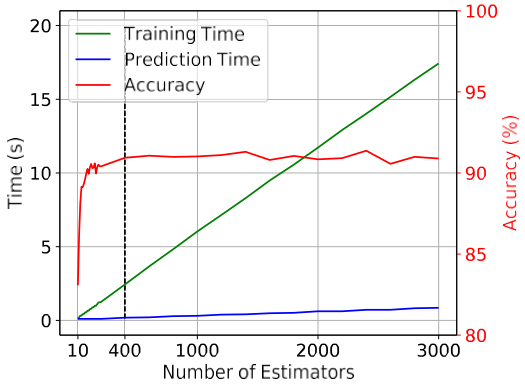


FIGURE 7. Effect of number of estimators using random forest classifier.

number of participants in the experiment, our work achieves superior authentication accuracy to [16]. Moreover, our work shows a comparable performance to [14] notwithstanding its larger scale of system (more than three times), while using less number of features. For an unbiased comparison, we additionally implemented the other methods and tested their performance on our dataset. Table 5 shows that our method can achieve the lowest EER using the least number of features. Furthermore, LiSA-G requires only one period of data (i.e. it requires only one period of sensor data, 100 data samples in our case) from each user for authentication and identification in the system, whereas 8-20 periods are needed in other works. Consequently, it reduces the volume of data and the time required for authentication.

In addition, we varied the period size to analyze its effect on the authentication performance of our approach. As shown in Table 6, the authentication performance generally degrades as the period size increases. We can trace the source of such performance degradation to 2 main changes: 1) reduction of

ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

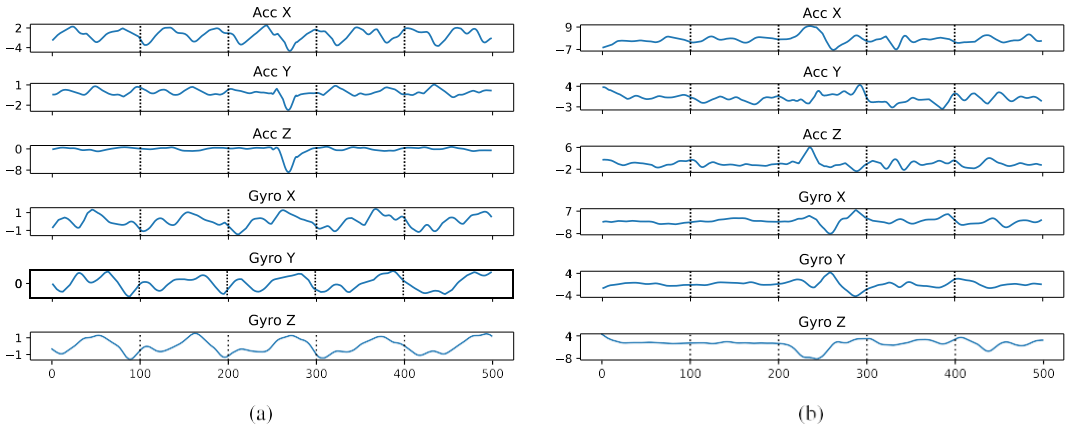


FIGURE 8. Failure cases. (a) Inconsistency in Acc_Y and Acc_Y. (b) Inconsistency in Acc_X, Acc_Y, $Gyro_X$, $Gyro_Y$, and $Gyro_Z$.

feature matrix, and 2) increased generality in each feature. Interestingly, this result shows that setting the period size to the data size sampled in 1 second (100 Hz in smartwatches) achieves the best authentication accuracy.

To test the feasibility of LiSA-G in practical deployment, we show the computation time for LiSA-G when implemented using Python 3.5 in Ubuntu 16.04 LTS 64 bit equipped with Intel Xeon(R) CPU E5-1650 v4 @ 3.60GHz x 12, NVIDIA TITAN Xp, and 32GB RAM. In the test, we range the number of estimators in the random forest classifier from 10 to 3000 in an increment of 10 since the number of estimators affects the computation time. In Figure 7, we show the training time, prediction time, and accuracy of LiSA-G. Specifically, the training time represents the computation time required to train the classifier given all the training data sequences, while the prediction time represents the computation time required to predict or authenticate a user given a data period. Despite the linear increase in both training and prediction time in proportion to the number of estimators, LiSA-G completes its training process in significantly short period of time regardless of the number of estimators. Moreover, LiSA-G predicts or authenticates a user almost in real time (less than a second).

