Behavioural and Face Biometric Based Authentication for Mobile Applications

D. D. U. Kumarage

2016

Behavioural and Face Biometric Based Authentication for Mobile Applications

A dissertation submitted for the Degree of Master of Science in Information Security

D. D. U. Kumarage
University of Colombo School of Computing
2016

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge. it does not contain any material published or written by another person, except as acknowledged in the text.
Students Name: D. D. U. Kumarage

Signature: Date:03-02-2017

This is to certify that this thesis is based on the work of
Ms. D. D. U. Kumarage
under my supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

Certified by:

Main supervisor Name: Dr. S. P. Wimalaratne

Signature: Date: 03-02-2017

Abstract

The usage of smartphones and other mobile devices like tablets are gradually increased over the past decades. With the privilege of using smartphones, reliable user authentication on mobile devices has become a crucial requirement. Many authentication strategies have been proposed as alternatives to traditional password protection approach. Shape-based authentication methods like password pattern and biometric based authentication methods such as face biometric and fingerprint are some of the popular alternatives among them. However password pattern approach is considered to be the most popular authentication method which is accepted by many users.

In terms of security, password pattern is rather weak since the pattern can be easily stolen and reproduce. Thus, this project propose an implicit authentication approach that enhances the security of password pattern by adding additional security layer. the secondary layer will use a combination of face features and touch behaviours to identify the legitimate users. This study was conducted by using an Android application to collect data from user input on the touch screen. This data collection is done in two consecutive steps as face features and username is collected in the first step and touch behaviour in the second step.

Data analysis and user verification are done by using a statistical method which uses confidence interval calculated by using the sample mean and standard deviation to identify legitimate users. The final decision is made based on a pre-calculated threshold values which are 0.7 as the face threshold and 0.46 as the touch threshold value. Finally the this method could achieve 76.11% accuracy rate with 91.67% success rate, 8.33% false rejection rate and 39.44% false acceptance rate.

Acknowledgements

Dr. Prasad Wimalaratne, Senior Lecturer of University of Colombo School of Computing and my supervisor of the research project for the priceless help, guidance, and advice given throughout the research.

Dr. T. N. K. De Zoyza, Senior Lecturer University of Colombo School of Computing and coordinator of the MIS post-graduate degree program, for his encouragement and support to make this a reality.

To all the staff of the University of Colombo School of Computing, my dear lecturers, and teachers of school for the knowledge and advice provided to make me the person whom I am.

To all my friends especially Mrs. Inoshika Fernando Munasinhe, Mr. Avinesh Munasinhe, Mr. Dhanushka Sandaruwan Athukorala, Mrs. Teckla Dirukshi, all friends in MIS 2013 batch and my colleagues for their tremendous support, and encouragements provided from the beginning of this project in various ways.

To my dear parents who devoted their strength to strengthen our lives for everything they gave me to achieve not only this goal but also many things in my life.

Also to my dear brothers and sister and mother-in-law for the invaluable support provided me throughout this period.

Last but not least my dear husband to whom more credits should goes to; for being the shadow of this achievement.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

Worldwide mobile device users are rapidly increasing and recent statistics indicate that mobile devices especially smartphones are gradually replacing PCs and becoming a key computing platform [1] [2]. Nowadays, mobile devices, play a vital role as a tool for people on the internet. People use mobile devices to search information, store information, entertainment, social activities, e-learning activities and especially on-line business activities such as e-banking, e-commerce. With this technological advancement, people more rely on mobile devices which inevitably make them store more sensitive data on these devices.

As the demand of the mobile application rises, particularly smartphones, more security challenges are evolved. Mainly the user authentication has become more challenging requirement. Many studies on authenticate legitimate users and detect frauds on mobile devices has been conducted and identified that there is a high necessity of enhanced level of user authentication on a mobile device with relevant to future social, political and economic impact of the internet through mobile devices[2].

This project is focused on introducing secure multi factor authentication mechanism for mobile applications which integrates password pattern with user behavior and facial biometrics. Proposed authentication scheme is based on knowledge factors (password pattern) and inheritance factors (biometrics). This approach allows users to use highly usable and memorable password patterns for authentication and device will take care of the user validation through behavioral and facial biometric combinations.

1.1 Motivation

The degree of mobile application security depends on the trustworthiness of the user authentication. Knowledge-based authentication methods, such as traditional pin and password-based authentication methods found to be inadequate and feeble for securing current mobile devices and services [3] and many types of researches have been done in this area to introduce enhanced authentication methods.

With the Android OS, password pattern was introduced as a shape based authentication technique.The shape is consists of an arbitrary number of strokes which are drawn between nine dots. This allows the user to easily memorize the password [5]. However, these methods are vulnerable to various types of security threats or attacks, such as brute force attacks, shoulder surfing [4][9] and smudge attacks [23].

In order to overcome these weaknesses, biometric based authentication has been introduced as an alternative authentication method. The biometric technologies identify a person based on unique and immutable user characteristics [10]. These technologies use ”something user is” to identify user rather than just something user know. Biometric authentication related researches have been done mainly based on two areas as behavioral biometrics and physical biometrics. Physical biometrics is a relatively stable physical feature of a human body for example fingerprint, facial characteristics, iris pattern [11]. Behavioral biometrics is based on traits acquired from human behavior or habits. Signature, voice, touch dynamics, keystroke dynamics are fallen into behavioral characteristics. When comparing to knowledge-based authentication methods, biometric based authentications are considered to be more secure [26].

However, all of these technologies have drawbacks. There is a risk in fail sometimes for some people [12]. For an example, face recognition can be failed due to facial changes with age, weight, medical conditions and injuries. Moreover, if biometric information is compromised, it is not feasible to undo or revert back. Also user behavior may change according to the user condition and the environment.

The usability (user willingness) of the authentication mechanism also considered as an important factor. Some of the authentication methods that provides a stronger level of security obtain low ranks in terms of the usability [25]. Hence above factors influence to explore on new usable authentication scheme which is based on multi-factor authentication.

1.2 Significance of the research

Most of the mobile authentication related researches are based on only one biometric factor such as fingerprint authentication, face unlock, iris scan or user behavior. Resource limitations and user satisfactory factors may have an influence on this matter. This research is focused on design reliable authentication scheme by using two biometric types face and user behavior.

The significance of this project is, enhance security by adding additional security layer (multi-factor authentication) to the password pattern authentication method. Despite some weaknesses, password pattern is considered to be the most widely accepted and deployed authentication method. Therefore mobile users are already familiar with this process and touch dynamics and facial features will be acquired during user entering the password pattern which implies to have a high usability level than other biometric data acquisitions.

1.3 Goals & Objectives

The main goal of this project is to introduce novel authentication scheme based on multi-factor authentication for mobile applications. It extends the password pattern approach by adding two biometrics such as user touch behavioral biometrics and facial biometrics-based user validation as an additional security layer such that users are authenticated not only by the password pattern but also the touch gestures and user facial features.

The above goal will be accomplished by fulfilling the following objectives

1. Review the literature concerning the usage of touch behavior biometric and facial features to distinguish legitimate users.
2. Investigate the ability to use this two biometric validation on resource-constrained devices such as mobile devices.
3. Identify suitable classification technique.
4. Compare the result of proposed authentication method with existing research statistics to identify whether this approach has any improvement by means of accuracy and user acceptance.
5. Identify if any improvement is required to improve performance.

1.3.1 Research Questions

This project is based on following research question.

- What are the alternatives to improve mobile application security without losing user acceptance.
- Whether it is possible to use combination of face and touch dynamic biometrics to distinguish users.

Here we assume that the way user holding the device also can be considered as a unique characteristic.

1.4 Scope

The trustworthy authentication mechanism is very important for any computing device and application which handle critical tasks. However, with emerging trends, it is very clear that mobile devices have more priority than other computing devices in day to day life which introducing more security threats for mobile applications.

In this project, mobile platform was selected based on their market share which allows a greater accessibility to users. According to the current trend, Android mobile devices have the strongest user demand [27]. Also, there is a similar trend in selecting Android as a prototype development platform by researchers [26].

Hence this project is mainly focused on mobile application authentication. And the implementation of new concept will be done for Android mobile applications.

1.4.1 Design Concerns

Several points have identified that need to be considered when designing the proposed multifactor authentication method.

1.4.2 Computational Capability

The main concern on providing a solution for securing mobile authentication is the resource constraints. the mobile devices have less computational capability processing power and the limited storage. So typically the computation capability of mobile devices are lower compare to other computing devices like desktop computers. That implies algorithm complexity, communication cost, processing delay should be considered when designing the solution.

1.4.3 Energy Consumption

Minimize energy consumption is one of the other main challenging issues in mobile devices. Unlike desktop computers, mobile devices are operated by using battery power. So energy consumption should be considered when using various sensors embedded in the mobile device to extract data and when deciding the sampling rates.

1.4.4 Accuracy

The second layer of authentication in proposed authentication method is comprised of both touch dynamics and facial feature extraction. Facial features acquired in different occasions tend to vary due to surrounding light conditions and background noise levels. When compare to facial feature extraction, feature acquisition of touch dynamics is less sensitive to those environmental factors. However touch dynamic biometrics features also tend to change on different occasions due to various reasons like mood, distractions.

By considering above factors, to maximize the accuracy, the combination of touch dynamic biometric features and facial features have to be used for authentication.

