IDENTIFYING SMARTPHONE USERS BASED ON SMARTWATCH DATA

BY SIDHIKA VARSHNEY

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

DR. JANNE LINDQVIST

and approved by

New Brunswick, New Jersey

January, 2017

ABSTRACT OF THE THESIS

IDENTIFYING SMARTPHONE USERS BASED ON SMARTWATCH DATA

by SIDHIKA VARSHNEY

Thesis Director: DR. JANNE LINDQVIST

In recent years, smartphones have become part and parcel of peoples life. Smartphones are used for all day-to-day critical tasks like money transfer, storing important documents and other information. This thesis presents, a user identification system based on smartwatch data. For identification of a user, walking activity and call receiving activity are analyzed when the phone is on the table and in pocket or bag. The recorded data from four smartwatch sensors enables the calculation of mean, variance, skewness, and gamma distribution parameters. These features are used to train the model. The presented system was tested on 20 participants and has an Equal Error Rate (EER) of 0.052.

Acknowledgements

I would like to thank my adviser,Dr. Janne Lindqvist for guiding me through my thesis. I am grateful for his advice and for investing so much with me. Additionally, he helped me understand the concepts and made me acquainted with the customs of academic life and research.

I would like to thank my past and present colleagues, Can Liu, Xianyi Gao and other members of the HCI lab. I thank the Department of Electrical and Computer Engineering for supporting and offering diverse courses which helped me to form the basics of higher level concepts. It was a privilege being part of this Department. My heartiest thanks to its faculty, staff and students. I would also like to say thank you to all of the volunteers who helped me in completing this research.

Finally, I would like to thank my parents, family and my friends for all of their love, encouragement and support. I would also like to thank Rishabh and Neha for their support through out my thesis.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In today's time, the most common methods for authenticating a user on a mobile or desktop system requires a password at the entry point. These methods have both usability and security flaws. If the user inputs the correct password, then only he or she is granted access to the device. From a usability perspective, users have to focus on each step of authentication every time they interact with their devices. These steps cause more inconvenience in the case of mobile devices because their usage period is short. Having to enter a PIN or Password every time is too cumbersome and time consuming for small activities like checking e-mails, or surfing on social media. This is why users generally keep weak and short passwords, thus resulting in the weak authentication system. For these conventional methods, once the entry point authentication is passed correctly the device is not capable of finding the intruders. This research deals in developing an identification system which is based on behavioral biometrics for securing mobile devices.

Biometrics is the science of establishing the identity of an individual on the basis of physiological or behavioral attributes of the person. It is an automated method of identifying an individual because biometric authentication techniques are done completely by machine [1]. Biometric identifiers are distinctive to each individual rather than the token id number, password, and number for verifying or recognizing the identity of an individual [2]. Examples of physiological characteristics, which are currently used for automatic identification, include retina, finger vein patterns, iris features, voice or face patterns, biological characteristics, and user keystroke-which is a way of determining how fast a user types, and the amount of time they spend pressing each key.

A Biometric system should fulfill the following properties:

1. Universality: Every person should have the biometric identifier which is being in the system.

2. Uniqueness: No two persons should have the same biometric identifiers.

3. Permanence: The Biometric identifier in the system should be invariant with time.

4. Collectability: Biometric characteristics must be measurable quantitatively and easy to acquire.

5. Performance: The Biometric technique accuracy level.

6. Acceptability: The level of user acceptance of the biometric system.

7. Circumvention: The level of difficulty in order to forge an identification/ authentication [3].

Smart cards having fingerprint, face, and hand geometry, are used as access devices in centralized systems; fingerprints and signature are taken as a mandatory requirement for a driver license [4]; visa on passports has both a hand measurement and photograph [5]; season pass for amusement parks is linked to the finger prints of the purchaser together with a unique barcode [6]; home automation system are linked by automatic voice recognition systems [7]; and delivery of confidential files through iris recognition and finger print matching [8]. All of these seem completely different in terms of purpose, procedures, and technologies, but each system uses biometric authentication in one way or another.

1.1 Motivation

The motivation of the thesis is to develop a novel system based on a smartwatch and android device which can be used for the identification of a user. Nowadays smartwatches have functionality that is enhanced beyond timekeeping. These are full-edged digital tools. Smartwatches are capable of collecting information from internal and external sensors in addition to retrieving data from devices like desktop and laptop computers.

Smartwatches have sensors like heartrate sensor, temperature sensor and other that could be used to collect information of a user which are unobservable. Thus, making applications like gesture-based passwords and devices more secure while authenticating the user.

1.2 Contribution

In this study, an identification system using a smartwatch has been developed. Every person has a unique style of walking and receiving a call. Therefore, for the purpose of identification only, call receiving and walking activities are considered. The smartwatch under test has 12 software plus hardware sensors available out of which only 4 sensors are used for this study. First, we developed an application for data collection. This is followed by a feature extraction in order to train the classifier. The Support Vector Machine (SVM) classifier, with different types of kernels is used for the identification purpose. The system is tested with data of 20 participants and has achieved an Equal Error Rate (EER) as low as 0.052 for the SVM with Gaussian kernel.

1.3 Outline

In Chapter 2, previous work related to this field is described. Chapter 3 has detailed information about the hardware used in this research. Chapter 4 elaborates the design of whole identification system and implementation for data collection. In Chapter 5 and 6, the methods and results of data analysis are discussed respectively. In Chapter 7 results achieved in this research are discussed. Discussions and conclusion are in chapter 8 and 9 respectively.

Chapter 2

Related Work

This chapter discusses the related work in the field of behavioral biometrics. Due to the rapid growth of cities by the mid-1800s along with the industrial revolution, there was a formal need to recognize human physiology. In 1858, the first systematic capture of hand images, for the purpose of identification was recorded by Sir William Herschel. He recorded handprints on the back of a contract for each worker to distinguish employees from others who might come and claim to be the worker when pay day arrived [9]. H. Gamboa et al. presented a verification technique based on the behavioral biometrics. In this research, they verified the user from the human computer interaction through a pointing device, typically a mouse pointer. The interaction of the user and pointing device is analyzed for extracting behavioral information which was used to authenticate the user [3].

Other than the PIN-based, geometry of hand and patterns, gestures based authentication also gained importance in this field. In 2003 Rekimoto et al. introduced a technique based on synchronization tapping of digital devices to make a network connection. In SyncTap if a user wants to connect two devices, he or she has to press and release the connection button on both devices synchronously. The system works by comparing the recorded timings locally on both the devices to authenticate. This system can also be used for making secure connections [10]. Another application based on synchronous gestures is for connecting multiple devices bumping into each other for interaction [11]. In Lester et al. a method based on coherence function was presented. The walking data is recorded using a low-cost micro electro-mechanical systems (MEMS) accelerometer used for detecting whether the two devices are associated with the same person or not. The system is 100% accurate when comparing 8 seconds of data using a sliding window and tolerant to inter-device communication latencies and requires little communication bandwidth [12].

Many techniques have been promoted recently as alternative forms of authentication systems. In 2007 M. Shahin presented a system based on the hand vein authentication system utilizing the fast-spatial correlation of hand vein patterns. The results verified that there is no matching between the left and right hands vein pattern of the same person. Therefore, the probability that two people could have the same vein pattern is very low. Their system operated at a 97% genuine acceptance rate and a 99.98% genuine reject rate [13]. In 2012, M. Frank et al. investigated whether a classifier can continuously authenticate users based on the way they interact with the touch screen of their phone. They proposed a set of 30 behavioral touch features that can be extracted from raw touch screen logs and demonstrated that different users populated distinct subspaces of these features. The experimental results of the proposed system failed as a standalone authentication mechanism for long-term authentication [14].

L. Ballard et al. took a step to develop methodologies which considered the threat models that have been widely ignored. The research also presented a generative attack model based on concatenative synthesis that can provide a rapid indication of security afforded by the system [15]. In 2012, N. Sae-Bae et al. presented a novel multi-touch gesture based authentication system, recording gestures from all five fingers which achieved accuracy of 90% [16]. However, the system is prone to shoulder surfing, finger oil traces and potentially provides significantly large entropy.

Liu et al. presented a recognition algorithm named as uWave using a single three axis accelerometer. This system required a single gesture pattern for training and achieved 98.6% accuracy [17]. Mayrhofer et al. presented two concrete methods for authentication, ShaVe and ShaCk. In these methods, sensing and analysis of shaking movements were combined with the cryptographic protocol for secure authentication. ShaVe (Shacking for Verification) use accelerometer data to verify whether the key agreement has taken place with the intended devices by comparing the similar movements. ShaCk (Shake to Construct a key) use accelerometer data for the extraction of feature vectors for the construction purpose. ShaVe technique is more secure than

ShaCk. However, ShaCk is computationally less expensive [18]. Wu et al. presented an acceleration based recognition approach known as Frame-based Descriptor and multi-class SVM. This research represented acceleration data as a frame based descriptor for the extraction of discriminative information. A SVM-based classifier used for the recognition purpose resulted in 98.93% for 4 gestures and 89.29% for 12 gestures for the user-independent case; whereas, for the user-dependent case, the recognition rate is 99.38% for the 4 gestures and 95.21% for all of the 12 gestures [19]. In further research to improve the results, the team developed an efficient adaptive update method. They used a minimum route determination algorithm in DP matching which achieved an equal error rate of 4.0%. This system employs dynamic time warping and affinity propagation algorithms for training and for recognition purpose it utilizes the sparse nature of gesture sequence by implementing compressive sensing for gesture recognition [20].

In 2012, researchers from Carnegie Mellon University showed that accelerometer readings could be used to extract entire sequences of entered text on a smartphone touch screen keyboard [21]. They trained their system only on the acceleration measurements of a security sensitive task of password entry. Researchers also presented that accelerometer readings can be used to infer location and named the system as ACComplice [22]. In this system, it is demonstrated that the device owner could be located within 200 meters radius using accelerometer reading. Fujinami et al. proposed a system for recognizing the storing position of the phone on the body as a context of device and user. The system employed 3 axis readings of the accelerometer for recognizing 9 positions using Machine learning algorithms with 60 features. The result of the offline experiment showed an overall accuracy of 74.6% in a strict condition of Leave-One-Subject-Out test [23].

Li et al. proposed a novel system for continuous authentication. The system uses a classifier for learning owners finger movement patterns and matches the current users finger pattern without hindering the working of any other application. In this way, the system keeps re-authenticating the true owner of the smartphone [24]. In Tian et al. a unique authentication method was proposed named as KinWrite. It authenticates using handwriting, in which a user can write a password in a space provided instead of typing it. Writing in space adds behavioral biometric characteristics like handwriting and personalized passwords, which are difficult to replicate by another user. In KinWrite, Kinect was utilized to capture the handwriting. Experimental results showed that with 100% precision, 70% accuracy or a 99% accuracy could be achieved [25].

In 2012, J. Guerra-Casanova and his team members proposed an innovative biometric technique for mobile devices based on hand gesture movements in the air. A sequencing algorithm is implemented to correct the hand gesture movements because it is not possible to repeat the same gesture movement every time in the air. User authentication involves user enrollment and verification. For enrollment, it is important for the user to perform the same 3-D hand gesture several times to create a biometric template. The system is only utilizing the 3 axis readings of the accelerometer. The research team performed a complete evaluation of this technique by analyzing 100 user's databases including real attempts at falsification [26]. Another similar work was proposed by Okumura and his team. They proposed a method that authenticates the owner using acceleration signal from an embedded accelerometer in the mobile device. The user has to shake the device to unlock it, and the system verifies the owners acceleration signal by using a DP matching algorithm which can adapt fluctuations caused by different grips [27]. R. Murmuria et al. proposed a continuous authentication system based on the power consumption, touch gestures, and physical movement. They were one of the first research groups to propose power consumption as one of the authentication systems in Android devices. They employ the fact that different users perform different tasks on various applications while modeling user behavior. They also employed an anomaly detection algorithm for each model and placed a limit on a fraction of anomalous activities which can be considered as normal for any given users [28].

Mantyla et al. built a box including several types of sensors attached to the mobile devices in order to automatically deduce what is the state of the phone when the user is using it. They divided the gestures into two categories: static and dynamic. Static is when the user keeps the phone on one ear and talks to someone: whereas, dynamic is when a person is moving the phone from one ear to another while talking. For phone gesture recognition purposes, they used Hidden Markov Model (HMM) and the Self-Organizing Map of Kohonen [29]. Feng et al. designed a continuous mobile authentication system, FAST. It works on the fact that all users have different touch features such as pressure and speed and acceleration of finger movements. FAST has a unique feature, it collects the sensor data without disrupting the working of other applications. It first detects whether the user is the owner of the device or not. If it determines that user is different, post authentication techniques are performed [30]. In 2014, Michael Sherman and his team presented a free form of gesture password for authentication. In this they showed that signatures and angular shapes are the best remembered passwords [31]. S. Li et al. implemented a user authentication system, Headbanger for head-worn devices by monitoring the head movement in response to an external audio stimulus. They showed that it is robust against imitation attacks, accurate and light weight. The processing latency on Google Glass was around 1.9 seconds [32].

Chapter 3

Hardware

This chapter describes in detail smartwatch, smartphone and technology used for this project.

3.1 Smartwatch: Motorola Moto 360 (1st Generation)

Wristwatches have evolved significantly over the last half-century from the introduction of the first digital watches in the 1960s to what we now term "smartwatches"—fully programmable watches containing active electronics mostly in the form of sensors and with the capability to interact with other devices. The first smartwatch was introduced in 2000 by IBM, which demonstrated a prototype watch running Linux and powered by an ARM processor [33].

(a) Moto 360 watch front view.

(b) Moto 360 watch back view.

Figure 3.1: Moto 360 with a front and a back view.

The Motorola Moto 360 watch has a design that is more reminiscent of a regular watch with a circular design, supporting capacitive touch display with a Atmel MXT112S Capacitive Touchscreen Controller. The watch is 46mm in diameter and

11.5mm thickness paired with a leather band. It is encased in stainless steel and glass, and the straps are made of leather sourced from the Chicago tannery [34].

It weighs 49 grams and has a screen size of 1.56 inches across with a resolution of 320 x 290 that results in a pixel density of 205 pixels per inch. It has only one physical button for activating the display of the watch. There is a black slice at the bottom of the screen in which an ambient light sensor has been placed [35]. Moto 360 is the first Android Wear smartwatch to feature inductive charging. It charges wirelessly via Qi magnetic induction and a Qi standard charger; it can simply be charged using any Qi wireless charger. It has two microphones, Wolfson Microelectronics WM7121 at the Top and a Wolfson Microelectronics WM7132 MEMS at the bottom Port Analogue Silicon Microphone, and a Bluetooth 4.0 and Wi-Fi for connecting to other devices. The Moto 360 watch has the Android Wear operating system. The system on chip is the Texas Instruments X3630ACBP OMAP 3 (Open Multimedia Application Platform) application process and 1 GHz Cortex A8 CPU. It has a fixed point digital signal processor and a 4GB (512 MB) Mobile LPDDR. It also has a 3.8 V, 300 mAh battery rated at 1.1 Wh of energy [36].