While the computation time shows the linearity, the accuracy of LiSA-G shows drastic increase at the certain number of estimators (approximately 400), and remains almost constant. Considering the relatively static accuracy after the drastic increase, we can estimate the minimum computation time for LiSA-G to achieve reliable user authentication results. In our test, LiSA-G achieved 91.8% of accuracy with 400 estimators, which required approximately 3 seconds of training time. Overall, such prompt responsiveness of LiSA-G, while achieving reliable authentication accuracy, can enable to authenticate users in real time, incurring no authentication delay. Moreover, considering the specification of the machine in our test (only one desktop or even Raspberry-pi), LiSA-G can be deployed in cost-effective way, without requiring additional devices.

ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

C. AUTHENTICATION FAILURE CASE ANALYSIS

As mentioned in the performance evaluation, LiSA-G achieves an EER of 8.2%. Although LiSA-G shows high accuracy of authentication, there exist some authentication failure cases. Here, we present a few failure cases in Figure 8 and discuss the underlying causes of them. To compare the failure cases to the success (accurate authentication) cases, we visualize the failure cases located at the center of the figure, i.e., the data sequence between 200 and 300.

In LiSA-G, the essence of authentication process is to learn and utilize the distinct and consistent pattern in the data sequences from each user, concurrently considering all the 6 types of data sequences: $Acc_{x,y,z}$ and $G_{x,y,z}$. However, in terms of the consistency in data patterns, failure cases show the most inconsistent data patterns over all. In Figure 8a, accelerometer data on both x and y axes (i.e Acc_x and Acc_y) in the failure case shows different patterns from those in the accurate authentication cases. Figure 8b shows more inconsistent data patterns across gyroscope and accelerometer data on x and y axes.

The underlying causes of such failure cases can be attributed to: 1) sensor fault, and 2) temporarily abrupt or inconsistent physical movement. Considering the imperfection of sensor hardware, there can exist sensor fault values incurred from various processes while recording sensor values. Even though we conducted the data preprocessing to remove the undesired fault or noise values, there can still exist fault values. On top of the sensor fault, users' temporar- ily inconsistent physical movement in the data collection pro- cess can incur the failure cases. Since we requested each user to walk in the realistic environment where some people but him/her were present concurrently, some interaction between the users and the others could happen, which could incur temporarily inconsistent physical movement.

V. APPLICATION SCENARIO

To better illustrate the potential of our framework, we suppose two application scenarios where a user wearing a smartwatch

ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

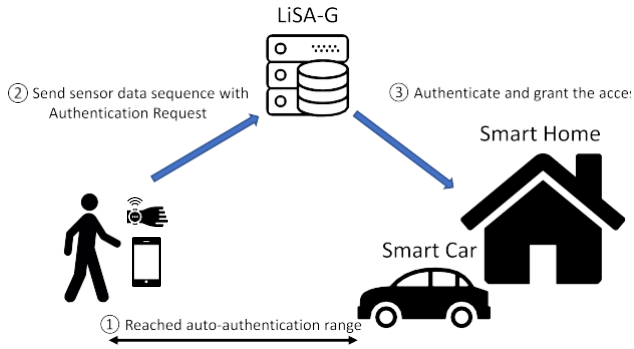


FIGURE 9. Application scenario of LiSA-G.

is approaching: 1) a smart home system, and 2) a smart vehicle system as shown in Figure 9. In the first scenario, whenever the user approaches the smart home system equipped with our framework within a certain range, the smartwatch automatically collects and transmits the stream of sensor data to the system. Once the stream of sensor data is received at the smart home system, the framework authenticates the user by analyzing and comparing the pattern in the received data to the stored pattern for the authentic user. Based on the authentication result, our framework can automatically grant each user's access to the system. Similarly, whenever the user approaches the smart vehicle within a certain range, the smartwatch transmits the stream of sensor data to the smart vehicle equipped with our framework. Then, the framework automatically authenticates the user, and decides his/her access to the vehicle. Moreover, LiSA-G can be effective authentication framework for people with senile dementia or disability to memorize or type password, especially in the indoor environment settings like hospitals, special care homes.

VI. DISCUSSION AND LIMITATION

Although our authentication framework shows promising results, there are still limitations and some technical hurdles to overcome in order to make our work fully operational in the real world. In this section, we discuss some issues.