1.5 Overview

In details structure of the thesis is as follows. The second chapter describes the research activities in the area of biometric-based user authentication undertaken in past few years. The third chapter describes the design and methodology used for the research. The fourth chapter will be in the implementation which includes data acquisition, preprocessing, feature selection decision making and data adoption approach. Project evaluation and result analysis will be included in the fifth chapter. Finally, the sixth chapter will conclude the research based on its behavior.

Chapter 2

Literature Review

2.1 Overview

With the technological advancement in the computing and communicational devices, the popularity of mobile devices has been increased with their network connectivity capabilities. A survey published on ”eMarkerter” website in the year 2014 was predicted that the number of smart-phone users worldwide will exceed 2 billion in 2016 and over half of mobile phone users will be switched to smartphones by 2018 [32]. Nowadays, it is clear that mobile devices are gradually replacing the desktop computers and have become a dominant computing platform in human life in many aspects including business, social networking, gaming, navigation, e-banking, and handling payment through Near Field Communication (NFC) technology etc. [14].

The increasing popularity of mobile devices make people rely more on mobile devices and it implies the increase of sensitive data stored on those devices. However, applications and data stored on mobile phones are found to be less protected from unauthorized access than a conventional desktop or laptop computers. Unfortunately, the portability of mobile devices also has a negative impact in terms of security. Because of the portability, they are more vulnerable to be stolen which may cause to sensitive data leakage or device misuse.

A significant number of research efforts can be seen in this area with different types of authentication strategies; especially from the last decade. Also, there are few surveys have been conducted to evaluate user’s security needs, their concerns and user awareness which reveal that there is an increasing user interest in perceived security and data protection[25][26].

If consider the history of user authentication of mobile devices, knowledge-based authentication methods like Personal Identification Number (PIN) or password pattern can be identified as the primary methods used to authenticate mobile users [33]. The introduction of the graphical password (password pattern) over text-based password is considered as one of the main turning point [15]. However, in security point of view, these techniques were not successful because they are vulnerable to a number of security threats including brute force attacks, spy on the password (shoulder surfing) and smudge attacks [4] [5].

As an alternative, researchers have proposed using user biometrics to identify the legitimate user against fraud users [16] [6] [13]. Biometric is described as a technique of recognizing individuals based on their unique and immutable characteristics. biometric-based user authentication is considered to be more secure than either PIN or password pattern based authentication because they are hard to be stolen or lost or forged. Especially they also provide the advantage that users are not required to remember their passwords [17].

Aside from those advantages, using biometrics also have some issues in being used as user authentication method in some cases. Jain et. al. [39] have conducted a research on this topic and they presented four features that describe ”What biological measurements qualify to be biometric”.

Universality Majority of the users should be able to use biometrics

Distinctiveness These user characteristics should be sufficient enough to verify user identities.

Permanence Characteristics should vary over time.

Collectability Should be able to measure characteristics quantitatively.

Those are the major factors that biometric measures should fulfill. *Performance*, *Acceptability*, and *Circumvention* are other concerns that encompass with accuracy, speed, resource requirement , user acceptability and the possibility to grant access to other fraudulent users.

biometric-based authentication is divided into two categories as physical biometrics and behavioral biometrics. Relatively stable features of a human body such as facial characteristics, iris pattern or fingerprint are used as physical biometrics. behavioral biometrics are traits that acquired from human behavior such as touch gestures (keystroke dynamics), voice and signature [26].

Physiological Biometrics

This section includes a survey of existing authentication methods used in mobile devices and discusses their advantages and disadvantages, security level and usability based on research worksand surveys conducted in this area. Also, this section will provide the study of data acquisition methods, feature selection, and decision-making technologies that are supposed to use in this project.

2.2 Mobile User Authentication

Traditional PIN and password pattern based authentication mechanisms which were introduced as primary authentication mechanisms to authenticate users on mobile devices are perceived as not adequate anymore because they are vulnerable to a number of security attacks such as shoulder surfing, brute force attack and smudge attack [26]. However, the sensitivity of data stored on devices and the context of use implies the requirement of another level of protection. Using biometric information for user authentication can be identified as one of the most promising approaches introduced in last decade. Recently, biometrics authentication has been receiving an extensive attention with increasing demands for personal identification and access control on mobile devices like smartphones.

Biometric authentication methods use human physical or behavioral characteristics to identify the legitimate user and thus having many advantages over other methods such as users are not required to remember their password, no token (written note) can be lost or stolen, and also biometric data are harder to crack [25]. These advantages has influenced leading market players in mobile devices like Android and Apple to move into this area. Based on their research works, Android released face unlock feature with their OS version 4.0 (Ice Cream Sandwich) [15] [17] and iPhone released fingerprint unlock feature (Touch ID) with 5S [18].

However, an effective authentication method should suit to the mobile device in terms of their resource constraints and should have a good user acceptability while providing a high level of security. Few surveys can be found in this area which was conducted based on the effectiveness of existing authentication methods. The survey conducted by H. Sieger et al. (2010) has presented a good analysis based on the effectiveness of different types of biometric authentication methods on mobile devices. In their research perceived security protection and the usability factors were considered as main criteria [25]. During the study, they have identified some biometric methods which are not suitable or cannot be implemented for mobile devices due to hardware constraints. Using palm print, hand vascular and hand or ear geometry for user recognition are among them. According to their research, there are some other authentication methods that provide high security but have less user acceptance. In their list of most secure authentication methods, iris and voice-based authentication methods have taken the top places but they have low ranks in terms of the usability [25].

Nedaa Zirjawi et. al. (2015) also have done a survey to analyze user perception of different biometric authentication techniques in terms of information security, data privacy, and user willingness to accept additional security features. Especially they have considered the impact of demographic factors like age and smartphone to user preferences. Figure 2.1 clearly illustrate the how users are concern about using their sensitive data to protect smartphone data.

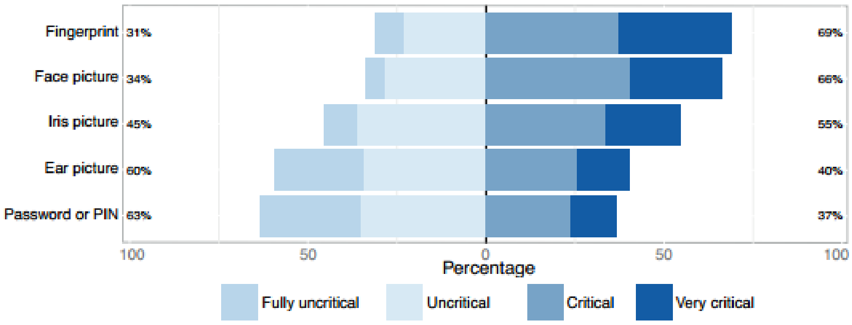


Figure 2.1: Criticality of user biometric information[36]

There is another survey which is done by De Luca et al.(2015)[34] also emphasise the same fact that usability is the most influencing factor to which biometric authentication method will be selected by the user. In their publication, they have pointed out two primary usability issues. One is **slow authentication speed and inconvenience** and other is **social awkwardness**. They have found that these factors are highly influenced to limit the participants adoption of a biometric-based authentication methods [25]. If consider face recognition approach, align face correctly in front of the camera can be difficult as well as awkward to do in a public area. In the case of fingerprint biometrics, it is hard to scan fingerprint if fingers were too oily or dry. Also using a device cover also affects the fingerprint reading. There is a great impact from such experiences to reduce the user willingness to use fingerprint biometrics as the authentication method [26].

Behavioral biometric authentication is the other type of biometric authentication. Though physical biometric approaches have more accuracy level than behavioral biometric approaches, behavioral biometric also have some advantages over physical biometrics. For example, once physical biometric such as iris or face biometrics are compromised, there will be no replacement; or the number of replacement can be limited if it is a fingerprint. But in an event of behavioral biometric like touch dynamics, if the touch dynamic template associated with one password pattern is compromised, then the new template can be regenerated with a new password pattern. Moreover, behavioral biometrics can be collected during the user's normal life activities which illuminate the social awkwardness.

When authenticating users by using behavioral biometrics, it is important to classifying the behavior. behavioral biometrics can be classified into five categories based on the collected user information types [13]. The first category is based on authorship based biometrics which examines the text or drawing user entered to the device. It observes the style peculiarities typical of the author such as vocabulary, punctuation or brush strokes. The second category is based on Human computer interaction (HCI). Different user strategies, styles are examined here. This category can be divided further into categories as human interaction with input devices (keyboard, mouse, and haptics) and advanced human behavioral biometrics (strategy, knowledge or skills exhibited by the user during interaction with software). The third category is monitoring indirect HCI based biometrics (system calls, registry access etc.). Fourth is **motor-skill** based biometrics (muscle utilization) and fifth is based on **pure biometrics**.

For mobile device authentication schemes, **authorship** based biometrics are used. Among the research works done for behavioral biometric-based user authentication, keystroke dynamics and touch dynamics have taken a significant amount of attention in last few years. With the technological advancement and arrival of the mobile touch screen, research interest has been diverted more into touch dynamics over the keystroke dynamics. Figure 2.2 illustrate the evolution of research works in the area of touch dynamics as analyzed by Shen Teh et. al (2016) in 2016 in their survey[26].As state in their survey touch dynamics has notable advantages than physical biometric in terms of user authentication on mobile devices. Basically, it can be implemented easily by using existing resources like sensors embedded into the mobile device. And the most important fact is a collection of touch dynamics is a non-intrusive process which does not add additional burden to the mobile user.