Altogether there are 12 hardware and software sensors, listed using the listing function in Android. Accelerometer, step counter, light sensor, and gyroscope for example are few sensors out of the 12. For this project, readings of following sensors have been recorded for the purposes of our analysis:

- **Accelerometer** is used for measuring the acceleration of force in metre/s^2 in all the three physical axes (x, y, and z) including the force of gravity that is applied to the device. It is a hardware type of sensor which is commonly used in motion detection applications.

- **Light Sensor** is used for measuring illumination lux (lx). It is hardware type of sensor used for controlling screen brightness.

- **Magnetic field Sensor** is used for measuring the ambient geomagnetic field for the three physical axes (x, y, z) in microT. It is also a hardware type of sensor used for creating a compass.

- **Gravity Sensor** is used for measuring the force of gravity in metre/s^2 that is applied to a device on all three physical axes (x, y, z). It can be a software or a hardware type of sensor used in motion detection applications.

- **Linear Acceleration Sensor** is used for measuring the acceleration force in all of the three physical axes excluding the force of gravity. Linear acceleration is equal to the subtraction of force of gravity from the accelerometer sensor's readings. It is the type of hardware and software sensor used for monitoring acceleration along a single axis.

- **Rotation Vector Sensor** is used for measuring the three elements of the device's rotation vector resulting in orientation of a device. It can be a software or hardware type of sensor which is used in detecting motion and rotation [37].

3.2 Why Smartwatch?

Today, smartwatches have become a daily use electronic device, they connect to a smartphone through a Bluetooth link and rely on them for updating applications and transferring data to the internet. We chose a smartwatch over a smartphone for our experiments because a smartwatch is worn on the wrist which helps in receiving highly accurate data of their hand movements from various sensors. In addition to the sensors offered by smartphones, a smartwatch has additional sensors like heart rate and gesture sensor. Since heart rate and hand gesture are unique for every person, this can be utilized for identification purposes.

To summarize we discussed the hardware of smartwatch, sensors we are using in this study and why we selected smartwatch data for identifying the users.

Chapter 4

Design And Implementation

This chapter describes the flowchart of the identification system. The chapter starts with the flowchart explaining all steps taken which led us to our final results. The second section of this chapter, explains the implementation and working of applications used for the data collection.

4.1 Flowchart

Figure 4.1 shows the flowchart of the data flow system from the collection step to the identification step. The first block is the Android Watch which is used in this study to collect the data. The users were asked to wear this Android watch and perform activities discussed in section 5.1.2.

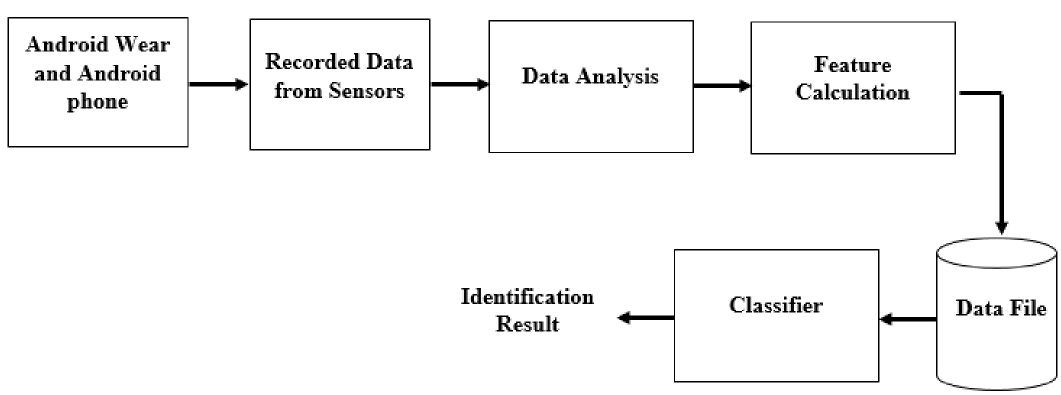


Figure 4.1: Flow chart of the identification system.

The second block is the data collection step from four sensors gravity sensor, accelerometer, linear accelerometer and rotation vector sensors in x, y and z directions. The collected data is processed and saved in separate files. This data is used to calculate the magnitude and features which are employed to train the classifier for identification. This process will be elaborated in the next section. The last block is the classifier which will be trained by the training data obtained from the feature calculation block and tested by testing data. The classifier identifies the user and returns the maximum probability of the matching class. The probability of all users is then used to calculate the equal error rate (EER).

4.2 Apparatus

For data collection, an Android application was developed for our phone and Android Wear (Moto 360 watch) using Android Studio. The android application for the smart watch is a non-launching application displaying a notification if the application is running in the background. Android wear connects to the phone via Bluetooth.

The mobile application had three functional buttons:

- **START:** This button sends a message to the Android Wear to enable the sensors and keep recording the data from these sensors onto the watch in JSON format.
- **STOP:** This button sends a message for disabling the sensors and hence stops the recording.
- **EXPORT:** This button sends a message for exporting the sensor's data from the watch to the phone.

In Android Wear application, a receiver is made which performs operation upon receiving the following messages from the phone:

- **Start Message:** When this message is received, Sensor Service starts which enables the four Sensors: gravity sensor, accelerometer, linear accelerometer and rotation vector sensors.
- **Stop Message:** When this message is received, Sensor service is stopped, resulting in disabling all four of the sensors which were enabled when a start message is received.

- **Export Message:** When this message is received, Android Wear starts sending the data to the phone. It stops the application once the data transfer is complete.

4.2.1 Flowchart of Data Collection

Figure 4.2 shows the flowchart explaining the message flow and actions performed when the phone is the sender and the Android watch is a receiver. As explained in Section 4.2, the phone application has three buttons. When any button is pressed, a message is sent to the Android watch via Bluetooth using Message API. When watch receives any message, it checks whether the message is “START”,“STOP” or “EXPORT.” Application in the Android watch runs in the background, displaying a notification when data collection starts.

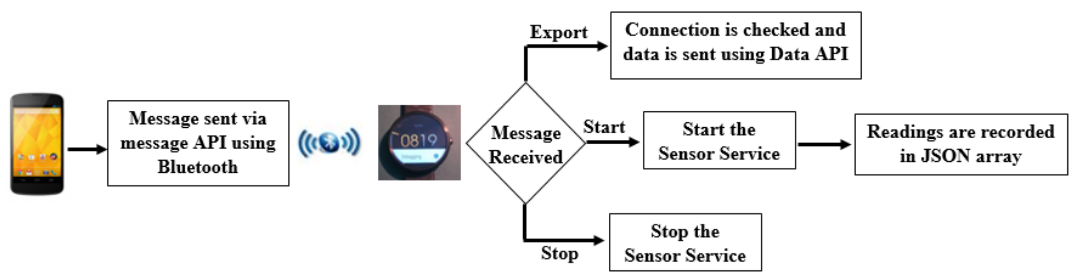


Figure 4.2: Flow of messages from phone and actions performed in Android watch.

Upon pressing the Start button on the phone application it sends a START message to the watch. On receiving this message the Sensor Service is started, enabling all the four sensors: gravity sensor, accelerometer, linear accelerometer and rotation vector sensors with a 20,000 microseconds delay. Whenever there is a change in the sensor value, sensors reading in three directions namely x-direction, y-direction and z-direction are recorded in a JSON array. When the Stop button is pressed, then a STOP message is sent. Once the Android watch receives the STOP message, disables all of the sensors and stops the sensor service.

Figure 4.3: Flow of data from android watch to phone.

When the export button is pressed, an EXPORT message is sent, and the Android watch starts sending data in batches of 3000 characters each using the Data API. At this point the phone becomes the receiver, and the Android watch becomes the sender, as shown in Figure 4.3. Before the Android watch sends data, it checks the Bluetooth connection and starts sending it. When data comes on the receiver, it starts appending the data into text file and upon completion saves the data in the phone's internal memory.

In summary, this chapter describes the steps for the identification model. The second section of this chapter discusses the apparatus used to collect the data from participants. We have a total of four files of sensor readings in three directions from which the magnitude is calculated using three direction readings. In the next chapter, we will discuss the participants and activities performed.

Chapter 5

Method

This chapter describes in detail the data collection method used for this experiment. The first subsection in this chapter discusses the participants and the second subsection explains the procedure of data collection and activities.

5.1 Data Collection

5.1.1 Participants

In this experiment, 20 participants took part voluntarily. Out of 20 participants, 10 were female, and 10 were males between the ages $20-30$ and the number of participants that wore the smart watch in left, and right hands were equal. They received the instructions for all of the activities they were supposed to perform for this experiment. The information about the participants is listed in Table 5.1.

	Left Hand	Right Hand
Male	4	4
Female	2	3
Total Participants	10	10

Table 5.1: Information about the participants

5.1.2 Procedure

This section elaborates the procedure for recording the data for call receiving and walking activities which will be for the identification experiment.

All of the participants were asked to wear the smartwatch on their hands in which they usually wear their normal watch. The smartwatch was connected to the Android phone in which data files will be saved. Once the Start message is sent to the Android Wear, the data recording starts and participants perform the activities described below. While the participants are performing the activities, the data is getting recorded in the background whenever there is a change in any of the four sensors. After the completion of each activity, a stop message is sent to the device and data is exported to the phone device.

5.1.2.1 Call Receiving (Phone on Table)

Figure 5.1 shows the activity performed. In this activity the following steps were performed to record the data:

Figure 5.1: The hand movements of a participant receiving the call when the phone is on table.

1. Participants were asked to sit on a chair, and the phone was placed on a table. They were requested to receive a phone call normally as they would while at work. They were also asked to pick up the phone from the table.

2. Participants got a call from the person who was monitoring the experiment. They both talked for 10 seconds before the call is ended.

3. Participants then placed their phones back on the table. Few participants kept their phone after locking and a few did not. One of the participants had a cover on his phone, so he kept his phone after closing the cover.

4. Repeat steps 1-3 for 9 times.

Figure 5.2a shows when a participant took out the phone and received the call and Figure 5.2b shows how the participant received the phone when it is in a sling bag. In this activity, following steps were performed to record the data:

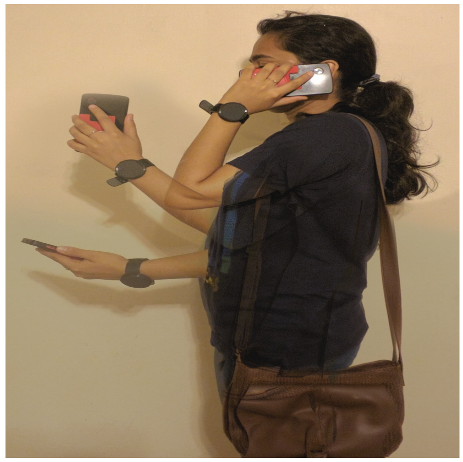
1. Participants were asked to stand normally and keep their phones in a pocket or sling bag, whichever they usually preferred to carry. They were requested to receive the phone as they normally would

2. Participants got a call in the same way as in step 2 of section 5.1.2.1.

3. Participants kept their phones either in their bags or pockets. Few participants kept their phones in their front pockets and a few in their back according to their usual habits. Two female participants kept their phone in a sling bag.

4. Repeat steps 1-3 for 9 times.

(a) Shows the hand movement of a participant receiving the call when the phone is in their pocket.



(b) Shows the hand movement of a participant receiving the call when the phone is in a sling bag.

Figure 5.2: Shows the two different types of methods when one of the participants took out the phone from their pocket and the other from the sling bag.

5.1.2.2 Walking

Figure 5.3 shows the walking action used for this experiment. In this activity, the following steps were performed to record the data:

Figure 5.3: Shows the hand movements when the participant walks.

1. Initially participants were asked to stand straight without making any motion.

2. An alarm rang which gave them the signal to start walking. They were requested to walk as they walk on the street. Participants were engaged in conversations so that they did not get conscious while walking.

3. Another alarm rang after 1 minute giving them the signal to stop.

4. Repeat steps 1-3 for 9 times.

In each activity, readings in three directions, namely x-direction, y-direction and z-direction, were recorded for all four sensors enabled for this experiment. The text file in which readings were stored at the end of the activity is further processed to save the sensors reading in four separate files for the four sensors.

In summary, this chapter discusses about participants and activities they performed. We have a total of four files of sensor readings in three directions for each activity, and the magnitude is calculated using three direction readings. In the next chapter, we will discuss the graphical techniques to analyze the data and the features which can be computed to train the classifier for identification.

Chapter 6

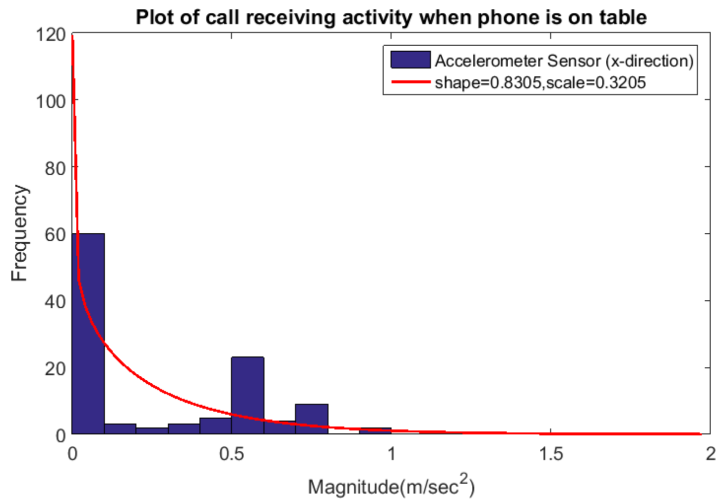
Data Analysis

This chapter discusses the graphical techniques used to analyze the data. This chapter also discusses the feature selection and classifiers used for this experiment.