1) Attacker Model: LiSA-G has not considered any attack models for gait authentication. To deploy our work in real work, We first need to prove how robust it is to impersonation attack where an attacker tries to deceive the authentication system by mimicking a legitimate user's gait.
2) Walking condition: At present, LiSA-G only considers a normal walking pattern. However, walking patterns can vary depending on environment, body condition, emotion, etc. Specifically, the arm swing while walking is used to maintain the center of gravity and angular momentum. Under various circumstances such as carrying heavy weights or having an injury, the arm swing can be hindered [20], which can affect the current gait authentication performance. Thus, multiple walking profiles for each user are required to authenticate each user in varying walking conditions. Also, we need

to identify how gait authentication is affected under different walking conditions.

3) When to initiate gait authentication: As IoT devices have limited battery, continuous authentication is not desirable. One of the plausible solutions is to use location information (GPS or IP and MAC addresses of the authenticating entity) to trigger the authentication process. For example, when a person is walking to smart home, the authentication system in the home will send the data packet that contains its MAC address to trigger the authentication process only when the person and the smart home are in proximity.
4) How to support a much larger number of users: Currently, we can support 51 users in the system. However, when the number of users in the system increases, the accuracy can decrease. To enable the authentication system in larger scale and satisfactory accuracy, we consider incorporating adaptable feature selections according to the number of users in the system, and applying deep neural network to utilize hidden features in the authentication process.

VII. CONCLUSION

Walking pattern, e.g., arm swings in a walk can be utilized to effectively authenticate users. In this work, we proposed a new gait based authentication framework, LiSA-G, that is reliable, user-friendly, and easily deployable. Our framework can classify users with a higher accuracy (91.8% success rate) than other existing works while using less number of features by extracting a new combination of features that are related to the human behavioral traits. The proposed framework is user- friendly as it capitalizes on users' mundane activities, and easily deployable as commercially available smartwatches are used. Considering its application to the IoT ecosystem with limited resources, the proposed framework is designed lightweight by eliminating the gait cycle detection process and using much less amount of data. We expect that the pro- posed framework can help to provide seamless authentication and can be readily integrated with other systems to provide multi-factor authentication.

ACKNOWLEDGMENT

(Pratik Musale and Duin Baek contributed equally to this work.)

REFERENCES

[1] S. Draper. (Dec. 2018). *Wearable Device Sales Will Grow 26 Percent Worldwide in 2019, Says Research Company Gartner*. Accessed: Feb. 1, 2019. [Online]. Available: https://www.wearable-technologies.com/2018/12/wearable-device-sales-will-grow-26-percent-worldwide-in-2019-says-research-company-gartner/

[2] T. Micro. (Mar. 2018). *Are your Wearables Fit to Secure You? Researchers Outline 3 Attack Surfaces*. [Online]. Available: https://www.trendmicro.com/vinfo/ph/security/news/internet-of-things/are-your-wearables-fit/-to-secure-you-researchers-outline-3-attack-surfaces

[3] M. Prigg. (May 2013). *Google Glass Hacked to Transmit Everything You See and Hear: Experts Warn 'the Only Thing it Doesn't Know are Your Thoughts*. [Online]. Available: http://www.dailymail.co.uk/sciencetech/article-2318217/Google-Glass-HACKED-transmit- hear–experts-warn-thing-doesnt-know-thoughts.html

[4] K. Rawlinson. (Jul. 2015). *Hp Study Reveals Smartwatches Vulnerable to Attack*. [Online]. Available: http://www8.hp.com/us/en/hp-news/press-release.html?id=2037386

[5] S. Faris. (Jul. 2016). *Do You Suffer From Password Rage?*. [Online]. Available: http://theweek.com/articles/637588/suffer-from-password-rage

[6] J. Chatzky. (May 2017). *Password Rage, it's a Thing*. [Online]. Available: https://lifelockunlocked.com/tips/password-rage-thing/

[7] A. Hanamsagar, S. S. Woo, C. Kanich, and J. Mirkovic, ''Leveraging semantic transformation to investigate password habits and their causes,'' in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2018, p. 570.

[8] Y. Zhao, Z. Qiu, Y. Yang, W. Li, and M. Fan, ''An empirical study of touch-based authentication methods on smartwatches,'' in *Proc. ACM Int. Symp. Wearable Comput. (ISWC)*, New York, NY, USA, 2017, pp. 122–125. [Online]. Available: http://doi.acm.org.proxy.library.stonybrook.edu/10.1145/3123021.3123049

[9] J. Myerson. (Mar. 2017). *How to Fool a Fingerprint Sensor*. [Online]. Available: https://www.electronicproducts.com/Mobile/Devices/How_to_fool_a_fingerprint_sensor.aspx

[10] S. Khandelwal. (Mar. 2015). *Hacker Finds a Simple Way to Fool Iris Biometric Security Systems*. [Online]. Available: https://thehackernews.com/2015/03/iris-biometric-security-bypass.html

[11] D. Gafurov, E. Snekkenes, and P. Bours, ''Spoof attacks on gait authentication system,'' *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 491–502, Sep. 2007.