In terms of security, touch dynamics can be used in conjunction with the password pattern to gain an enhanced level of security. Here the way user performs the password pattern is monitored [4]. It includes touch screen data of current smartphones such as pressure, coordinates, size, speed, time etc. to distinguish the legitimate user from an attacker.

Despite drawbacks, password pattern is considered to be the most widely accepted and deployed authentication method. Therefore using touch dynamic combined with password pattern has high potential to obtain higher user acceptance than other biometric-based authentication methods.

By considering above factors, this research is mainly focused on using touch dynamics effectively for user authentication by combining with the password pattern and facial features. Therefore analysis on existing research efforts on the area of touch dynamics will be done here.

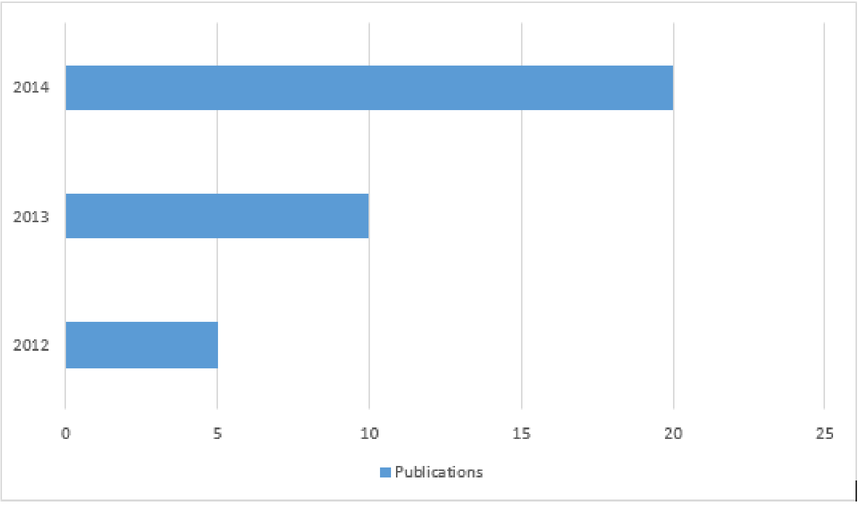


Figure 2.2: Emerging trend of research works on touch dynamics [26]

2.2.1 Touch Dynamics Biometrics

Touch dynamic is a behavioral biometric process of capturing and assessing the way of a person touches on a touch screen device. These touch information can be used to identify a user. Within this process, digital signature (template) will be generated by using touch dynamics upon the user registration and this signature is considered to be a unique identifier to identify individual users. To increase the strength of this signature, touch dynamic features can be used in conjunction with a password pattern.

However, most of the early research works were based on keystroke dynamics. The research was done by Gaines et al. in 1980 is one of the early research works done to authenticate individuals who seeking access to confidential/sensitive information stored in a computer. In their research, they used the timing between successive keystrokes [35]. With the rise of smartphones and tablets, researchers have shown more interest towards touch dynamics in recent past. Figure 2.3 illustrate the evaluation of touch dynamic related research works as captured by Shen Teh et. al. (2016).

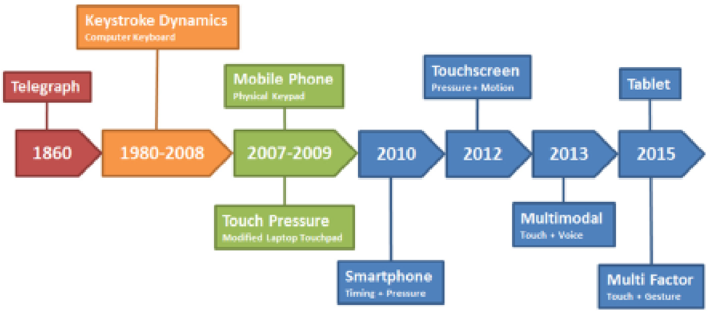


Figure 2.3: Evolution of touch dynamic biometric research works [26]

The following section will discuss the pros and cons of using touch dynamics for user authentication over other biometric methods.

Advantages

Human touch pattern on a touch screen contains unique features which are useful in **distinguishing users**. These unique features are extracted by processing the raw information collected via the device touch screen. In most of the research works spatial, motion and timing features such as pressure, coordinates, size, speed and elapsed time have been used for user authentication. Because of these multi-dimensional features are unique which means hard to replicate again and again, touch dynamics offers a high probability of identifying users accurately.

Beside the distinctiveness, it offers a number of advantages over other biometric authentication systems. The conjunction of touch dynamics with password pattern in smartphones provides **an advanced level of security**. Despite the vulnerabilities, password pattern has a higher user acceptance which makes it more popular among most of the smartphone users [12]

[13]. Combining these two will produce a user-friendly authentication mechanism with high assurance level which can authenticate users non-intrusively during the user login attempt. Most notably touch dynamic can be used for **authorization through constant monitoring** users touch patterns. This operation can be run during an active login session to deter other users to perform unauthorized operations [25][26].

With comparison to physical biometric approaches, there are few other advantages of touch dynamics features. In the case of compromising the touch dynamic template that used to authenticate the user, it is **possible to revoke** the touch dynamic template. In such case users just have to change their touch dynamic template by simply changing their password pattern. But finding an alternative for iris or face biometric is not possible. In contrast to physiological biometric data acquisition process, collective touch dynamics is **easy to implement** and also **cost effective**. It does not require special hardware such as hardware that uses for iris data collection or fingerprint data collection. Instead, it can use built-in sensors to capture touch dynamics.

In addition, due to data acquisition is done while user engaged in their normal routine, helps to **increase the usability** of touch dynamic based authentication methods over other physiological authentication methods. Because of it is using a touch pattern which is familiar to the majority of the smartphone users and there is **no social awkwardness** like doing iris authentication in a crowded place, can be identified as one other advantage of using touch dynamic for user authentication.

Disadvantages

As common for all the authentication methods, few disadvantages could be found in touch dynamic based user authentication as well. When referring to the research work done in this area, touch dynamic methods exhibits **lower accuracy** than physiological biometric methods. Unlike iris and fingerprint biometrics, there is a tendency to vary the touch biometric feature data due to external factors like mood or other distractions [26]. Therefore this need to be considered during design user authentication methods that using touch dynamic features.

In terms of the computational power, mobile devices have less computational capability than a conventional desktop or laptop computers. Therefore this will directly impact on the **algorithm complexity** which implicitly affects the security level and the reliability of the authentication methods. Also when using touch dynamic authentication methods on mobile devices, it is very important to consider the **energy consumption** of the particular application [26]. Because different types of sensors embedded into mobile devices are used to collect the user touch behavioral features. Thus the usage of these sensors, usage frequency, and the duration will have a direct impact on the energy consumption of the device [37].

Moreover, it is a well-known fact that human behavior does not steady as physiological features and it **can be changed over time**. For an example user's touch patterns typically change as they get more familiar with the device and the password pattern [26]. Therefore the touch dynamic based authentication methods should have the ability to adapt itself to the change of user touch pattern.

2.2.2 Facial Biometrics

Over the last two decades, many research attempts can be found in the area of user authentication based on face recognition. Face recognition play vital role not only user authentication but also in security and surveillance purposes. Thus currently face recognition has received a significant attention in fields like banking, law enforcement, e-learning.

The face recognition process is a combination of three different processes; face detection, feature extraction and face recognition. Various research worksare conducted in each of these stages. U. Bakshi et. al. [40] has done a comprehensive survey on face detection and feature extraction techniques which are integral and important part of face recognition. According to their survey, there are four types of methods are used for face feature extraction; template based, appearance based, color segment based and geometric-based methods. The template-based methods are identified as a simple method to implement. But as these methods do not represent whole structure of the face it could affect the accuracy level of the methods. The second one is appearance-based methods and it can be identified as an alternative for template-based methods as they use optimal feature points which represent global face structure. The third type is color segment based methods and those methods use the color model to detect the skin and feature with morphology operation. These methods are considered to be inefficient in the perspective of performance. Last is the geometric base methods like Gabor wavelet.It uses transform face feature extraction to extract stable and scale invariant features. Table 2.1 contains a summary of their survey [40].

Author	**Technique**	**Methods**	**Features**	**Advantages**	**Disadvantages**
T.Kanade,1997	Geometry based	Gabor wavelet method	Eyes, mouth, nose	Small database Recognition rate 95%	Large no.of features are used
A.Yuille, D. Cohen, P.Hallliman,1989	Template based	Deformable template	Eyes, mouth, nose, eyebrow	Recognition rate 100% Simple	Does not represent global face structure
C.Chang, T.S.Huang C.Novak,1994	Color based	Color based feature extraction	Eyes, mouth	Small database Simple manner	Performance is limited due to diversity of backgrounds
Y.Tian, T.Kanade, J.F.Cohn,2002	Appearance-based	PCA,ICA,LDA	Eyes, mouth	Small no. of features Recognition rate 98%	Need good quality image Large database required

Table 2.1: Face feature extraction techniques [40]

Also, literature could be found for some research efforts that used techniques like Eigenface technique genetic algorithm for reliable face detection. A set of eigenvectors that used for face detection is called as Eigenfaces. This technique was introduced by Sirovich and Kirby in 1987 [21]. Then Turk and Pentald used this technique in their research done in 1991 for face classification. Even though this approach had some issues still it is considered as a benchmark for newer approaches [20].