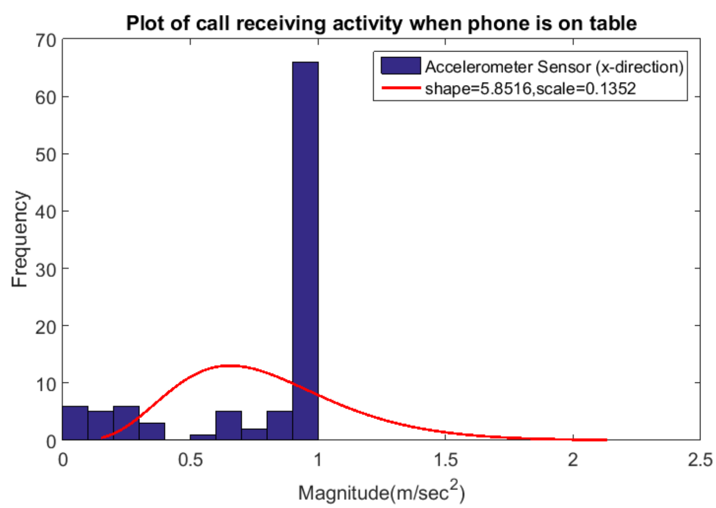
The techniques used to find the essential characteristics are as follows:

6.1 Histogram

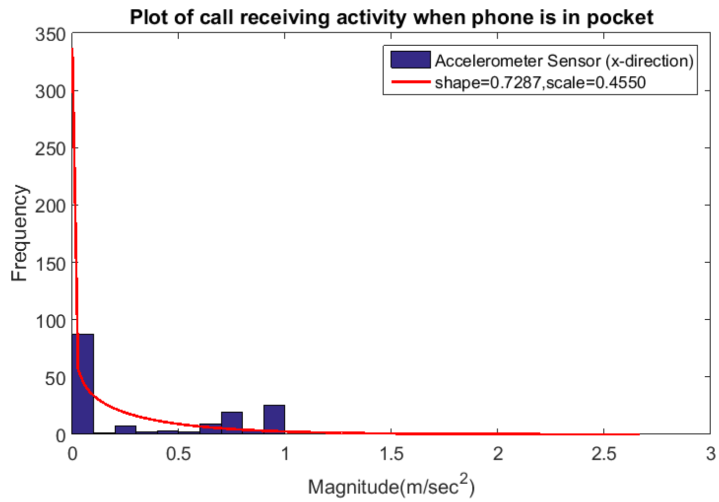
A histogram is a graphical representation of the continuous data. Its inspection tells us about the distribution, outlier and other characteristics of the data. Here the amplitude of sensors is random within the finite set of values. The distribution of values is assumed to belong a gamma distribution family. Figure 6.1 shows the histogram plots fitted with a gamma distribution curve of the Accelerometer in the x-direction for participant 1 and participant 2 for three activities.



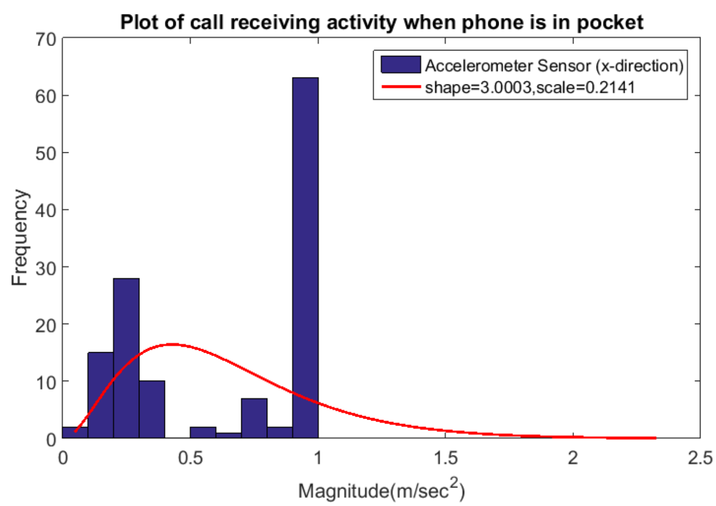
(a) Histogram plot fitted with a gamma distribution curve of accelerometer data in the x-direction obtained from the smartwatch, when participant 1 is receiving a call. Shape and scale value for gamma distribution curve is 0.83 and 0.32 respectively. Maximum frequency range for this activity is between 0 and 0.1.



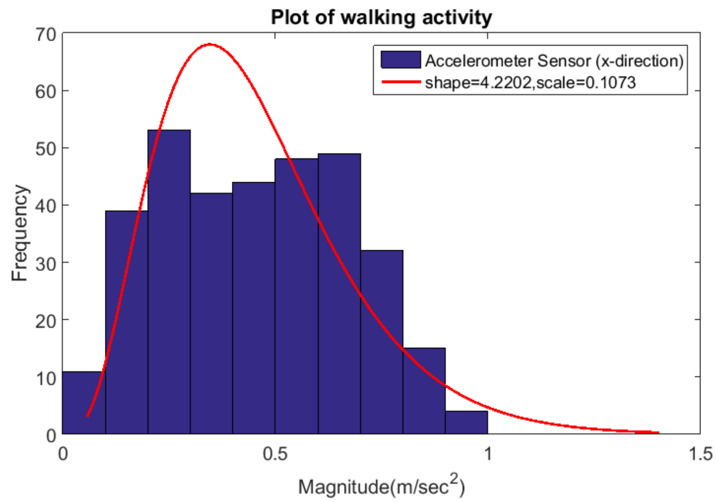
(b) Histogram plot fitted with a gamma distribution curve of accelerometer data in the x-direction obtained from the smartwatch, when participant 2 is performing the same activity performed in Figure 6.1a. Shape and scale value for gamma distribution curve is 5.85 and 0.135 respectively. Maximum frequency range for this activity is between 0.9 and 1.0.



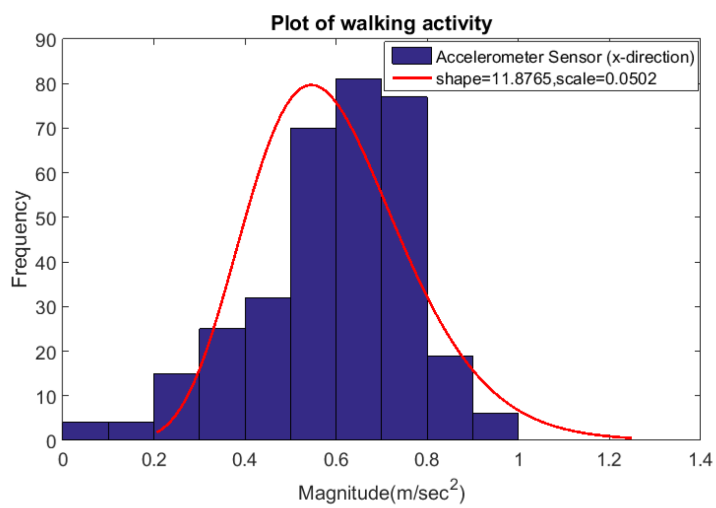
(c) Histogram plot fitted with a gamma distribution curve of accelerometer sensor data in the x-direction obtained from the smartwatch, for the activity in which participant 1 received the call when phone is in their pocket. Shape and scale value for a gamma distribution curve is 0.72 and 0.45 respectively. Maximum frequency range for this activity is the same as in Figure 6.1a.



(d) Histogram plot fitted a gamma distribution curve of accelerometer data in the x-direction obtained from the smartwatch, when participant 2 is performing the same activity performed in Figure 6.1c. Shape and scale value for a gamma distribution curve is 3.0 and 0.21 respectively. Maximum frequency range for this activity is same as Figure 6.1b.



(e) Histogram plot fitted with a gamma distribution curve of accelerometer data in the x-direction obtained from the smartwatch, when participant 1 is walking. Shape and scale value for a gamma distribution curve is 4.22 and 0.10 respectively. Maximum frequency range for this activity is between 0.2 and 0.3.



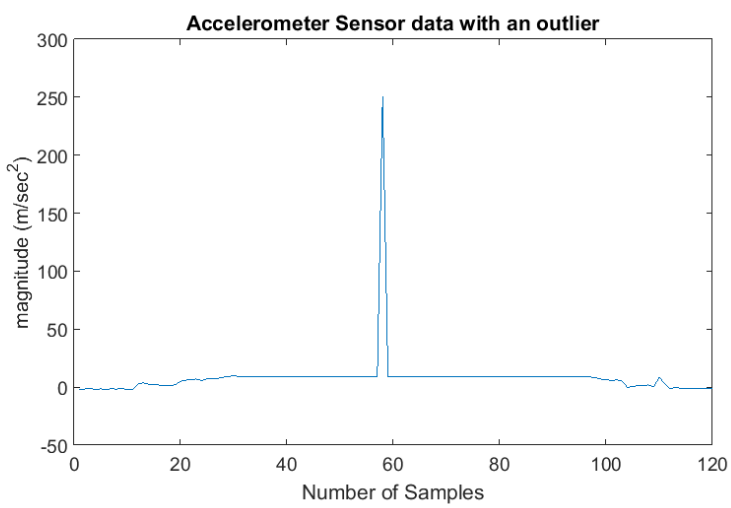
(f) Histogram plot fitted with a gamma distribution of accelerometer data in the x-direction obtained from smartwatch, when participant 2 is walking. Shape and scale value for a gamma distribution curve is 11.87 and 0.0502 respectively. Maximum frequency range for this activity is between 0.6 and 0.7.

Figure 6.1: Shows the difference between different participants performing the same activities and similarity in call receiving activities for the same participant.

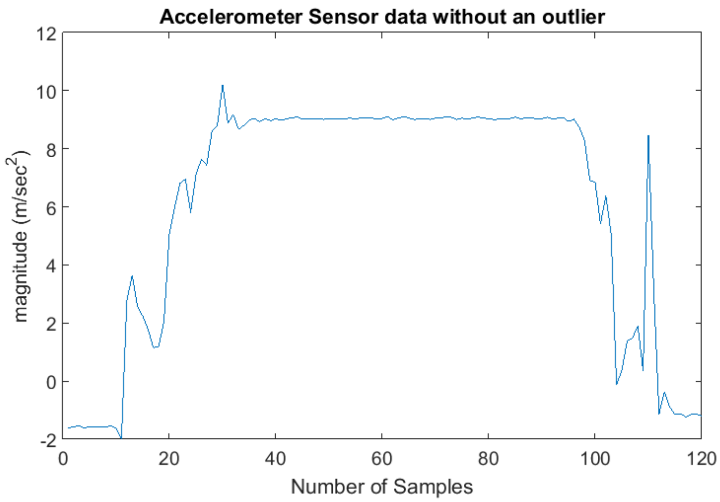
From Figure 6.1a) and 6.1b) observe that for participant 1 the maximum frequency is in the range of 0-0.1 and for participant 2 it is 0.9-1.0 in the case of call receiving activity when the phone is on the table. Figure 6.1a and 6.1c shows that for participant 1's call receiving activity the maximum frequency range is the same i.e. 0-0.1; whereas, for participant 2 Figure 6.1b and 6.1d indicates that it is in the range of 0.9-1.0.

6.2 Winsorized Mean and Winsorized Variance

In winsorized mean, the average value of all the observations is calculated using arithmetic formula after replacing the p% of smallest and the largest observations in the data by the closest observation to them. Winsorized mean is an improvement over the standard mean in this data analysis because it removes the outliers from the data. For this experiment, 10% of the largest and smallest observations of the data are replaced by the closest values. Similar to the standard mean, the standard variance estimate also is widely affected by the outliers; that is why 20% of observations are winsorized for calculating variance.



(a) Data with an outlier.



(b) Data without an outlier.

Figure 6.2: Accelerometer data in x direction for call activity. Here in graph (a) the outlier is positioned at sample number 58 with a magnitude of 256.66 will affect the mean. Graph (b) shows the plot after the replacement of the outlier sample with the closest smallest value.

.

From the above plot of the call activity, we can see that there is an outlier of value 256.66. This outlier will affect in the mean value of readings. To remove the effect of outliers, the winsorized mean and winsorized variance are calculated.

Figure 6.3 is a mean versus variance plot of Accelerometer data in the x-direction for six trials for all of the activities. In this graph, we can see that for different participants, and activities there are different clusters. From this, we conclude that the mean and variance can be possible features for training the classifier.

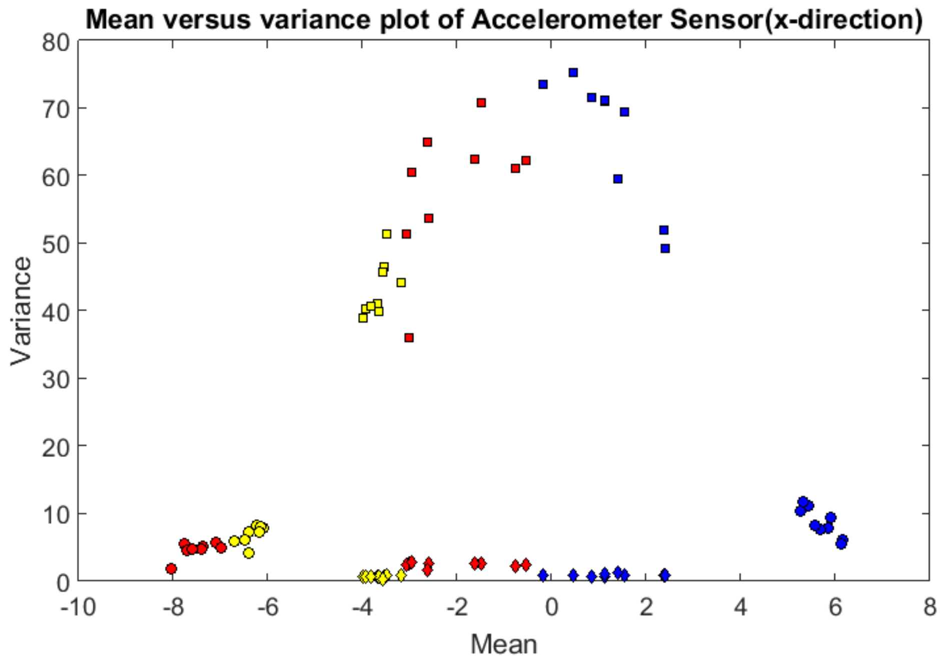


Figure 6.3: Mean versus variance plot of Accelerometer data in in x-direction for 3 participants and all the activities. All the subjects can be grouped into clusters. Thus, mean and variance are the possible features for the identification model.

In Figure 6.4 we observe that the magnitude of gravity for all of the users and for the activities lies in the range of 9.80 $\mathrm{m\,s^{-2}}$ to 9.81 $\mathrm{m\,s^{-2}}$. Due to this observation, the mean and variance of the magnitude of the gravity sensors is not used for training the model. Similarly, in the case of the rotation vector sensor readings in the x-direction, y-direction and z-direction are not distinguishable in clusters. This can be seen in Figures 6.5, 6.6, 6.7.