[12] A. Hadid, M. Ghahramani, V. Kellokumpu, M. Pietikäinen, J. Bustard, and M. Nixon, ''Can gait biometrics be spoofed?'' in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 3280–3283.

[13] (Sep. 2018). *It's the Era of the Smartwatch: Idc Says Device to Rule Nearly Half of Wearables by 2022*. Accessed: Mar. 1, 2018. [Online]. Available: https://economictimes.indiatimes.com/magazines/panache/its-the-era-of-the-smartwatch-idc-says-device-to-rule-nearly-half-of-wearables-by-2022/articleshow/65810524.cms

[14] G. Cola, M. Avvenuti, F. Musso, and A. Vecchio, ''Gait-based authentication using a wrist-worn device,'' in *Proc. 13th Int. Conf. Mobile Ubiquitous Syst., Comput., Netw. Services (MOBIQUITOUS)*, New York, NY, USA, Nov. 2016, pp. 208–217.

[15] S. Terada, Y. Enomoto, D. Hanawa, and K. Oguchi, ''Performance of gait authentication using an acceleration sensor,'' in *Proc. 34th Int. Conf. Telecommun. Signal Process. (TSP)*, Aug. 2011, pp. 34–36.

[16] A. H. Johnston and G. M. Weiss, ''Smartwatch-based biometric gait recognition,'' in *Proc. IEEE 7th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Sep. 2015, pp. 1–6.

[17] M. Muaaz and R. Mayrhofer, ''Smartphone-based gait recognition: From authentication to imitation,'' *IEEE Trans. Mobile Comput.*, vol. 16, no. 11, pp. 3209–3221, Nov. 2017.

[18] M. Shahzad, A. X. Liu, and A. Samuel, ''Secure unlocking of mobile touch screen devices by simple gestures: You can see it but you can not do it,'' in *Proc. 19th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, New York, NY, USA, 2013, pp. 39–50. [Online]. Available: http://doi.acm.org.proxy.library.stonybrook.edu/10.1145/2500423.2500434

[19] A. Mahfouz, T. M. Mahmoud, and A. S. Eldin, ''Poster: A behavioral biometric authentication framework on smartphones,'' in *Proc. ACM Asia Conf. Comput. Commun. Secur. (ASIA CCS)*, New York, NY, USA, Apr. 2017, pp. 923–925. [Online]. Available: http://doi.acm.org.proxy.library.stonybrook.edu/10.1145/3052973.3055160

[20] S. M. Bruijn, O. G. Meijer, P. J. Beek, and J. H. van Dieën, ''The effects of arm swing on human gait stability,'' *J. Exp. Biol.*, vol. 213, no. 23, pp. 3945–3952, 2010. [Online]. Available: http://jeb.biologists.org/content/213/23/3945

[21] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, ''The WEKA data mining software: An update,'' *ACM SIGKDD Explorations Newslett.*, vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278

[22] H. Yoon, S.-H. Park, and K.-T. Lee, ''Exploiting ambient light sensor for authentication on wearable devices,'' in *Proc. 4th Int. Conf. Cyber Secur., Cyber Warfare, Digit. Forensic (CyberSec)*, Oct. 2015, pp. 95–100.

[23] S. Vhaduri and C. Poellabauer, ''Wearable device user authentication using physiological and behavioral metrics,'' in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6.

[24] T. S. Enamamu, N. Clarke, P. Haskell-Dowland, and F. Li, ''Transparent authentication: Utilising heart rate for user authentication,'' in *Proc. 12th Int. Conf. Internet Technol. Secured Trans. (ICITST)*, Dec. 2017, pp. 283–289.

[25] A. Voit and S. Schneegass, ''FabricID: Using smart textiles to access wearable devices,'' in *Proc. 16th Int. Conf. Mobile Ubiquitous Multimedia (MUM)* New York, NY, USA, 2017, pp. 379–385. [Online]. Available: http://doi.acm.org/10.1145/3152832.3156622

[26] A. Developer. (Jun. 2018). *Sensoreventlistener*. Accessed: Jun. 6, 2018. [Online]. Available: https://developer.android.com/reference/android/hardware/SensorEventListener

[27] A. Savitzky and M. J. E. Golay, ''Smoothing and differentiation of data by simplified least squares procedures,'' *Anal. Chem.*, vol. 36, no. 8, pp. 1627–1639, 1964.