Other than that, some research workshad been conducted on use face shape context to compare faces [20][22]. Basically, facial recognition systems are mainly based on size, shape, and distances between landmarks like eyes, nose, jaw and cheekbones [12]. However, there are some extended research worksthat used wrinkles and moles to improve the reliability of the system.

With the rapid growth of the mobile usage, many research workshave been conducted one using face recognition in mobile applications [11]. For an example, Android released their ice cream sandwich version with face unlock feature. According to surveys that carried out in this area has found that using face unlock feature is not convenient sometimes [16] and unfortunately many security breaches had been reported in this area.

Advantages

If consider data acquisition in face recognition approach, it does not require expensive and high sensitive equipment as used for iris and retina identification. Instead, it can use the inexpensive fixed camera to obtain facial features. This is one of the main advantages of using face recognition over other methods. Also, face recognition is not distracted by background noises and other noise variances due to the device as in voice recognition approach.

Furthermore, another biometric technique which is based on fingerprint or hand has a high tendency of being unsuccessful or useless due to tissue damages such as bruised or cracks. So comparing to hands and finger-based biometric techniques, face recognition has less chance of being useless due to such damages. Additionally, the ability tp execute face recognition passively without distracting users is another advantage of face recognition. Therefore this technique is beneficial for security and surveillance purposes because face recognition can be done without any explicit user involvement since face images can be captured from a distance by a camera.

Disadvantages

However, any biometric method may present some drawbacks because they involve human and biological characteristics. According to literature, current face recognition methods perform well in controlled environments such as frontal face images are acquired with strict constraints as defined in face recognition standards. Because the human face is a three-dimensional object and it might affect to identify the true face due to facts like uneven illumination. Also, there are few other variables affect the face recognition performance including wearing glasses, different skin coloring and facial expression due to different moods [40].

In fact, facial feature extraction in the unconstrained environment is a time-consuming as well as challenging process due to arbitrary illumination, pose variations (size) and the orientation [40]. Therefore much research attention has been devoted in the area of facial feature extraction. There are a significant amount of research workscan be found on feature extraction and they have mainly focused on what type of features are used and their effectiveness [40].

So relying on one authentication factor not sufficient enough when providing reliable authentication scheme. The chief technologist of the Center for Democracy & Technology, Joseph Lorenzo Hall has mentioned that You cannot use a biometric as a primary authenticator, or you're going to have a bad time, and he recommend to use passwords as the first line of security, and then use biometrics as an additional factor in security [41].

2.3 Device Selection and Data Acquisition

Data acquisition is the first operation of user authentication process which acquiring raw dynamic data. These raw data are acquired repeatedly over a specific time interval. Device selection is one of the important factors in this process. Because different devices have a different set of sensors which can be used to collect various types of features. In the touch dynamic research domain, commercial off-the-shelf smartphones and digital tablets are commonly used for data acquisition [26]. As the focus of this project is on using touch dynamic features for user authentication, literature in the area of collecting touch dynamics on mobile devices by using the touch screen is explored here.

The data acquisition of the majority of the research works has been done by using smartphones [26]. As predicted in the survey published on ”eMarketer” website in the year 2014 number of smart-phone users worldwide is increasing rapidly [32]. Therefore high usage of smartphones for research works in touch dynamic domain can be due to this fact that larger population of mobile device users are using smartphones. But there are few exceptional literature can be found which are used tablets for their researches. The research done by Bond and Ahmed Awad (2015) is one of them.

The resources available in the device is one of the criteria that should be considered while selecting the device. As mentioned earlier different mobile devices are equipped with different resources. Modern devices come with advanced technologies and more processing capabilities which useful to run complex algorithms. Also, they are bundled with higher precision and resolution sensors, which can be used to capture features with higher quality.

Second criterian is the development platform. It is important to select a development platform which allowed to do modifications and also the availability of open source library functions that can be utilized for the feature extraction. In terms of **customizability**, Android will take the

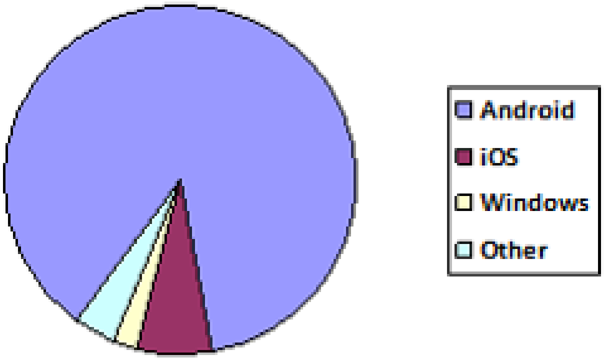


Figure 2.4: The distribution of underline research development platform [26]

high rank over another development platform. Because Android platform provides open source library functions [42]. The second factor is device platform need to be **flexible**, which means applications should be able to easily install on the mobile device without publishing it to a mobile application store. Both Android and Windows platforms provide this flexibility [26]. Furthermore, both Android and Windows device platforms provide direct access to system and data files without installing any third party applications or configuration changes. The third factor that should be considered when selecting the development platform is the cost of available development tools. In the survey done by Pin Shen Teh et. al (2016), they have done a good analysis of these factors. Table 2.2 shows the result of their survey.

	Android	iOS	Windows
Development Tool	Free	$99/year	Free
Publishing Fee	$25 one-off	No	Yes
Side Loading	Yes	No	Yes
Programming Language	Java	Swift	C# or VB
Open Source	Yes	No	No
Cross Platform Developments	Yes	No	No
File System Visibility	Yes	No	Yes

Table 2.2: Comparison of Different Development Platforms

Moreover selecting the mobile platform that popular among the large population (large market share) also very important. Because if the market share is higher, it will helpful to obtain a higher user acceptance. According to the current trend, Android mobile devices have the strongest user demand [27]. Therefore Android platform has been selected in many research worksto implement the prototype development [26] as shown in Figure 2.4.

Based on the literature survey done by Pin Shen Teh et. al. (2016), research workshas used two approaches to select the device to use for the experiment. One is to use a predetermined device which is decided by the researcher or researchers who doing the research and another approach is to use the user's own device for the experiment. Among these two approaches, the first approach is considered to be the most popular approach among research activities. According to Shen Teh et. al. it is around 94% percent [26].

In literature, two main reasons could be found for selecting predetermined device for data acquisition. One is using the same device to acquire user data helps to inconsistencies due to different behaviors of different devices. For example, availability of sensors may vary from one device to other and also there is a possibility to have variations in sensor sensitivity or the resolution [43]. The second fact is that users are much familiar with their own devices. Therefore there is a possibility to introduce a bias in the experimental results [26].

However, there are some different perspectives can be found in the literature. Some experiments have used user's own device to collect data as it is helpful to reach a large set of population groups. Because of data acquisition is done by using user's own device via a mobile application, it is not required them to physically present in the data acquisition process. Some of the research works have emerged the fact that the usage of user's own device helps to obtain an experimental result which reflects their actual usage behavior [26]. The reason given by them is that user's behavior while using their own device can be varied when performing a task on another device in the controlled environment. Also, their concern was collecting data under a controlled environment may produce an over-optimistic result compared to researched done in an uncontrolled environment.

But the research works who opposed with using uncontrolled environment has raised the fact that there is a risk in using uncontrolled experimental settings. Since it may lead to getting tampered or changed data which affect the quality of the experiment. Because main reason for conduct experiment under controlled setting is to reduce the effect of external factors such as distractions, sickness, cognitive load, etc. to acquired data. Thus the majority of research works have used controlled and supervised environments.

2.4 Data Pre-Processing and Feature Extraction

The data preprocessing will be done to remove outliers in the raw data. In the research done by Zheng et. al. (2014), They have employed an outlier removal process to improve the accuracy and the performance since they found that occasionally some participants fail to perform smooth tapping intentionally or unintentionally [45].

Feature extraction is the main task of behavior modeling component. The goal of feature extraction process is to come up with a specific representation of the data that can highlight relevant information. This process is carried out in both user registration and user authentication to identify and extract distinctive features which are common to a user by using acquired raw data. Then these features are used for template generation.

2.4.1 Touch Dynamic Feature Extraction

In touch dynamic feature extraction process, extracted features are mainly belonging to three high-level categories such as spatial, motion and timing features. Research use a different number of features which goes under those categories to generate the user authentication template. For example, the research done by Yuxin Meng et. al.(2013) have extracted 21 different features to construct the authentication signature. It includes average touch movement speed per direction (8), a fraction of touch movements per direction (8), average single-touch time, average multitouch time,the number of touch movements per session, the number of single-touch events per session, and the number of multitouch events per session [42]. This feature extraction is different for each research. Zheng et. al. (2014) has collected feature data of timestamps, acceleration, angular acceleration, touched-size and pressure [25]. In the research done by Wajeeh et. al. (2015) have used only three features to distinguish user's behavior representation. Those are finger pressure, the area of finger pressure and timing [46]. Here the time at each node on the pattern has been taken as the sequence of time information.

There was some other research work could be found in the area that used gravitational information and velocity in addition to pressure, time and touch area. Research done by Antal et. al. (2016) could be considered as an example. In their research work, they have extracted 11 features to generate the template. It includes duration, the length of the segment defined by the two endpoints, average velocity acceleration at the start, mid stroke pressure, mid stroke finger area, mean pressure, mean finger area, mean gravity x, mean gravity y and mean gravity z.