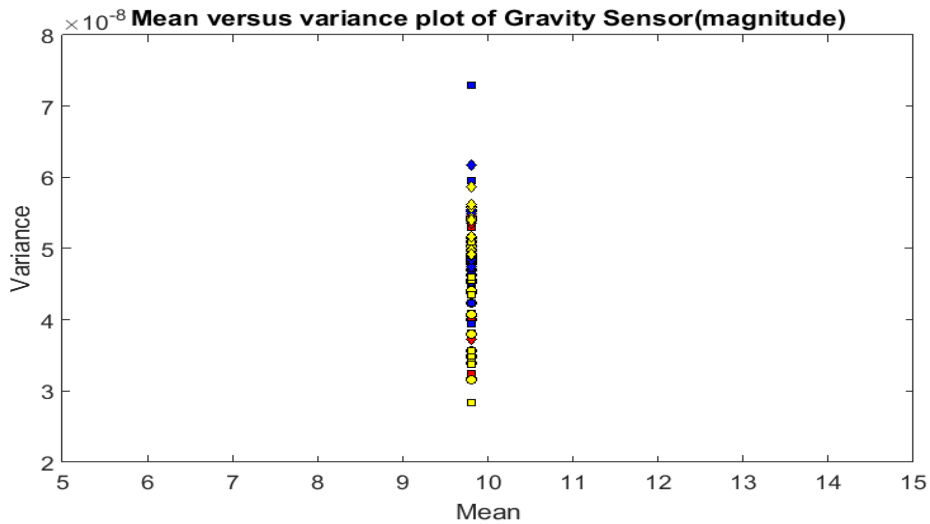


Figure 6.4: Mean versus variance plot of the magnitude of the gravity sensor data for 3 participants and all the activities. For all of the participants the value lies in the range of 9.8 to 9.81 $\mathrm{m\,s^{-2}}$.

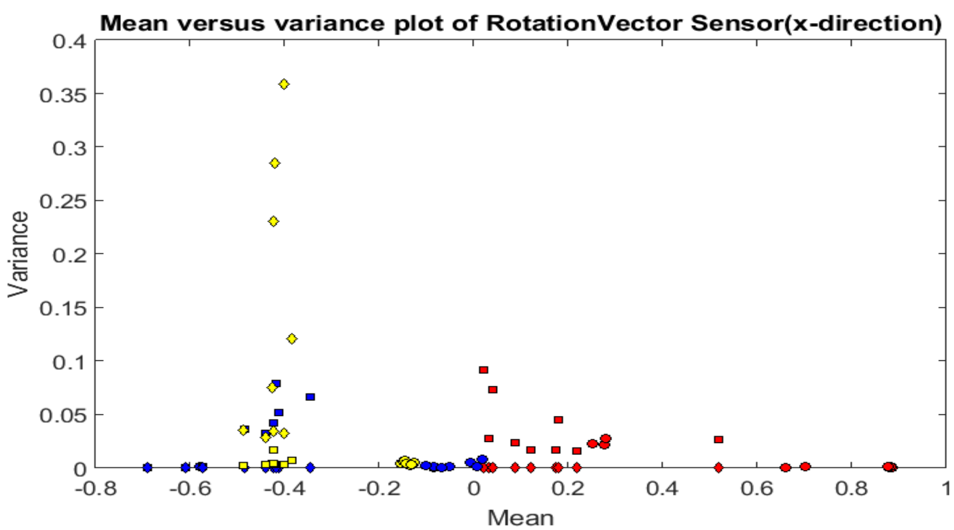


Figure 6.5: Mean versus variance plot of the rotation vector sensor data in the x-direction for 3 participants and all the activities. All of the subjects and activities points are not forming clusters as in Figure 6.3.

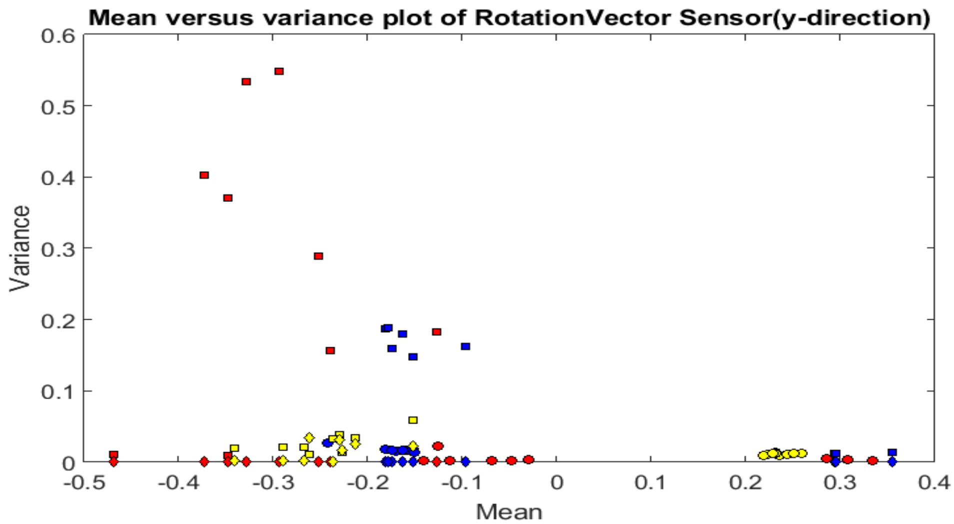


Figure 6.6: Mean versus variance plot of the rotation vector sensor data in the y-direction for 3 participants and all of the activities. All of the subjects and activities points are not forming clusters as in Figure 6.3.

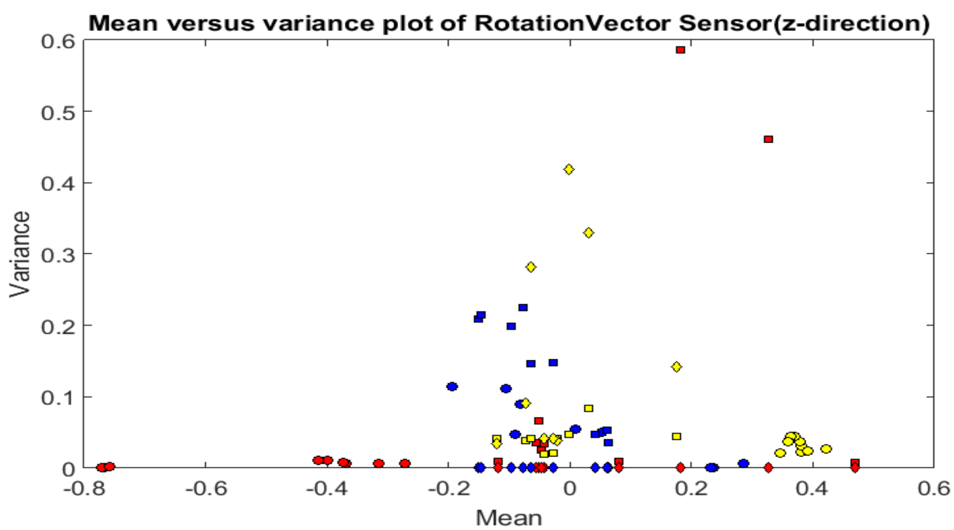


Figure 6.7: Mean versus variance plot of the rotation vector sensor data in the z-direction for 3 participants and all the activities. All of the subjects and activities points are not forming clusters as in Figure 6.3.

6.3 Cross-Correlation

Cross-Correlation is a method to determine to what degree the signals are related to each other. Suppose there are two signals X and Y. When X and Y are similar, their integral product is maximized. If X and Y are the exact opposite of each other, then the product of negative-negative is also maximized. On the other when two signals, do not match then cross-correlation is minimum. In this experiment, we have found the cross-correlation between each participant and each activity for all the sensors in three directions and magnitude.

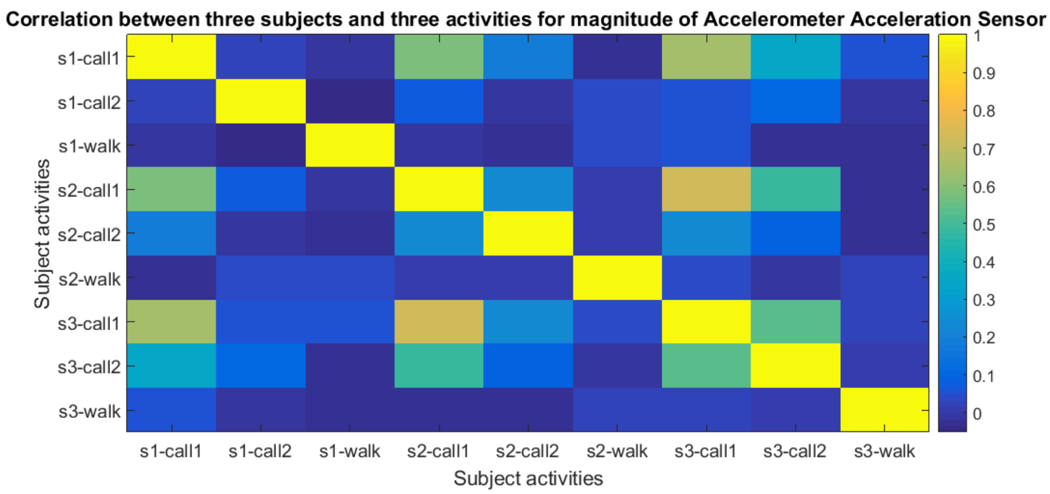


Figure 6.8: Shows the cross-correlation calculated between 3 participants and all the activities. The yellow color is showing that they are highly correlated as the cross-correlation is computed between the same person and activity. Dark shades of blue demonstrate that the activities and person are not related; therefore, they can be identified using the information of these activities.

From Figure 6.8 we can see that the participants are not correlated to each other. Correlation cannot be used as a possible feature to train the system because it is dependent on the activities and future participants. If we are training the system with new activity, then the cross-correlation between all activities and new activity has to be calculated. Whereas if we are using statistical features like mean-variance then simply adding these features of new activity will work.

6.4 Features

6.4.1 Winsorized Mean and Winsorized Variance

In Section 6.2 we see that winsorized mean and variance form clusters when plotted. Therefore, for training the classifier winsorized mean and variance of Accelerometer and Linear Acceleration data in the x, y, z directions and their magnitude will be used. In the case of Gravity sensor, we are using only data in the x, y and z-direction because magnitude is not different for various activities as observed in Figure 6.4. Similarly, in the case of the Rotation Vector Sensor we are only using mean and variance calculated from the magnitude of the sensor data because in every trial the x, y and z-directions are not lying in the same range. From the Figure 6.5 we can also see that they are distinguishable in clusters.

6.4.2 Gamma Distribution

Gamma Distribution is a two parameters type of continuous probability distributions. It is related to the beta distribution. The two free parameters are α and θ [38]. In this chapter, we saw that the Gamma distribution curve fits the histogram well. Shape and scale factor for each participant lies in the same range and for different participants they are different. From this observation, they are considered as a possible feature for the identification model.

6.4.3 Skewness

Skewness is a measure of the asymmetry of the probability distribution of the data around the sample mean. The skewness of any perfectly symmetric distribution is zero, for example the normal distribution. If the skewness is negative, it means that the data is spread out more to the left of the mean. If the skewness is positive, the data are spread out more to the right than to the left. Skewness is given by the formula below:

$$s = \frac{E(x-\mu)^3}{\sigma^3} \tag{6.1}$$

where, E(t) is the expected value of the quantity, μ is the mean and σ is the standard deviation. As skewness also depends on the mean and standard deviation, which is equal to square root of the variance, it can also be a possible feature for training the classifiers of the identification system.

6.5 Feature Set

Figure 6.9 shows the feature matrix for one subject. Section 6.2 concludes that the magnitude of the gravity sensor and data of the rotation vector sensor of three directions is not useful. Therefore, for the extraction of features, only 12 sources of data are considered. For one subject, each activity has a feature vector of size 1 x 60 as five features are calculated for each data source. There are nine trials of each activity.

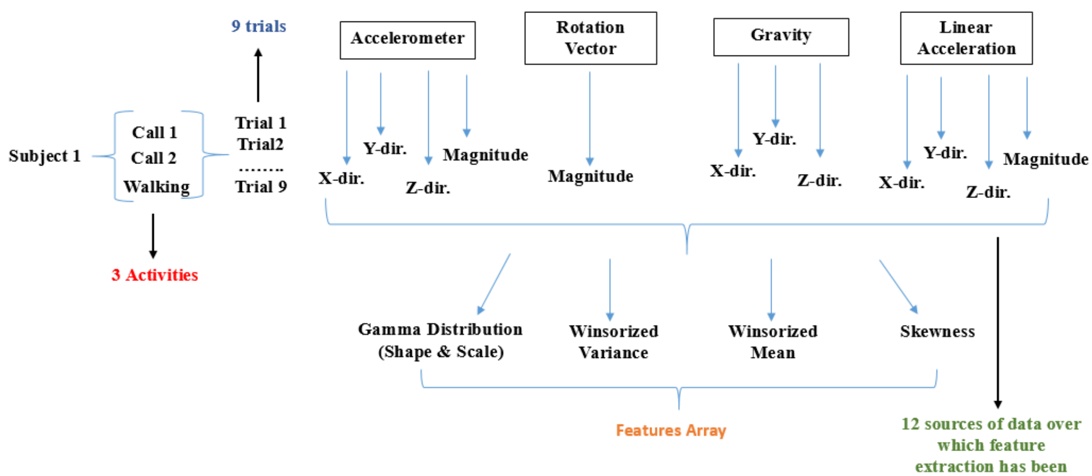


Figure 6.9: Shows the feature array for training the classifier.

So, stacking all of the nine trials row-wise, we have a feature matrix of size 9 x 60. Now, there are three activities, and each activity has a 9 x 60 feature matrix. Stacking each 9 x 60 matrix row-wise one beneath the other, we end up now having a final 27 x 60 feature Matrix. While training, all of these 27 rows will have the same label representing that they all collectively represent one specific subject.

6.6 Classifier

6.6.1 Support Vector Machines (SVMs)

Support Vector Machines (SVM) are popular machine learning methods that belongs to a group of supervised learning algorithm. SVM was invented by Vladimir N. Vapnik and Alexey Ya. Chervonenkis in 1963. It can be used for classification, regression, and other learning tasks. Classification of two or more classes is a very common task in machine learning. In SVM, classification is performed by constructing a multidimensional hyperplane. There can be many hyperplanes classifying the same data, so the best hyperplane is the one having the largest distance to the nearest data point of any class. Such a hyperplane is known as a maximum-margin hyperplane. The first maximum-margin hyperplane algorithm is best for linear classification. In 1992, Vladimir N. Vapnik, Bernhard E. Boser and Isabelle M. Guyon suggested replacing every dot product in the original algorithm with a nonlinear kernel function [39]. Some popularly used kernels are Linear, Polynomial, Radian basis function (RBF)and Sigmoid. RBF and Polynomial kernels are popularly employed in SVM for classifying non-linear separable data-points. In this research, Linear, RBF and Sigmoid Kernel is used for classification.

6.6.1.1 Linear Kernel

Linear Kernel is one of the simplest kernel functions. The difference between the Standard SVM classification and SVM with a linear kernel is of a constant. In standard SVM, classification is done using just the dot product; whereas, the linear kernel function is the dot product plus a constant. Linear kernel performs well when the number of features is larger than the training data.

$$k(x, y) = x^T y + c \tag{6.2}$$

In SVM with a linear kernel, only Cost Factor (C) or penalty factor is critical. The cost factor is also known as a Soft margin. A soft margin is useful when the data is not linearly separable as it ignores few data points or places them on the wrong side of the margin. If C is very high, then the model becomes sensitive to outliers and overfitting occurs. If it is too small, then under fitting occurs. In Training a SVM model, a linear kernel performs faster than training with another kernel.

6.6.1.2 Gaussian Kernel

Gaussian Kernel is a type of radian basis function (RBF) kernel. The RBF kernel is generally used when data is not linearly separable. So data can be mapped into higher dimensional space using this kernel where separation can be easily achieved. A Gaussian kernel is implemented by the equation given below:

$$k(x, y) = exp(-\gamma||x - y||^2) \tag{6.3}$$

In the RBF kernel, there are two parameters which play important role in classification. The first parameter is the cost factor C as explained in section 5.1.1.1. It also controls the tradeoff between errors of the SVM on training data and margin maximization. The second parameter is Gamma, which is also known as the kernel coefficient. If gamma is small, then Gaussian is with a large variance, implying that if y is a support vector then this will have more effect on deciding the class of vector x even if the distance between them is large. Small gamma leads to low bias and large variance models, and vice-versa. If gamma is large, then the variance is small, implying y will not have influence on the decision making of class of vector x. Using Grid Search we can find the best possible combination of C and gamma [40].

6.6.1.3 Sigmoid Kernel

Sigmoid Kernel is a popular kernel in SVM because its origination is from the neural networks. Sigmoid Kernel is also known as Hyperbolic Tangent Kernel and as the Multilayer Perceptron (MLP) kernel. The SVM model using a sigmoid kernel works equivalent to perceptron neural network of two layers. The Sigmoid kernel function is given below in equation 6.4

$$k(x, y) = tanh(\alpha x^T y + c) \tag{6.4}$$

It has two parameters which can be adjusted, the slope alpha and the intercept constant c. If alpha is greater than 0, it can be considered as a scaling parameter of the input data and c as a threshold controlling shifting parameter of mapping. If α is less than 0, then the dot product is reversed together with scaling. When α is positive and c is negative then it is more suitable for the sigmoid kernel function. In this combination of α and c, if c is small enough then the sigmoid kernel matrix is conditionally positive definite (CPD) and thus a valid kernel. When α is also small then the sigmoid kernel will behave like the RBF kernel [41].

In summary of this chapter, we discussed the different graphical techniques to understand the characteristics of the data. In the chapter, later we discussed the possible features which can be used to identify the subjects. In the last section, we discussed the classifiers which will be used in identification of the subjects. In the next chapter, we will discuss the results using features and classifiers as were above.

Chapter 7

Results

This chapter discusses our results using the SVM classifier with a different kernels. We are following the forward feature selection procedure to analyze the performance of a classifier with different kernels. In the first section, all of the calculated features will be used individually to train the system and calculate the Equal Error Rate (EER) for all of the different kernels. Then combinations of the best feature from the first section, with other features, is used to determine the improvement in performance. The effect on EER due to the parameters of the kernel is also presented. In the last section, the selected parameters and features are used which resulted in the lowest EER which allowed us to observe the performance of the system with a fewer number of participants.

7.1 Equal Error Rate EER

A receiver operating characteristic (ROC), or ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR). For an ideal result, the ROC curve passes through the upper left corner. The EER is the point where ROC curve intersects the line FPR=1-TRP. In the ideal case, the value of EER is 0. Therefore, the lower the EER, the higher the overall accuracy.

7.2 *k*-fold Cross-Validation

Cross-validation is a technique used to evaluate the predictive performance of the statistical model. Cross-validation is performed to avoid the over fitting and evaluate how the model will perform with independent data. In the original model, the data is partitioned into training data and test data. The Cross-validation technique introduces the data to test in the training phase. This data is known as validation data. We are using the k-fold cross validation technique in which training data is divided into k-folds roughly containing an equal number of samples in each fold. Here in this experiment, k is equal to 4. In the k-fold cross-validation one-fold is used as the validation data and remaining *k-1* folds are used for training. This process is repeated k times in-order to use each fold one time as a validation data. The single cross-validation accuracy is obtained by averaging the k results from all the folds.

7.3 Why Forward Feature Selection ?

We are using the forward feature procedure. In this we first analyze the accuracy of the model using one feature at a time. Then selecting the best from this analysis and making combination with other features. By using this technique, we are able to select out three feature from the combination array to analyze the model. We used this approach in order to avoid over fitting by using a high dimension feature set. Thus, by using only the best features we are able to determine the lowest EER.

7.4 1 Feature Array

In this section, we first calculate the cross-validation accuracy by dividing training data (70% of data) into k folds. We are using only one estimated features of all the sensors data shown in Figure 6.9. Figure 7.1 shows that the mean feature is results in the lowest cross-validation error percentage which indicates that we are training our classifiers only with all of the training data. This process allows us to determine which feature gives the lowest possible EER with all of the kernels. When using the mean, variance or skewness feature, the array size is 27x12; whereas, when the system is trained with gamma distribution parameters (shape and scale) feature array is of 27x24 dimension. In MATLAB, a program was written for 10 test cases to randomly select from 1, 2, . . . 10 testing samples at a time from each subject and train with the remaining samples. For the SVM classifier, the LIBSVM library is available for MATLAB. To analyze the performance of the kernels, an Average EER of 100 iterations is calculated.

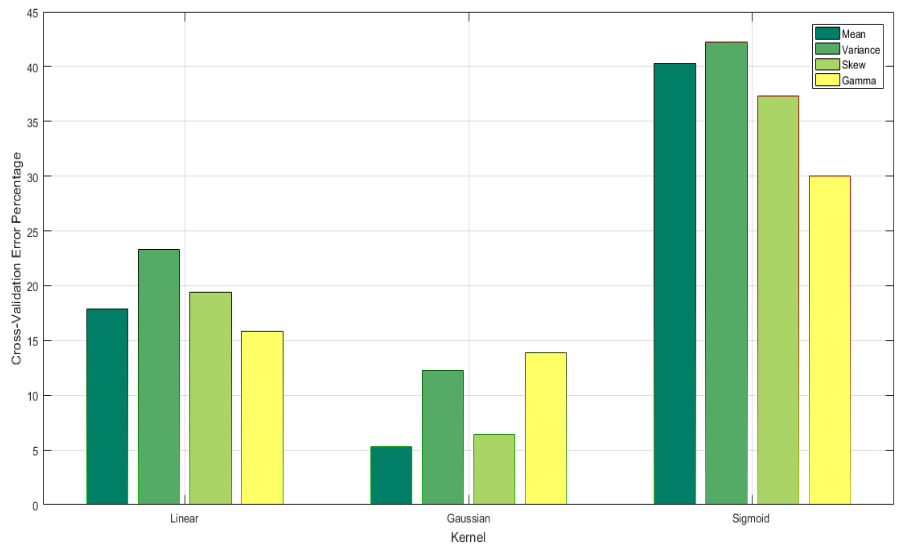


Figure 7.1: k-fold Cross-validation error percentage where k=4. Here the model is trained with one feature from the feature set.

7.4.1 SVM Classifier with Linear Kernel

In Figure 7.2 we observe that the Lowest EER values for all of the test cases are obtained when training the system with feature mean. The lowest EER is when the number of test sample selected from each participant data set is 1. Therefore, testing dataset is of 20 test samples. In the case of training set, there are 520 training samples, 26 for each participant. The lowest EER among all of the features is 0.2375 when the system is trained with the Gamma Distribution parameters, and 4 test samples from each participant are selected.

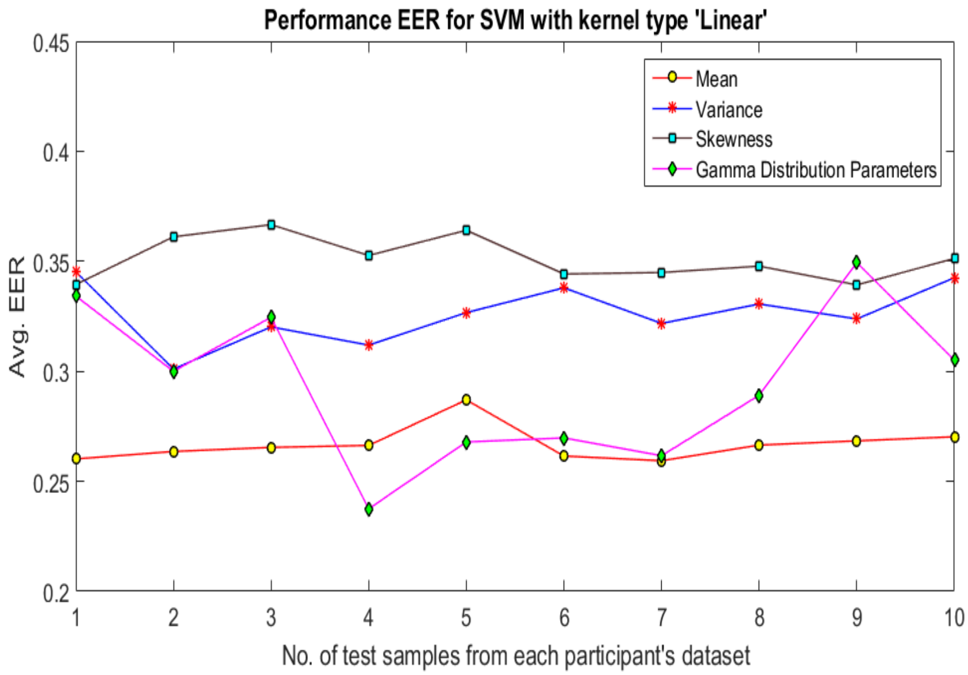


Figure 7.2: Performance of the system when trained with different 1 features array sets. Here the classifier is SVM with Linear kernel. The Mean feature is performing well for all of the test cases. The lowest EER for this system is 0.2375 when trained with Gamma Distribution parameters.

7.4.2 SVM Classifier with Gaussian Kernel

When a Gaussian kernel is used with SVM the EER is less in the case of mean as a feature. This can also be observed from the Figure 7.3. The lowest EER is 0.058 when two samples are randomly selected for testing the system from 27 trials of each participant. The remaining 25 trials will be used for training the system. This result shows that SVM with the Gaussian kernel is working better than the SVM with the linear kernel.

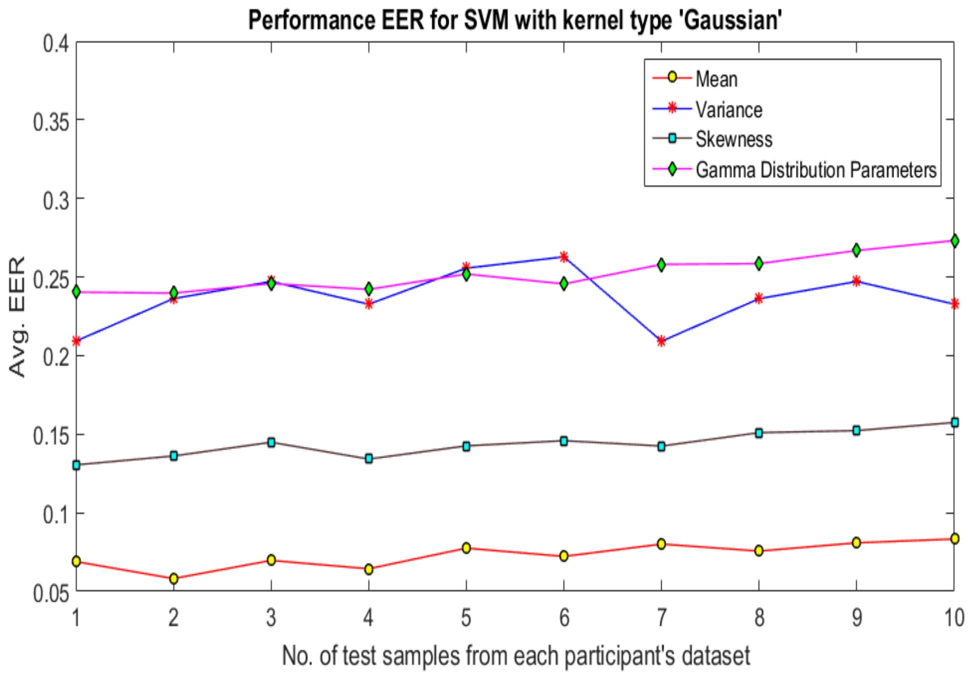


Figure 7.3: Performance of the system when trained with different 1 features array sets. Here the classifier is SVM with Gaussian kernel. Mean feature is performing well for all of the test cases.

7.4.3 SVM Classifier with Sigmoid Kernel

Figure 7.4 show that for all the test cases the lowest values of EER occur when the skewness feature is used for training and testing. But the lowest EER is 0.41 in case of mean as the feature for 1 test sample from each participant.