2.4.2 Facial Feature Extraction

In face recognition, face image representation is categorized into three main categories. First is appearance based (holistic) where the face is represented by a high dimensional vector containing pixel values. Second is feature-based where each vector summarizes the underlying content. And third is the hybrid approach [48].

Holistic or appearance based approach is considered to be the most typical to be used in face recognition. As it is using lexicographic ordering of pixel values to produce one vector per image, image is represented as point in a high-dimensional vector space. Therefore this dimensionality is equal to the size of the image in terms of pixels which is a large value. This is considered as the major drawback of this approach [40]. Also in literature some research worksshown that some of local facial features did not vary according to the user pose, lighting, and facial expressions and they suggest have suggest to break image into smaller sub-images to do feature extraction more comprehensive and effective way. Gabor wavelet is one of the most commonly used local feature extraction techniques [48] [40].

2.5 Data Classification and Decision Making

Data classification is one of the major processes in most of the biometric-based user authentication methods. In this process extracted feature data will be analysed and categorised into classes and compared them against reference templates. Here scoring value will be used and decision making is done based on that score value [26]. Different types of machine learning techniques have been used in literature for data classification and decision making, namely (i) Cluster Analysis , (ii) Decision Tree, (iii) Neural Networks, (iv) Support Vector Machine, (v) Statistical , (vi) Distance Measure and (vii) Probabilistic Modelling.

2.5.1 Cluster Analysis

The goal of cluster analysis classification technique is to group sample with similar properties to homogeneous cluster. Here it is assume that samples belonging to same subject have similar properties [47]. There are variant techniques can be found in the area of cluster analysis. K-mean and K-Nearest Neighbour algorithms are polar among them [26] [34] [37] [44].

2.5.2 Decision Tree

The decision tree technique is known for its low computational complexity [47]. Thus this technique has grab the attention of many research works. The J48 and the Random Forest (RF) can be identified as the most widely used decision tree techniques in touch dynamic based user authentication research domain [42] [26]. The main objective of decision tree technique is to create a tree-like model which used to predict the class of given test sample. Therefore this technique is particularly suitable for classification problems that has small number of output labels [26].

2.5.3 Support Vector Machine

The support vector machine technique is other machine learning technique commonly used for biometric-based research works. In support vector machine technique, first it determines how two classes of features differ from each other and then it will define a boundary to separate them. Then subsequent test samples are classified according to the boundary [47]. However it is not practical to divide extracted user touch features into two classes by using a linear boundary. Therefore more complex boundaries should be created to map feature data onto a higher dimensional feature space [47] [26].

2.5.4 Neural Network

Artificial Neural network technique is used to simulate the information processing structure of biological neurons [49]. The neural network architecture is consists of three layers namely input layer, hidden layer and output layer. User extracted feature data are fed into input layer.

This layer will assign weights to each neurons and passed within the hidden layer until output is produced. Then learning process will be executed to update weights of each neuron in the hidden layer to improve performance [49] [26]. This technique is considered to be the technique that produce output with more accuracy. But in terms of computational power it is expensive than other techniques [42] [47] also according to the survey done by Shen T eh. et. al. it is impractical to use in mobile devices with a memory less than 512MB [26].

2.5.5 Distance Measure

This is a scoring technique. It calculates the dissimilarity or similarity score between test sample and the training sample for a given subject. Then decision will be made based on the threshold value whether test sample is belong to the target subject. Most frequently used distance measure techniques are Euclidean and Manhattan distance [37] [26].

2.5.6 Statistical Techniques

Using statistical techniques like mean, standard deviation and deviation tolerance to identify legitimate users is one other popular method in the biometric-based user authentication domain [45][26]. Comparatively this method is considered to be a very important method for resource limited mobile devices because it is less complex and easier to implement than other methods.

2.6 Evaluation

This section provides an overview of the evaluation criteria/performance metric of biometric-based user authentication related research works. According to the literature, there is no widely accepted method for reporting the results [37]. Most of the study reports are consists of array of error rates and performance curves which make it hard to compare performance and eventually leads for drawing incorrect conclusions.

Moreover, the output of the pattern classifier may vary due to many factors such as chosen algorithm, training data set, chosen feature set and within-participant variations [37]. Thus the performance metrics calculated for different classifiers may affect. Next subsection will discuss the some widely used performance metrics that can be found in current literature.

2.6.1 Evaluation Criteria

Behavioral biometric-based authentication systems used two authentication methods such as verification and identification. The verification method is used to verify the claimed identity while identification method is used to classify and identify unknown identity. When assessing biometric-based authentication methods, three major criteria are used to evaluate the system.

- Accuracy

	Predicted Class		
		Positive	**Negative**
Actual Class	**Positive**	True Accept	False Reject
	Negative	False Accept	True Reject

Table 2.3: Confusion metrics for two class decision problem [37]

- Efficiency
- Usability

Accuracy

Based on the literature and the survey done by S. Teh et. al. (2016); Table 2.3 shows the different types of metrics that are commonly used for any classifier to evaluate the accuracy.

There are a different type of error rates are reported in the biometric-based user authentication research domain. Many studies report both false acceptance and rejection rates but do not consider much about true acceptance rate and true rejection rate. Thus there is some confusion can be seen in the literature. But following could be considered as the generally accepted metrics.

False Rejection Rate (FRR) The percentage ratio of the number of legitimate users who are falsely rejected over the total number of legitimate user tries are considered as the False Rejection Rate (FRR).

$FRR = \left(\frac{FalselyRejectedLegitimateUsers}{TotalLegitimateUserTries}\right) 100\%$

False Acceptance Rate (FAR) The percentage ratio of the number of falsely accepted illegitimate users over the total number of illegitimate user tries are considered as the False Acceptance Rate (FAR).

$FAR = \left(\frac{FalselyAcceptedIllegitimateUsers}{TotalIllegitimateUserTries}\right) 100\%$

Equal Error Rate (EER) This Equal Error Rate (ERR) value is obtained by finding the interception point of the FRR and FAR graphs as illustrate in Figure 2.5 [26]. Basically, ERR is considered as the single number performance metric used to measure and compare the accuracy level of different authentication methods. The accuracy is inversely proportional to the ERR which means when lower the ERR, accuracy will be higher. In order to obtain lower ERR value, both FRR and FAR values should have lower values. But as FRR and FAR are negatively correlated values, practically it is not possible to lower both FAR and FRR. Instead FAR and FRR can be adjusted based on the usability and security requirements as it meets lower ERR value.

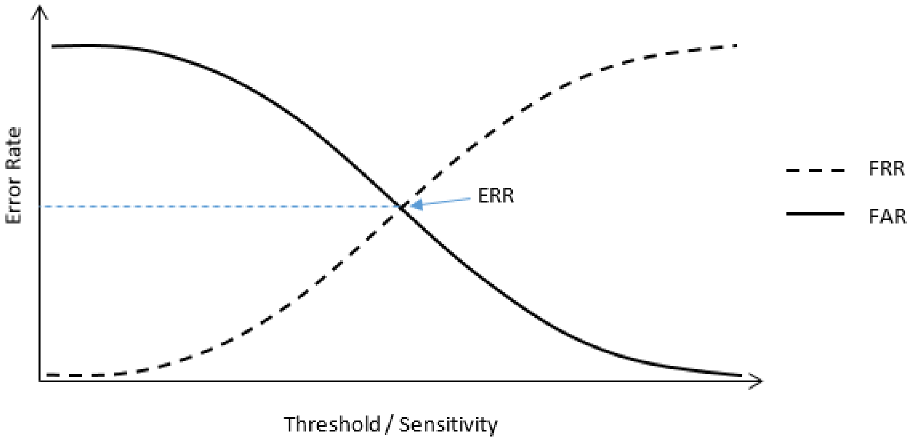


Figure 2.5: Equal Error Rate [26]

Above are the most commonly used metrics, but there are few other accepted metrics available in the literature.

Receiver Operating Characteristics (ROC) The ROC Curve is used to see the relationship between False Acceptance Rate (FAR) and Genuine Acceptance Rate (GAR) which is also known as True Acceptance Rate (TAR). The Genuine Acceptance Rate is the inverse of FRR and calculated by taking the percentage ratio of the correctly accepted legitimate users against the total number of legitimate user tries. Then accuracy is measured by using the graph that plot GAR against FAR for different matching threshold values (Figure 2.6) [26].

$GAR = \left(\frac{CorrectlyAcceptedLegitimateUsers}{TotalLegitimateUserTries}\right) 100\%$

The ROC curve is used to measure the overall usefulness of the results of the pattern classification. When the line comes closer t the upper right corner of the graph, the accuracy of identifying or verifying users getting higher. Moreover, as this curve is based on a threshold, it is useful for selecting a viable threshold to get more accurate results.

Area Under Curve (AUC) This measures the area under the ROC curve for given authentication system and a given user [37]. This value represents the probability of a true response either positive or negative. According to the current literature, the random classifier will have AUC value around 0.5 while AUC value for the ideal classifier is 1.0 (100%). This AUC value calculation is known to be not accurate 100% because it looses some of the information due to it lose the trade-off values that make up the curve.