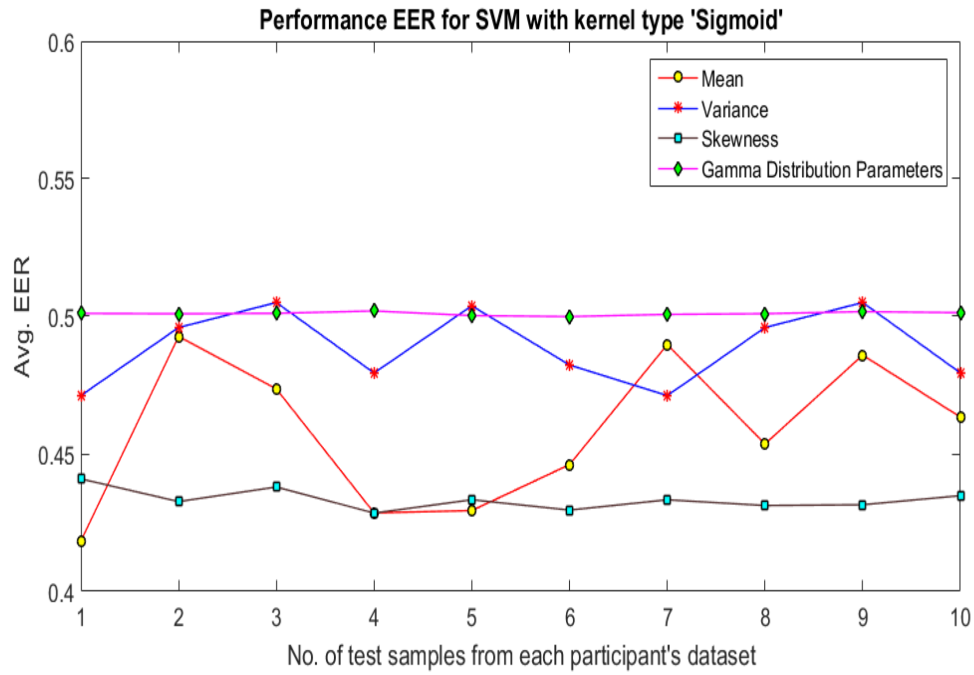


Figure 7.4: Performance of the system when trained with different 1 features array sets. Here the classifier is SVM with a Sigmoid kernel. Skewness feature is performing well for all the test cases.

As can be determined from the above graphs, the identification system performed well when the mean is used as the feature set for training the system. In this case, SVM with Gaussian kernel is performing better than the other two kernels. This can also be seen in Figure 7.5.

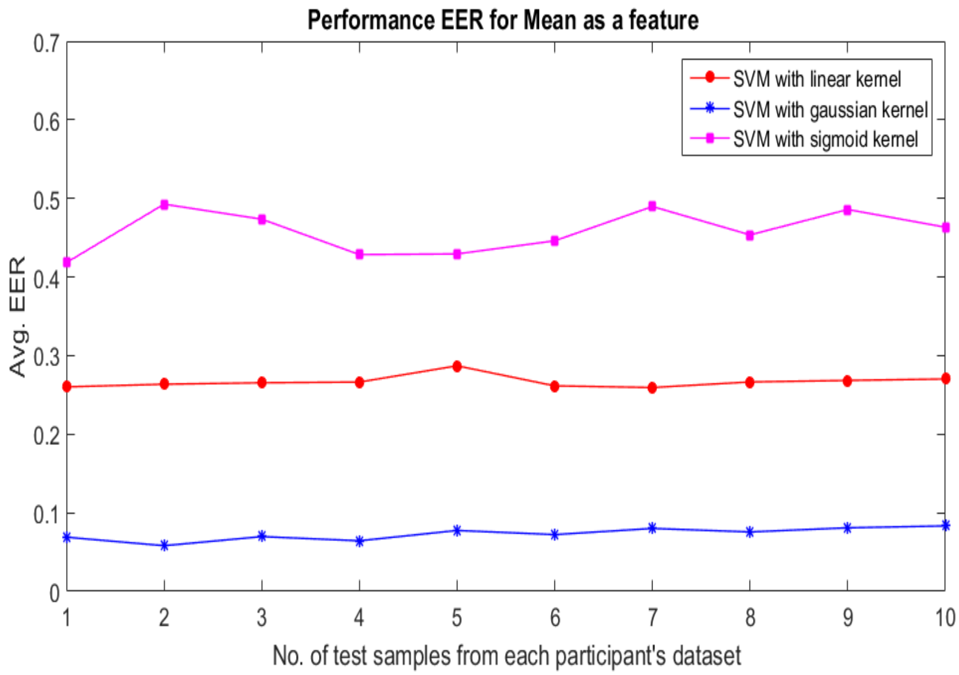


Figure 7.5: Performance of different kernels when the mean is used as a feature for training and testing the identification system.

7.5 2 Features Array

In Section 7.4, it was concluded that the mean outperformed all the other features by giving the lowest EER when SVM with a Gaussian kernel is used for classification. In this section, we begin by first calculating the cross-validation accuracy by dividing training data (70% of data) into k folds. We are using a combination of the mean with other calculated features. Figure 7.6 shows that combination of the mean with skewness and gamma distribution with the Gaussian kernel is resulting in the lowest cross-validation error percentage. There are three possible combinations: mean-variance, mean-skewness, and mean-gamma distribution parameter (Shape and scale). So the feature vector would be of dimension 1x24 for the case of the case of mean-variance and mean-skewness; whereas, for the mean-gamma distribution parameters it has a dimension 1x36.

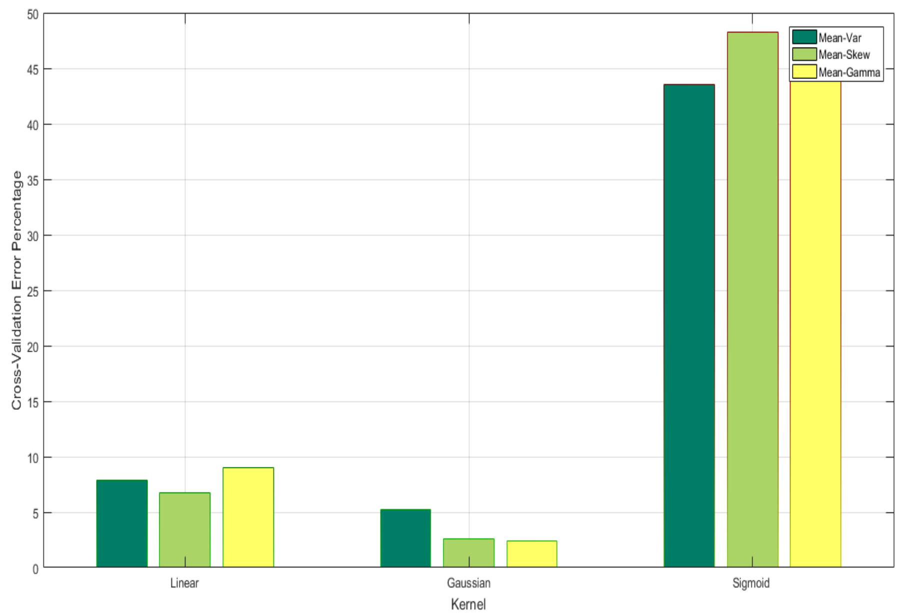


Figure 7.6: k-fold Cross-validation error percentage where k=4. Here the model is trained with a combination of mean feature with other features.

7.5.1 SVM Classifier with Linear Kernel

In Figure 7.7, we observe that the lowest EER occur for the mean-variance feature set for all the test cases when SVM, with the Linear kernel, is used for the classification. In this identification system, the lowest EER is 0.139, when the number of test samples selected are 1 and 8 randomly from each participants data set. In the case where number of test samples = 8, training set is of size 380x24 and testing data set size is 160x24.

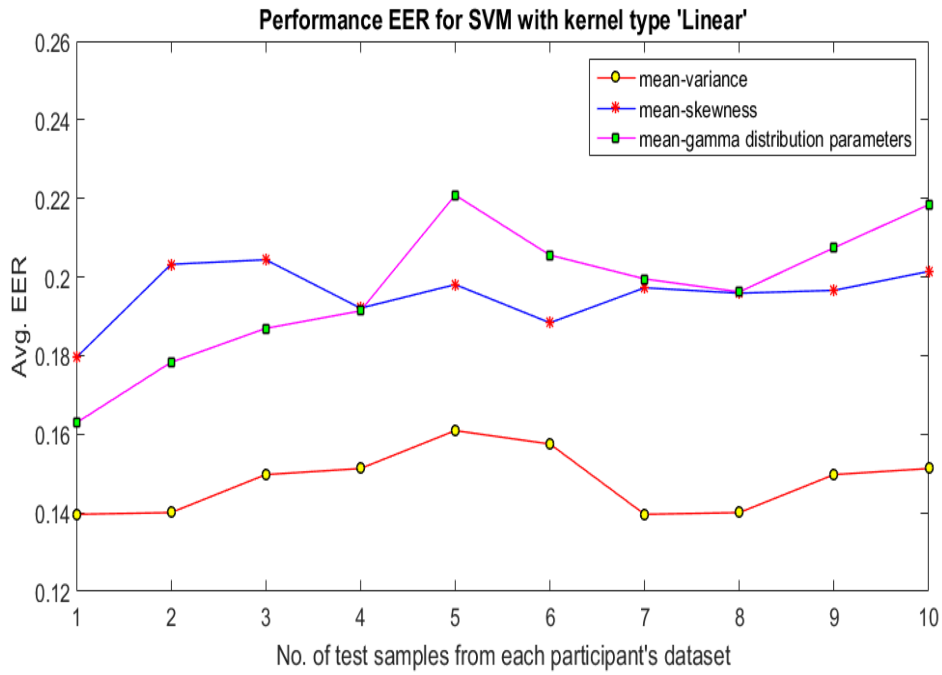


Figure 7.7: Performance of the system when trained with different 2 features array sets. Here the classifier is SVM with a Linear kernel. Combination of mean and variance features is performing well for all the test cases with lowest EER equal to 0.139.

7.5.2 SVM Classifier with Gaussian Kernel

From Figure 7.8, it can be seen that in this identification system, a combination of mean and skewness performs better than the other two feature sets. In this case, the lowest EER observed is 0.04 when the number of test samples from each participant is equal to 1. For 10 randomly selected test samples from each participants dataset, the average EER for 100 iterations is 0.06. This identification system is performing better than the identification system discussed in Section 7.4.2.

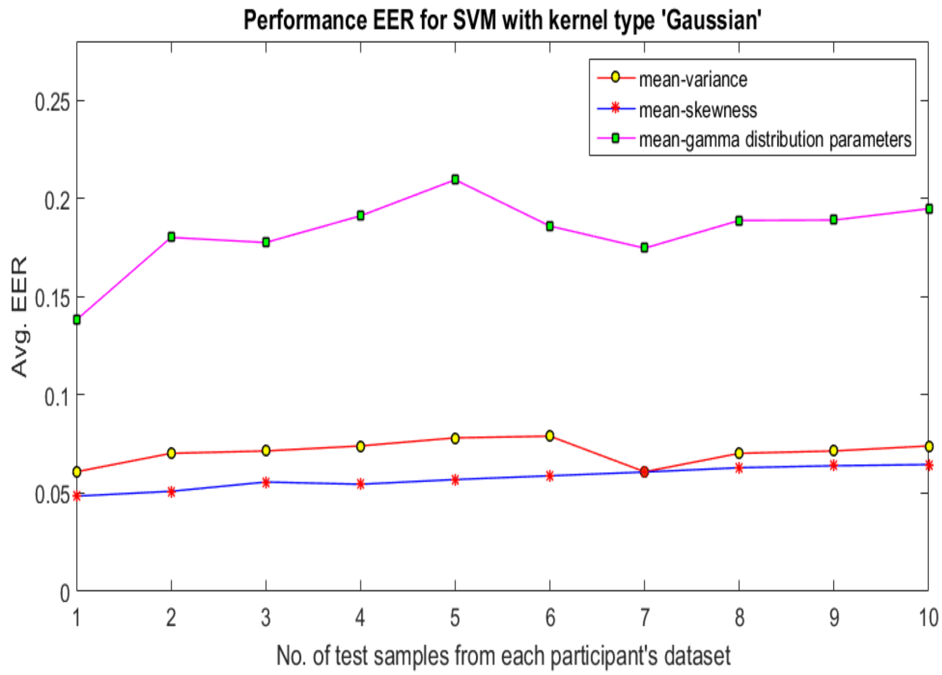


Figure 7.8: Performance of the system when trained with different 2 features array sets. Here the classifier is SVM with a Gaussian kernel. The combination of mean and skewness features is performing well for all the test cases with lowest EER equals to 0.048.

7.5.3 SVM Classifier with Sigmoid Kernel

From Figure 7.9, it is observed that the EER is higher for all of the test cases than the last discussed classifiers. Though the EERs are higher for all the test cases, the overall mean-variance combination is identifying objects more accurately among the all feature sets.

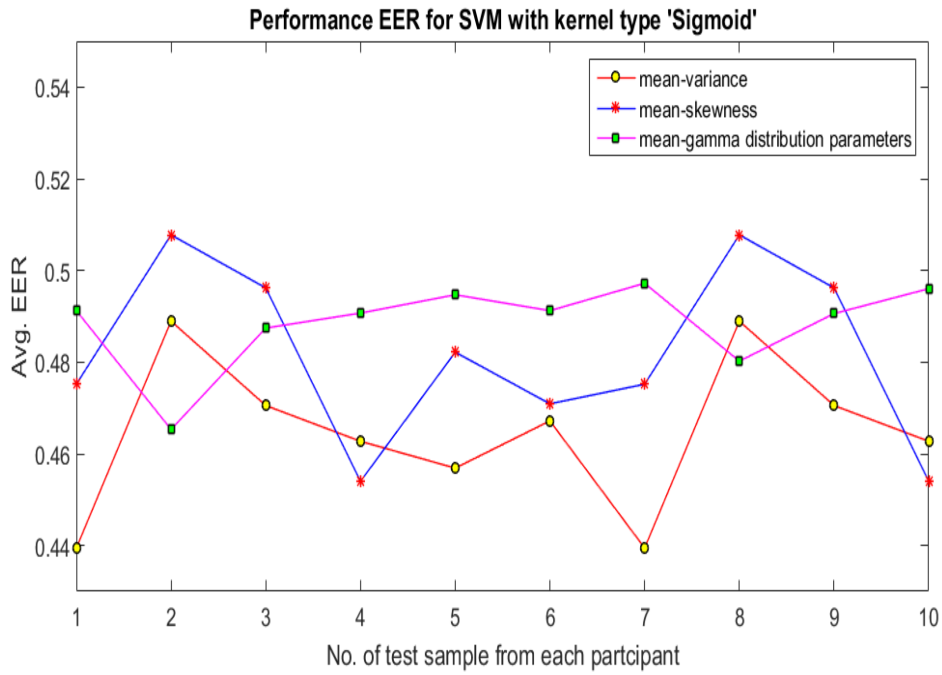


Figure 7.9: Performance of the system when trained with different 2 features array sets. Here the classifier is SVM with Sigmoid kernel. The combination of mean and variance features is performing well for all the test cases, but this classifier is performing worse.

In summary, the SVM classifier with a Gaussian kernel is performing better than the other kernels, such as those discussed in Section 7.4. Here the combination of mean-variance and mean skewness performed more accurately than the other feature sets. Therefore, in the next section, the system is trained with three features array combining mean-variance and mean-skewness with other remaining features.

7.6 3 Features Array

In the last section, we observed that the combination of mean-variance and mean-skewness performed well. So in the section we will first calculate the cross validation accuracy of the system with the mean-variance-skewness, mean-variance-gamma distribution parameters and mean-skewness-gamma distribution parameters features arrays. Figure 7.10 shows the model results in the lowest cross-validation error percentage when we cross validate with the combination of mean, variance, and skewness.

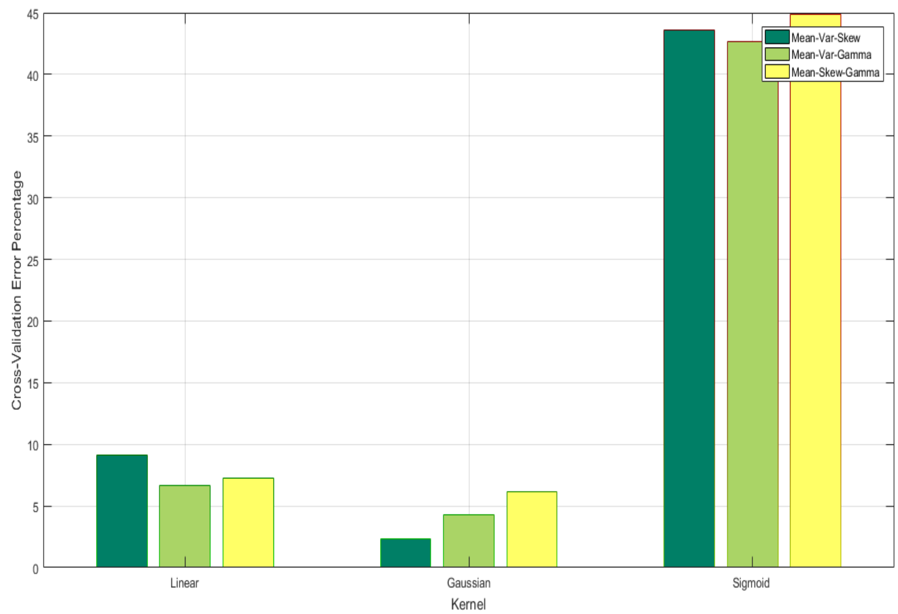


Figure 7.10: k-fold Cross-validation error percentage where k=4. Here the model is trained with the combination of three features from the feature set.

7.6.1 SVM Classifier with Linear Kernel

From Figure 7.11 we observe that for all the cases, except when the number of test samples =1, the combination of mean, variance and skewness identifying participants accurately achieving the lowest EER of value 0.11. Therefore, from this system we conclude that the mean, variance and skewness features could be used to train the final identification system.

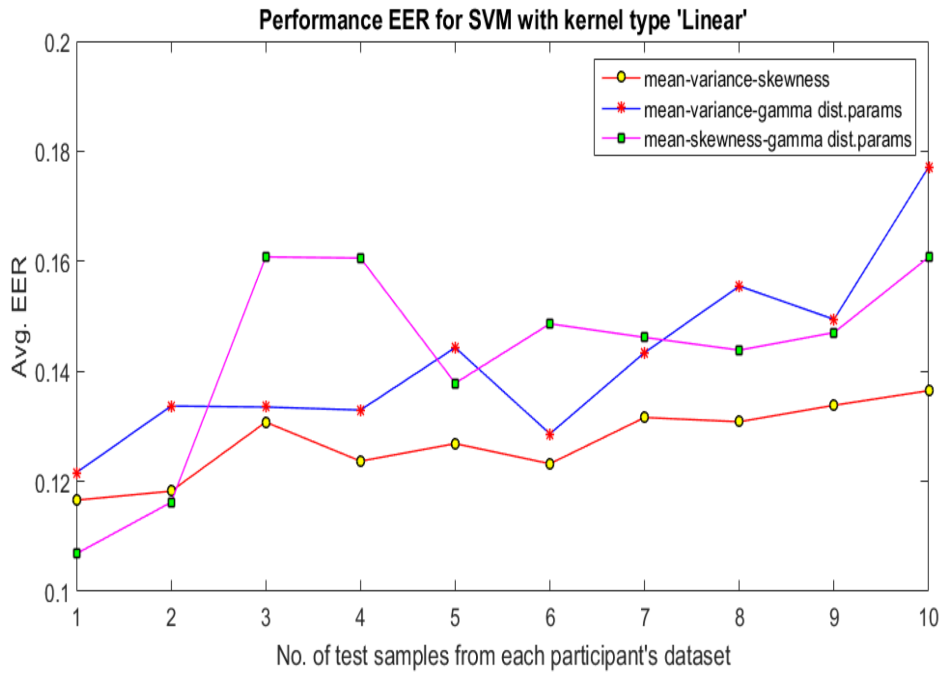


Figure 7.11: Performance of the system when trained with different 3 features array sets. Here the classifier is SVM with Linear kernel. Combination of mean, variance and skewness features is performing well for all the test cases with lowest EER equals to 0.11.

7.6.2 SVM Classifier with Gaussian Kernel

As we observed in section 7.6.1 the mean, variance and skewness features performed well. From Figure 7.12, we can also observe that this feature set is performing better than others when the classifier is SVM with a Gaussian kernel. Here the feature vector is of size 1x36. The lowest EER achieved is 0.04 for the number of test sample =1 from each participant.

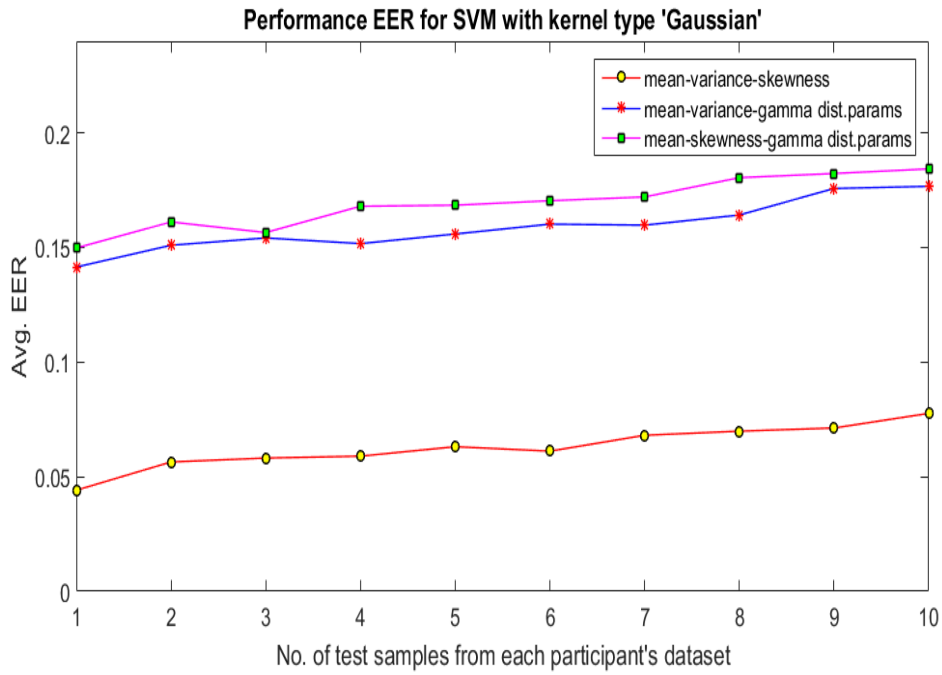


Figure 7.12: Performance of the system when trained with different 3 features array sets. Here the classifier is SVM with Gaussian kernel. Combination of mean, variance and skewness features is performing well for all the test cases with the lowest EER equal to 0.044.

7.6.3 SVM Classifier with Sigmoid Kernel

Figure 7.13 shows the performance analysis of the SVM classifier with a Sigmoid kernel. The performance of this classifier, when trained with the combination of mean, variance, and skewness, is better when the number of the test samples from each participant is equal to 1. The EER, in this case, is 0.45 which is worse than the EER we obtained in the previous section. For a few tests cases the mean, variance, and skewness feature set is performing better by giving low EER and for few test cases feature set comprising mean, skewness and gamma distribution parameter is performing better.

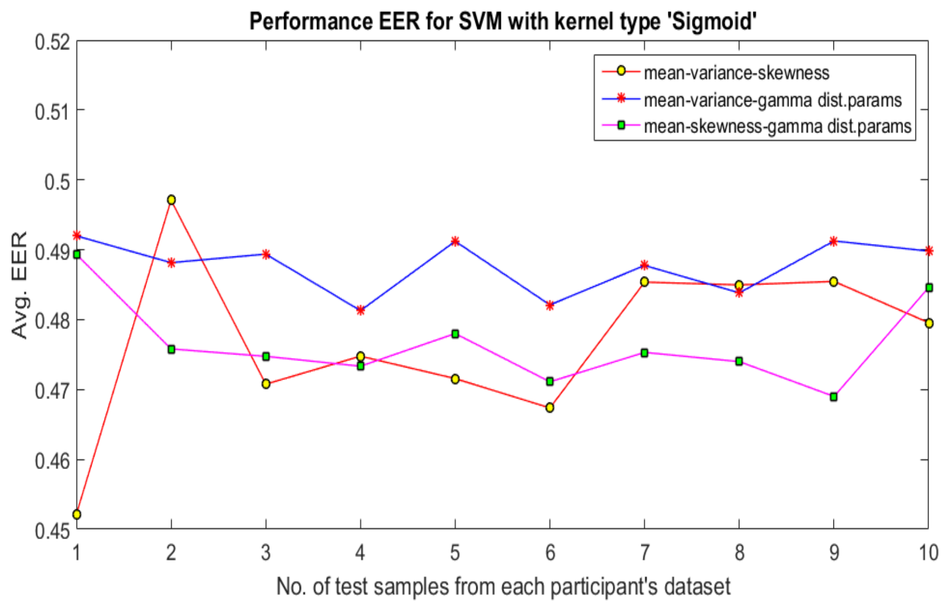


Figure 7.13: Performance of the system when trained with different 3 features array sets. Here the classifier is SVM with Sigmoid kernel. Combination of mean, variance and skewness features is performing well for all the test cases with the lowest EER equal to 0.45.

In summary of this section we trained our system with 3 features being mean, variance and skewness. From the above results, we can also conclude that the Gaussian kernel is performing better than the other two kernels. So, in the next we will vary the parameter of Gaussian kernel and analyze the performance when features are mean, variance and skewness for training the system.

7.7 Performance Using SVM with Gaussian Kernel and Different Parameters

In the last section, we concluded that 3 feature array of mean, variance and skewness yields the lowest EER. In this section, we are analyzing the predicted EER by selecting 75% of data for training and 25% of data for testing from the data set of each participant. Therefore, we have randomly selected 5 samples from the feature array for testing data set and remaining for training. In the last three sections, we observed that the SVM classifier with a Gaussian kernel is performing better than the kernel type ‘linear’ and 'sigmoid'. So, we are analyzing the performance of the SVM classifier with the Gaussian kernel by varying the gamma and cost function.

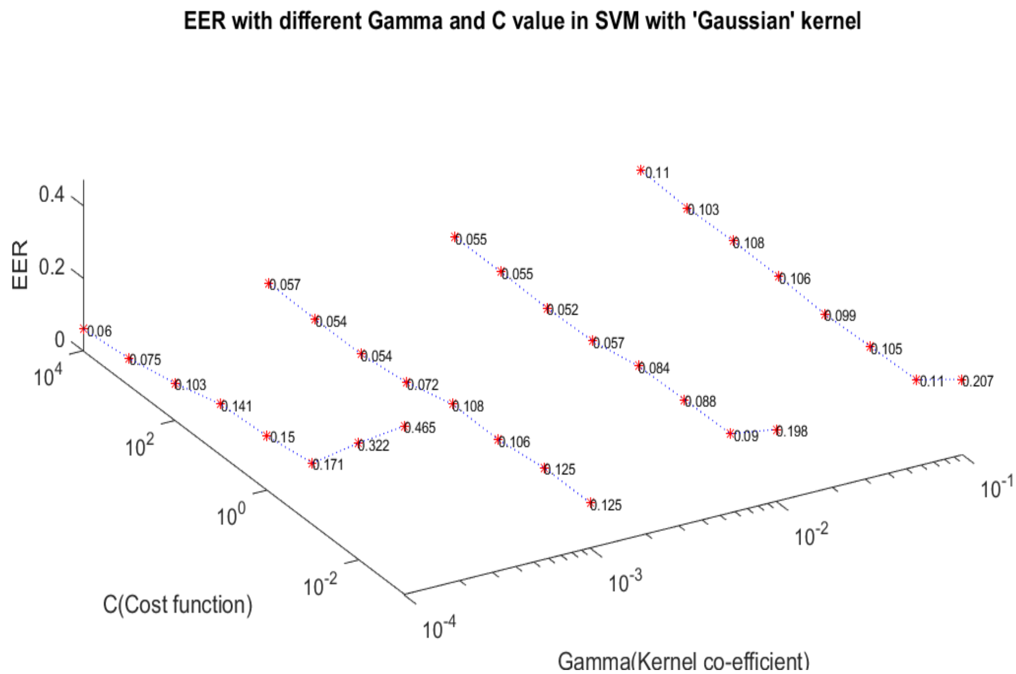


Figure 7.14: Predicted EER with different values of kernel co-efficient of Gaussian(rbf) kernel and cost function. The figure shows that the lowest EER is achieved when gamma is 0.01 and C=100.

From Figure 7.15 it can be observed that on varying the gamma and cost parameters of kernel function the value of EER is also varying. In this figure the X-axis represents Gamma which contains kernel co-efficient ranging from 0.0001 to 0.1. The Y-axis represents parameter C which is cost function, ranging from 0.001 to 10000. The Z-axis represents the predicted EER value when the system is trained with 75% of data from the feature array of each participant. When the value of Gamma=0.01 and C=100, the lowest EER 0.052 is predicted. The highest value of EER is 0.465 when Gamma=0.0001 and C=0.001.

7.8 Performance When Less Number of Participants

In this section, the performance of the system is being analyzed when it is trained with a fewer number of participants. The system is tested with 4 test cases: 5 participants, 10 participants, 15 participants and 20 participants. From Figure 7.15 we observe that lowest EER is when Gamma and C parameters are 0.01 and 100 respectively in the SVM with a Gaussian kernel. So, to check the performance with fewer participants we are using the same parameters in the SVM classifier.