Crude Accuracy (CA) Crude Accuracy is also known as the misclassification error. It is calculated based on the number of incorrect classifications made when comparing to the classification output where the actual class is a known fact.

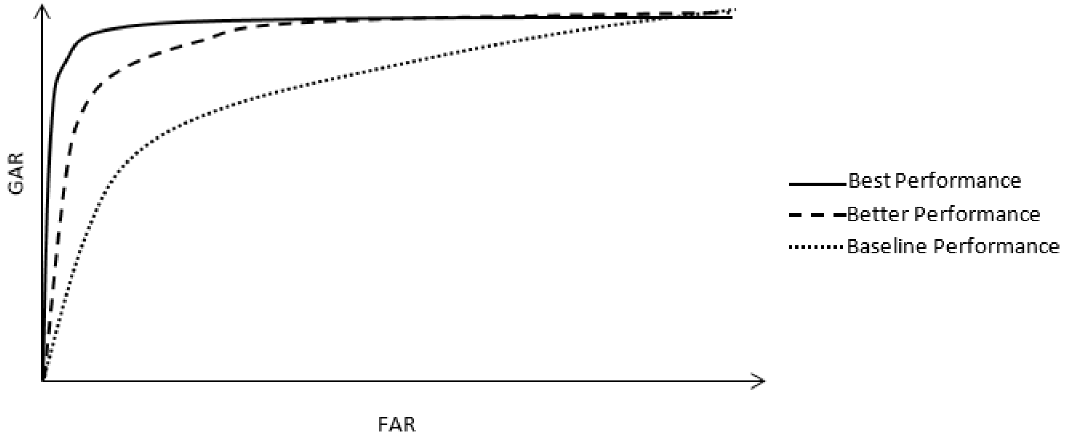


Figure 2.6: Equal Error Rate [26]

2.6.2 Behavioral Biometric Based System Evaluation

According to the standard published by the European Standard for Access Control Systems (EN 50133-1) for biometric-based user authentication systems, FAR must be less than 0.001% and FAR must be less than 1% in order to be used in production systems [37]. However, these specifications are more into physical biometric-based authentication systems and it is considered as not much suitable as a benchmark for behavioral biometrics because behavioral biometrics are known to be less distinctive than physical biometrics. Thus the error rates of related work in the particular field is used as a benchmark.

Here, the focus is on the research works that have used static mode for user authentication where the user is authenticated at the beginning of the login session or at some predefined intervals during the session. This is the most commonly used user authentication mode in mobile devices. According to the survey done by S. Teh et. al (2016), character-based and digit-based passcodes have been identified as the most used input types in static authentication mode. Table 2.4 shows the summary of their study on performance results of related research works.

According to the above table, S. Teh et al. (2016) have only considered about research works which have used either character input string, digits or symbols. But there are few other research works could be found in this are that used a password pattern as the input type. Angulo et al. (2012) have done a research based on two-factor authentication to enhance the smart device's security by adding biometric information to lock pattern. They were able to achieve 10.39% ERR by using Random Forest Classifier (RFC)[51][46]. The sample size they have used in their analysis is 32.

The research published by Luca et al (2012) is another example for using touch dynamic biometrics for improving security of password pattern. The Dynamic Time Warping (DTW) Technique has been used to analyze user input and they were able to achieve 77% accuracy

Study	Subject Size	Passcode Type	Input Length	Features	Method	EER	FRR	FAR	Accuracy
Dhage et. al (2015)	15	Character	10	Time	Statistical Techniques (ST)	0.806			
Trojahn et al (2013)	16	Character	11	Time,Spacial	Decision Tree (DT)		2.67	2.03	
Kambourakis et all. (2014)	20	Character	10	Time,Spacial	DT	26			
Giuffrida et al. (2014)	20	Character	8-9	Time,Motion	Distance Measure (DM)	0.08			
Bond and Ahmed Awad (2015)	25	Character	34	Time	Neural Network (NN)	9.3			
Buschek et al. (2015)	28	Character	68	Time,Spacial	Probabilistic Modelling (PM)	21.02			
Huang et al.(2015)	40	Character	11	Time	ST	7.5			
Antal and Szabo (2014)	42	Character	10	Time,Spacial	DM	12.9			
Sen and Muralidharan (2014)	10	Digit	4	Time,Spacial	NN	15.2			
Jain et al. (2014)	30	Digit	10	Time,Spacial	Support Vector Machine (SVM)	2.8			
Ho (2013)	55	Digit	4	Time,Spacial, Motion	SVM		5.3	4.4	
De Mendizabal-Vazquez et al(2014)	80	Digit	4	Time,Spacial, Motion	DM	20			
Zheng et al. (2014)	80	Digit	4	Time,Spacial, Motion	ST	3.65			
Tasia et al. (2014)	100	Digit	4-10	Time, Spacial	ST	8.4			
Wu and Chen (2015)	100	Digit	8	Time,Spacial, Motion	SVM	0.556			
Trojahn et al. (2013)	152	Digit	17	Time, Spacial	Cluster Analysis (CA)		4.59	4.19	
Jeanjaitrong and Bhattarakosol (2013)	10	Symbolic	4	Time, Spacial	NN				82.18

Table 2.4: Performance Result Summary

Pattern	FAR	FRR	Accuracy%
1	0.18	0.135	85.5
2	0.18	0.08	90.9
3	0.136	0.07	92.3

Table 2.5: Measurements of UA-UPTD Performance [46]

rate with 19% FRR and FAR. Waheeh et al.(2015) also attempted to enhance the security of user password pattern by using Singular Value Decomposition (SVD) algorithm to reveal basis vector for authorized users. Analysis has been done using three different patterns with different lengths and shapes. Table 2.5 shows the performance measurements of their approach (User Authentication based on Unlock Pattern Touch Dynamics UA-UPTD).

Efficiency

Efficiency is one of the three criteria to evaluate authentication systems in early research works. The efficiency refers to resource utilization and the performance of the system. This is very important for the context of resource-limited mobile devices. Because if the authentication process imposes a higher level of computational overhead or having considerable authentication delay then it affects the user acceptance.

Usability

The last criteria to evaluate authentication system is the usability. Users are reluctant to use any system that is slow or disrupt their normal activities even if it provides a higher level of security. Therefore the user acceptance or usability is a very important factor. So if the system has a low user intervention and has no authentication delay, then it can be considered as an acceptable authentication mechanism.

2.7 Summary

This literature survey is done for user authentication on mobile devices. Pattern password and face unlock are main authentication methods introduced for smartphones recently. However, there are only a few research works on this area. This research focuses on introduce multi-factor authentication mechanism to offer added layer of security. Based on the literature survey, summary of advantages and disadvantages of existing techniques are listed in Table 2.6.

	Advantages	**Disadvantages**
Password Pattern	Easy to memorize	Vulnerable to various types of security threats or attacks, such as brute force attacks, shoulder surfing, and smudge attacks
Physical Biometrics	Based on unique and immutable user characteristics, cannot social engineered or shared	Risk in fail sometimes for some people, Facial changes with age, Medical conditions, Injuries, Environment and user condition, Must be able to accommodate changes over time
	No need to remember the password. Always available to the individual,	If biometric information is compromised, it is not feasible to undo or revert back.
	High degree of confidence in user identity	Depends on the collected sample.
Behavioral Biometrics	No additional cost on hardware devices & relatively inexpensive to implement.	High non-matching rating
	High degree of social and legal acceptability	Vulnerable to mocking

Table 2.6: Pros and Cons of Existing Authentication Methods

Next chapter will discuss the methodology which includes data collection processing and analysis.

Chapter 3

Design

3.1 Overview

This project proposes a multi-factor authentication scheme based on password pattern and behavioral and facial biometrics. Since this is designed for mobile devices, resource utilization and perform authentication without utilizing additional hardware are highly considered here.

The design of authentication system is to accept four main input types for user authentication, namely

1. Username
2. Password pattern
3. Touch Dynamics
4. Features in user's face

The operation of the authentication system can be captured mainly in three major phases.

1. User registration, where sample user data acquired, pre-processed and stored.
2. User authentication, where acquired legitimate user data compare against sample data stored into files to determine the similarity or dissimilarity.
3. Data retraining, where stored data is updated to reflect any changes in the latest user data.

These three phases are accomplished by three main modules. Their functionality is described below.