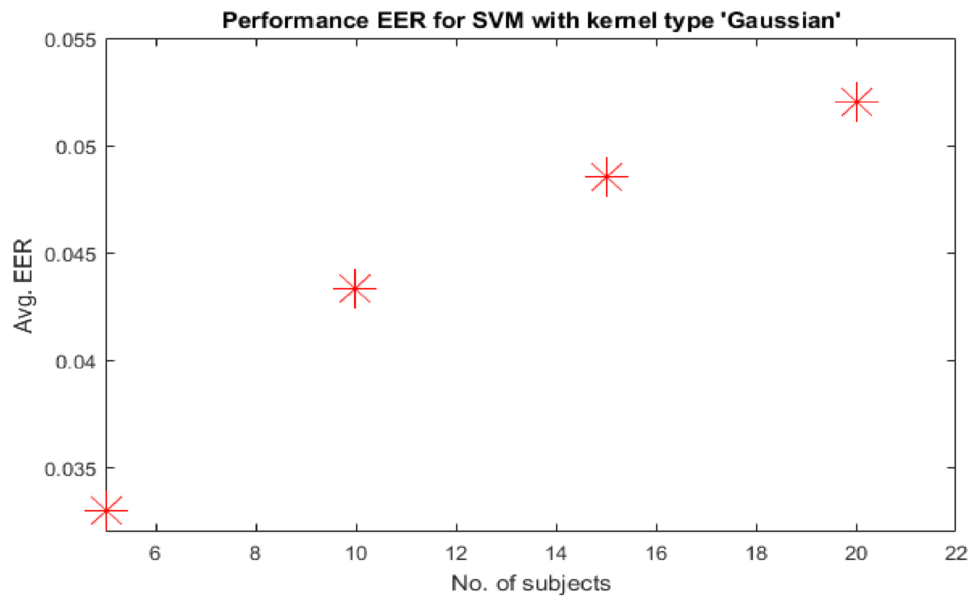


Figure 7.15: Predicted EER when the number of participants are varied. Here the classifier is SVM with Gaussian kernel. Parameters of Gaussian kernel set in this system: Gamma=0.01 and C=100.

Figure 7.15 shows that when the number of participants is 5, the system is identifying participants more accurately with an Avg. EER of 0.0483. As the number of participants is increased the avg. EER for 100 iterations is also increasing. When the number of participants is 15, avg. EER is 0.059.

To summarize, this section analyzed the performance of three classifiers by training the identification system with various feature combinations. It can be concluded that for this identification system, SVM with 'Gaussian' Kernel or SVM with 'rbf' kernel is performing best. When the system is trained with 75% of 3 feature array data set and the remaining 25% for testing of 20 participants, then the avg. EER for 100 iterations is 0.052. In the case of 5 participants with the same identification system, the lowest EER achieved is 0.048.

Chapter 8

Discussion

In this chapter, we present a discussion of the results obtained in this study and suggest future work to further improve the system.

In Section 7.4, we observed that when only the mean is considered as a feature the system performs better as compared to when the variance, skewness, and gamma distribution parameter as the feature with all the kernels. In Figure 7.5 it is observed that SVM with a Gaussian kernel is outperforming the linear and sigmoid kernels with the default parameters C=1, and gamma= 1/ (Number of features) in one feature array. Similarly, the low cross-validation error percentage of 5.48% is observed in the case of the mean feature when the classifier is SVM with a Gaussian kernel. The cross-validation is performed to see how well the model will perform with a real dataset.

In Section 7.5, we observed that when the number of features in the feature set is increased, the value of EER decreased. Therefore, two feature array in which the combination of mean with other features is used to train the SVM classifier with a Gaussian kernel which results in 0.058 EER when the number of test samples selected equals 5 from the dataset of each participant. This low EER performance can be seen in Figure 7.8, where the feature set comprises of mean, variance, and skewness, is performing and the predicted EER is the lowest for the same number of test samples as mentioned above.

After analyzing two feature array combinations, we analyzed the model with three feature combinations of mean, variance, gamma distribution parameters and skewness. We observed in Figure 7.12 that the combination of mean, variance and skewness is out performing other feature combinations with the SVM classifier with a Gaussian kernel. Similar results were observed in the cross-validation analysis. The cross-validation error percentage is lowest for mean, variance and skewness when analyzed with a SVM classifier with a Gaussian kernel.

Further on we analyzed the performance of the SVM classifier with a Gaussian kernel by varying the value of C and gamma (kernel co-efficient). In Figure7.15, we can see that on tweaking the value of Gamma and Cost function (C), the predicted EER is also varying. C is the regularization parameter. It chooses the margin in the hyperplane. If we choose a smaller C, then the optimizer in SVM will look for a larger margin hyperplane. If the value of C is larger, smaller - margin hyperplane is selected for classifying the training points correctly. When C is low, the variance is also low and the risk of under fitting increases. When C is high, the variance increases and chances of overfitting also become significant. In this study, we obtained the lowest EER when the value of C is 100 and gamma is 0.01. Finally, we varied the number of participants to examine the performance of the classifier with a fewer number of participants. As expected, with the lower number of participants the system yielded a lower EER, thus resulting in higher accuracy. The explanation for this unexpected high accuracy is as follows. If we increase the number of participants the chances some of them might receive calls and walk in similar way walking increases thus the accuracy would decrease. In this research, we are only using four sensors for data. In future work, we could utilize more sensors data like a Gyroscope Sensor, Magnetometer, pedometer, and heart rate sensor to derive more physical and behavioral characteristics of users. More data will give more information about the users which will increase the identification accuracy of users. We only derived mean, variance, and skewness as features for this study. Other features like wavelet decomposition and Fourier transform could be calculated to see whether these features will improve the performance or perhaps the results will be the same. Principal Component Analysis (PCA) which is a dimension reduction tool would also help us to see if the reduced dimensional features can predict, with the same or higher performance, than the one which was obtained in this study. Different Machine learning approaches such as Decision Tree, Neural networks, Nearest Neighbor Algorithm, and random forest can be used to identify Users. These algorithms could reduce the memory as well as computational cost of identification. Also in this work, we only studied three activities, phone receiving activity when it is on a table, in pocket or bag and walking. We could record other daily life activities like exercising, eating and typing which are unique to the users.

Chapter 9

Conclusion

With the advancement of technology, internet, and application development, people are using mobile phones for communication as well as other important activities. People are using smartphones for money transfer, documents sharing, photographs and other crucial information saved in the mobile device. This demands to increase the security of information stored on the phone. Today we have PIN passwords, pattern passwords and other types of user authentication system, but all are failing due to numerous factors such as smudge attack and common passwords. Many researchers are trying to utilize common human behaviors and physical characteristics to identify the mobile users. This thesis seeks to identify users by their hand gesture while receiving a call and during walking activity. The system is built using the smartwatch sensors for recording the activity of users. The challenge here was to identify users on the basis of only four sensors and activities. The four sensors values are used for feature extractions. We are performing the k-fold cross-validation where $k = 4$. We observed that the highest cross-validation accuracy when the system is trained with mean, variance, and skewness. We then train and test our model with the test data using the same features. The SVM classifier with a Gaussian kernel is used to identify the users with the lowest possible equal error rate of 0.06 for 20 participants with parameters: C=1, gamma= 1/(number of features). On varying C and gamma, we achieved 0.052 average EER when C= 100 and gamma=0.01. When the number of participants considered is five the average EER is 0.048. In the future, this system can be trained by more activities like walking, eating, exercising and other daily activities. Instead of using a SVM classifier, other classifiers like KNN and Hidden Markov model can be employed. The complexity of the system would increase with the growth in number of users and activities. In such scenarios, Neural Networks can be used to make more accurate predictions. In this work, we selected smartwatch instead of phone to collect data because smartwatch gives more personalized behavioral characteristics. For example, the way in which the user receives the phone, how often they move their hand while talking and doing other gestures.

References

[1] James Wayman, Anil Jain, Davide Maltoni, and Dario Maio. *An Introduction to Biometric Authentication Systems*. Springer London, 2005.

[2] David D. Zhang. *Biometric Solutions*. Springer US, 2002.

[3] Hugo Gamboa and Ana Fred. A behavioral biometric system based on human-computer interaction, 2004.

[4] G. Slagle. Standards for the drivers license.

[5] B. Wing. Overview of all ins biometrics projects.

[6] G. Levin. Real world, most demanding biometric system usage, February 2002.

[7] J. Markowitzn. Voice biometrics: speaker recognition applications and markets, 1999.

[8] J. Perkins. Ft-it: New services will keep eye on security: biometrics. financial times, 2001.

[9] Stephen Mayhew. History of biometrics, January 2015. Available: `http://www.biometricupdate.com/201501/history-of-biometrics`.

[10] Jun Rekimoto, Yuji Ayatsuka, and Michimune Kohno. Synctap: An interaction technique for mobile networking. In *In Proc. of Mobile HCI 2003*, pages 104–115. Springer, 2003.

[11] Ken Hinckley. Synchronous gestures for multiple persons and computers. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, pages 149–158, New York, NY, USA, 2003. ACM.

[12] Jonathan Lester, Blake Hannaford, and Gaetano Borriello. Are you with me? using accelerometers to determine if two devices are carried by the same person. In *In Proceedings of Second International Conference on Pervasive Computing (Pervasive 2004*, pages 33–50, 2004.

[13] Mohamed K. Shahin, Ahmed Badawi, and M. S. Kamel. Biometric authentication using fast correlation of hand vein patterns. *Cairo International Biomedical Engineering Conference*, Dec 2006.

[14] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *Information Forensics and Security, IEEE Transactions on*, 8(1):136–148, Jan 2013.

[15] L. Ballard, D. Lopresti, and F. Monrose. Forgery quality and its implications for behavioral biometric security. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(5):1107–1118, Oct 2007.

[16] Napa Sae-Bae, Kowsar Ahmed, Katherine Isbister, and Nasir Memon. Biometric-rich gestures: A novel approach to authentication on multi-touch devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 977–986, New York, NY, USA, 2012. ACM.

[17] Jiayang Liu, Zhen Wang, Lin Zhong, J. Wickramasuriya, and V. Vasudevan. uwave: Accelerometer-based personalized gesture recognition and its applications. In *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pages 1–9, March 2009.

[18] R. Mayrhofer and H. Gellersen. Shake well before use: Intuitive and secure pairing of mobile devices. *Mobile Computing, IEEE Transactions on*, 8(6):792–806, June 2009.

[19] Jiahui Wu, Pan Gang, Daqing Zhang, Guande Qi, and Shijian Lio. Gesture recognition with a 3-d accelerometer. In *Ubiquitous Intelligence and Computing*, pages 25–38. Springer, 2009.

[20] A. Akl and S. Valaee. Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation and compressive sensing. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 2270–2273, March 2010.

[21] Emmanuel Owusu, Jun Han, Sauvik Das, Adrian Perrig, and Joy Zhang. Accessory: Password inference using accelerometers on smartphones.

[22] Jun Han, E. Owusu, L.T. Nguyen, A. Perrig, and J. Zhang. Accomplice: Location inference using accelerometers on smartphones. In *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, pages 1–9, Jan 2012.

[23] Kaori Fujinami and Satoshi Kouchi. Recognizing a mobile phones storing position as a context of a device and a user. In *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pages 76–88. Springer, 2013.

[24] Lingjun Li, Xinxin Zhao, and Guoliang Xue. Unobservable re-authentication for smartphones. Available: `http://www.internetsociety.org/sites/default/files/02_1_0.pdf`.

[25] Jing Tian, Chengzhang Qu, Wenyuan Xu, and Song Wang. Kinwrite: Handwriting-based authentication using kinect.

[26] C. Guerra-Casanova, J.and Snchez-vila, G. Bailador, and A. de Santos Sierra. Authentication in mobile devices through hand gesture recognition. In *International Journal of Information Security*, volume 11, pages 65–83. Springer, Apr 2012.

[27] F. Okumura, A. Kubota, Y. Hatori, K. Matsuo, M. Hashimoto, and A. Koike. A study on biometric authentication based on arm sweep action with acceleration

sensor. In *Intelligent Signal Processing and Communications, 2006. ISPACS '06. International Symposium on*, pages 219–222, Dec 2006.

[28] Rahul Murmuria, Angelos Stavrou, Daniel Barbar, and Dan Fleck. Continuous authentication on mobile devices using power consumption, touch gestures and physical movement of users. In *Research in Attacks, Intrusions, and Defenses*, pages 405–424. Springer, Dec 2015.

[29] V.-M. Mantyla, J. Mantyjarvi, T. Seppanen, and E. Tuulari. Hand gesture recognition of a mobile device user. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 1, pages 281–284 vol.1, 2000.

[30] Tao Feng, Ziyi Liu, Kyeong-An Kwon, Weidong Shi, B. Carbunar, Yifei Jiang, and N. Nguyen. Continuous mobile authentication using touchscreen gestures. In *Homeland Security (HST), 2012 IEEE Conference on Technologies for*, pages 451–456, Nov 2012.

[31] Michael Sherman, Gradeigh Clark, Yulong Yang, Shridatt Sugrim, Arttu Modig, Janne Lindqvist, Antti Oulasvirta, and Teemu Roos. User-generated free-form gestures for authentication: Security and memorability. In *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '14, pages 176–189, New York, NY, USA, 2014. ACM.

[32] S. Li, A. Ashok, Y. Zhang, C. Xu, J. Lindqvist, and M. Gruteser. Whose move is it anyway? authenticating smart wearable devices using unique head movement patterns. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–9, March 2016.

[33] Alex Migicovsky, Zakir Durumeric, Jeff Ringenberg, and J. Alex Halderman. *Outsmarting Proctors with Smartwatches: A Case Study on Wearable Computing Security*, pages 89–96. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[34] Nicole Lee. Moto 360 review: It's the best android wear watch, but that isn't saying much, 2014. Available: `http://www.engadget.com/2014/09/12/moto-360-review/`.

[35] Phil Nickinsone. Moto 360 review: The first of the round android wear watches is stylish but imperfect. Available: `http://www.androidcentral.com/moto-360-review#hardware`.

[36] Teardown. Motorola moto 360 teardown. Available: `https://www.ifixit.com/Teardown/Motorola+Moto+360+Teardown/28891`.

[37] Android. Sensors overview. Available: `http://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-intro`.

[38] Eric W. Weisstein. Gamma distribution. Available: `http://mathworld.wolfram.com/GammaDistribution.html`.

[39] E. Boser Bernhard, M. Guyon Isabelle, and N. Vapnik Vladimir. A training algorithm for optimal margin classiers. Available: `http://w.svms.org/training/BOGV92.pdfl`.

[40] Csar R. Souza. Kernel functions for machine learning applications, March 2010. Available: http://crsouza.blogspot.com/2010/03/kernel-functions-for-machine-learning.html.

[41] Hsuan-Tien Lin and Chih-Jen Lin. A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods. Available: http://www.csie.ntu.edu.tw/~cjlin/papers/tanh.pdf.