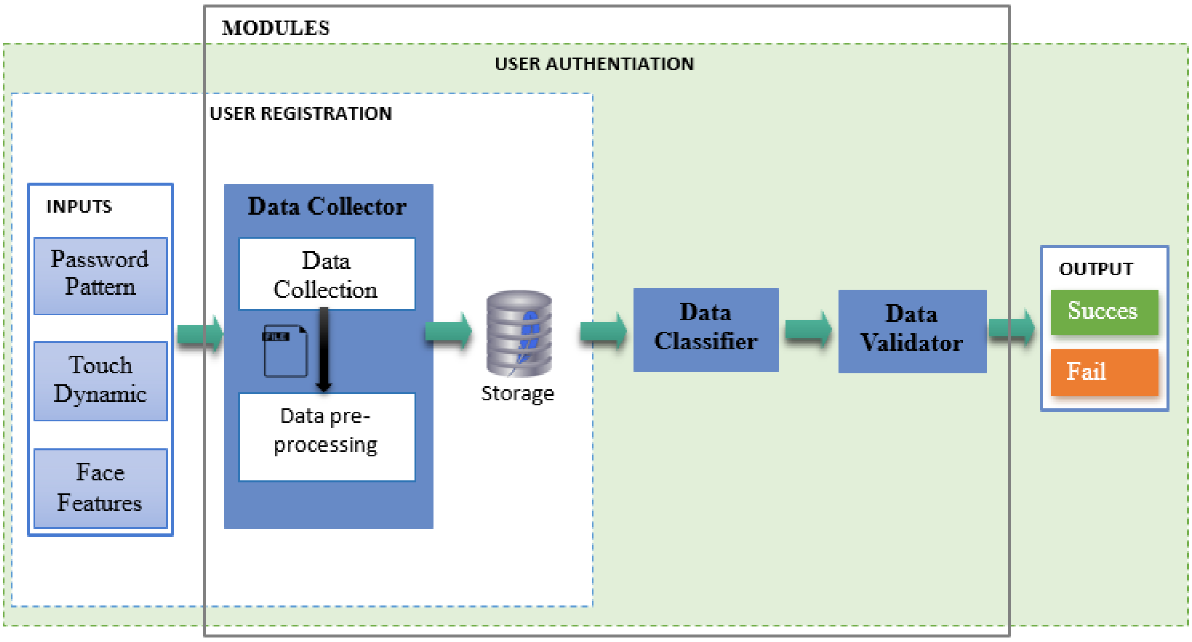


Figure 3.1: Architecture of proposed authentication system

3.2 Design

As described in previous sections this is an approach to introduce multi-factor authentication service for Android mobile applications. The main goal of this project is to authenticate a user based on face biometrics and user touch behavior on a mobile device.

Here the new approach is designed with three main components to collect information, classify information and finally validate and identify the legitimate user. So in order to identify legitimate users, the application should have prior knowledge about identifying legitimate users. Therefore, both training and recognizing phases are performed on the device. The main components of the system are illustrated in Figure 3.1

This methodology uses both facial biometrics and behavioral biometrics to distinguish users. So the Data Collector/Input Reader module will handle collecting user face features as well as user touch dynamics. This data collection is performed in two different stages. In the first stage, it will collect use face features while user entering his/her username. This data collection process is run as a background process which doesn't give any additional burden to the user. More importantly as mentioned earlier user face feature collection is done during the time user enter their username. The main reason for this is that user will focus on the phone screen while typing something on the screen.

The next stage is to capture user touch dynamics. This information is captured during user entering his/her unlock pattern. The way use holding the smartphone and how they touch the screen will be extracted using built-in sensors.

As illustrate in Figure 3.1, Data Collection module is a part of both user registration and user authentication scenarios. Activity diagram in Figure 3.2 brief the activity flow of the system. In user registration, data collection is a repetitive process where user face and touch data are collected multiple times. This will be done to increase the accuracy level and it is the normal behavior of any biometric based system. After collecting user biometric information, that information will be stored into files to be processed by data pre-processing unit including feature extractor.

The data pre-processing unit is a part of the Data Collector module.The purpose of this unit is to extract important features from collected raw data. This unit will be discussed in details in section 3.3.2.

The classifier is one other main module in this system which will do user classification based on extracted features and generate user signatures to be used by the validator later. Next module is the validator. Even though classifier and validator are considered as two different modules, these two modules work together to identify legitimate users by using their signatures.

3.3 Data Collector

The Data Collector module can be considered as the input reader of the system. This module is responsible for collecting raw data from the touch screen, front camera and some other built-in sensors like accelerometer and extract important features from raw data which are useful for user classification.

This module comprises of two processing units which are run in two stages. The first stage in data collector module is data acquisition. Within data acquisition stage, four types of data will be acquired. Those are;

1. Username
2. Face biometrics
3. Touch pattern
4. Touch behavior biometrics

Figure 3.3 illustrates the data flow within the data acquisition unit. The second stage is data pre-processing. The data pre-processing unit contains feature extractor to extract meaningful and important features from collected raw data. These extracted features are stored in an SQLite DB.

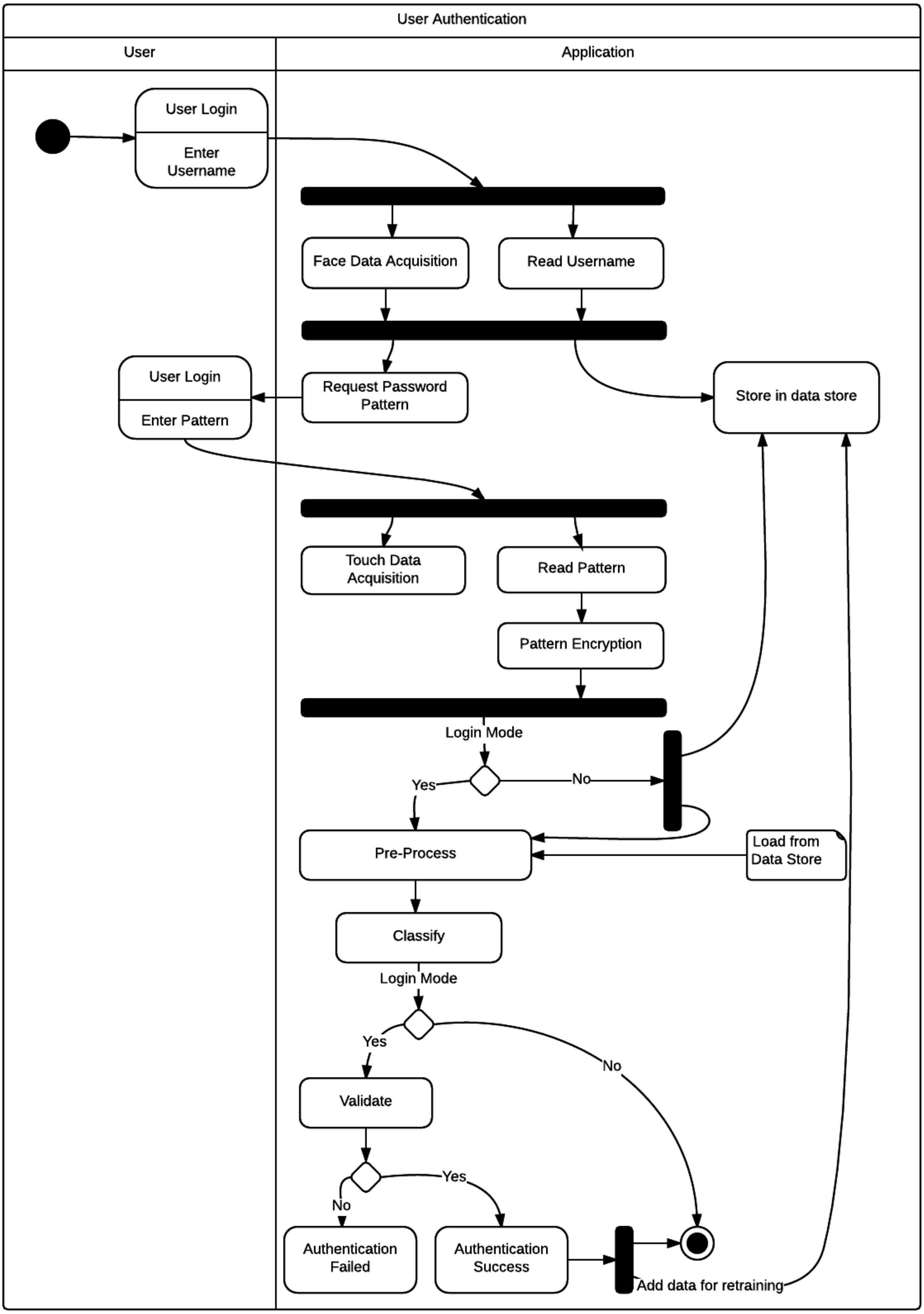


Figure 3.2: Activity flow

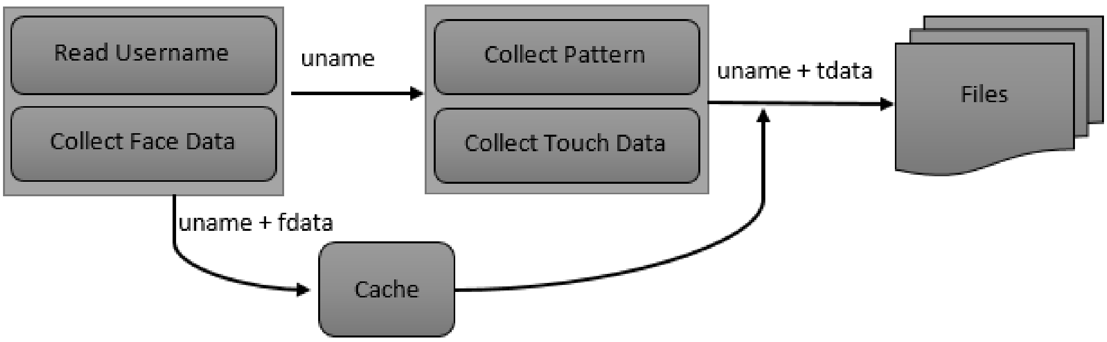


Figure 3.3: Data Acquisition Process

3.3.1 Data Acquisition

This is usually carried out as the first step in both user registration and user authentication phases. As mentioned earlier, four types of information will be collected as username, password pattern, face biometrics and touch behavior biometrics. Thus in the first stage, users are provided with a screen which contains text box to insert username and camera view to capture user's facial features. These facial biometric will be collected during the time user enter their username. The maximum number of frames (new faces or face updates) captured during this period is limited to 5 and finally, average values are taken.

This username and facial biometrics are stored into a cache before moving into next touch data acquisition stage. In order to collect touch data, users are provided with a screen contain pattern unlock screen to draw their preferred password pattern. Then entered password pattern will be encrypted and stored into different files in internal storage together with users touch behaviors and cached facial biometrics.

If this data collection process is a part of user registration, data collection is done as a repetitive process where one session contains 5 rounds. After each session data pre-processing unit will be invoked to extract features from these raw data. Face data acquisition and touch data acquisition will be discussed in detail in next subsections.

Face Data Acquisition

Possible facial features extracted from the human face can be categorized into three broad categories such as position, orientation, and activities. In facial feature extraction process, the face should be detected accurately as the first step (Figure 3.4). Therefore to handle the face detection of the video stream, Google Play Service Face Detection API will be used. This API simply detects areas in the image or a video that are human faces and it provides methods to read face position, orientation and activities as described below.

Position Face position detail includes width, the height of the area where a face was detected and other landmarks of the face such as left eye, right eye, left ear, right ear, nose base, bottom

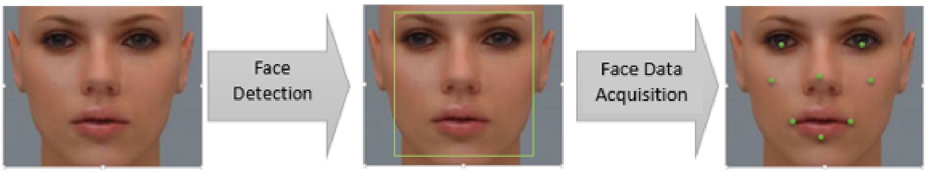


Figure 3.4: Facial Data Acquisition

mouth, left mouth and right mouth.

Orientation Describes the face posing angle with respect to X, Y and Z axis. As shown in Figure 3.5, the X, Y, Z coordinates are defined as the image is coordinated in XY plane and the Z axis as perpendicular to the XY and coming out of the frame. Rotation

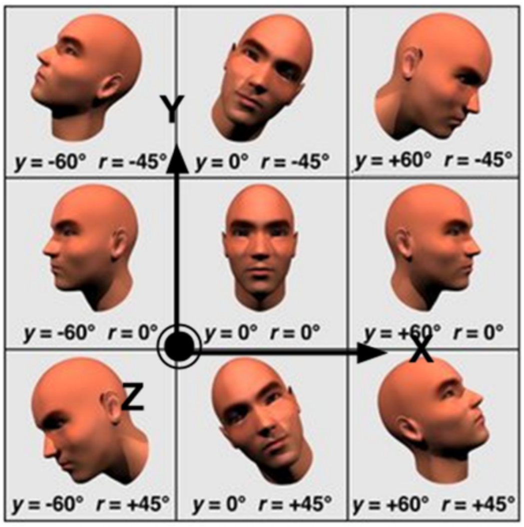


Figure 3.5: Possible Face Orientations [29]

Not only the face orientation, acceleration details of the device also collected (r).

Activity One of the other features provided by the Google Play Services API. Here it certain facial characteristics is taken into account generate advanced details like whether eyes are opened or not in the detected face and whether a face is smiling or not.

Touch Gesture Acquisition

In touch gesture acquisition process; spatial, timing and motion information of the user touch pattern will be taken. Here, touch gesture information is captured for each point of the pattern.

Spatial These features are associated with the physical interaction characteristics in between the user and the mobile device touch screen. Touch pressure, touch size, and touch position are the most common spatial features that can be seen in many research works [26]. Here touch pressure is obtained from **MotionEvent.getPressure**() API call and it returns the approximated force asserted upon each touch event. This value will be taken a value in the range 0 (soft touch) or 1 (hard touch). But this can be changed based on the device. Some devices return only either 0 or 1.

The touch size is another feature that often used with the touch pressure. This value represents the screen area being touched and each touch event is associated with a touch size. **MotionEvent.getSize()** android function is used to retrieve the touch size of each touch event.

Another important touch feature is the position. This feature is two-dimensional metric feature. This feature is associated with XY plane and X and Y coordinates are calculated for each touch event. This feature is considered to be important because these X and Y coordinates can be varied with user's fingertip size and the cognitive presence.

Timing This feature can be identified as a common feature in many touch dynamic biometric related research works. Once the user's finger reached the edge of any touch point, that point is triggered and thus that time will be read as the entry time. And when user's finger leave the edge of touch point that time is considered as the leave time. Likewise, the elapsed time from starting point to end point will be extracted by using Android OS API function calls as shown in Figure 3.6.

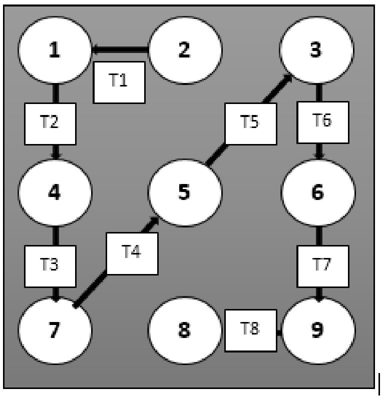


Figure 3.6: Timing

Motion Next is the Motion feature. Most of the mobile devices in the market are equipped with motion sensors, accelerometer, and the gyroscope. As mentioned in the literature, touch events usually make a small amount of movements and rotation to the device. Therefore these motion data also can be used to distinguish the subject. But as one touch event may cause multiple movements of the device, it is not possible to directly use this information as a touch feature. Instead some pre-processing such as applying statistical computations to generate meaningful value should be done and this value can be used in the data classification process.

So accelerometer sensor reading will be done while user entering the touch pattern to get information about user's device holding pose.

Device Selection

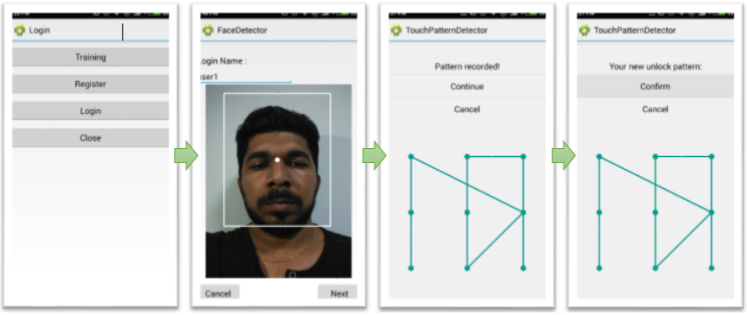
The device selection for data acquisition also an important factor because of different devices equipped with different sensors and features. In literature, majority of research works have used smartphones while only a few researchers have used tablets in their research works [26]. As the hypothesis of this project is to use facial features and touch dynamics as the third factor for user authentication, following factors were considered when selecting the device.

- Have android development platform.
- Good front camera to collect facial features.
- Multiple powerful built-in sensors to collect features like pressure, movements, and orientation.

Thus Samsung S3 mini model GT-18190N device with android version 4.1.2 and VGA front camera will be used for data acquisition because of it contains required sensors to collect touch pressure and touch area. Also Xiomi NOTE 1LTE, model HM NOTE 1LTE with android version 4.4.4 and 5MP front camera will be used for testing.

Training Data Set Acquisition

The availability of a public dataset in this research area would have been more useful to save time on data acquisition and also to focus on more challenging research issues and compare the performance of various algorithms by applying them on the same dataset. But the availability of public dataset in touch dynamic research domain is very limited and also those are related to string inputs. Therefore, the own dataset will be used for this project. The data acquisition flow is illustrated in 4.3 and each participant will have to repeat this process for 20 times as three sessions of 10 iterations in each.



.ogin Name :
iser1

Pattern recorded!

Your new unlock pattern:

Figure 3.7: Data Acquisition Flow

This data acquisition process can be a single session or can be divided into multiple sessions. According to literature, in the majority of research works data have been acquired in a single session [26]. This can affect the accuracy of the system because behavioral biometrics tends to change over the time even for same user [34]. Therefore using a single session may cause

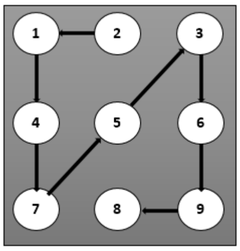


Figure 3.8: Password Pattern Interface

to miss inter-session variations of the user inputs.Ideally, the recommendation is to divide data acquisition operation into multiple sessions separated by some intervals and this approach has been adopted in many researches. Hence, data acquisition will be done in two sessions per each participant. Each session consists of 10 iterations. Thus a total number of iterations per participant will be 20.

The data set will be prepared by collecting data from 10 users. This process mainly consists of two phases.In the first phase, users are asked to enter their username. While the user is entering his/her username facial features are extracted during that period. A maximum number of frames captured here are limited to 5 frames per each session. Then average values will be taken.

In the second phase, users are requested to enter their password pattern. According to the literature in touch dynamic domain, the majority of experiments have used identical input string for all the subjects. It is around 72% [26]. The advantage of using an identical pattern for all subject helps to collect touch dynamic information independent from the touch pattern. Thus in this project participants will be provided with a touch pattern which is shown in Figure 3.8 in the training session. The Sequence of points is 2,1,4,7,5,3,6,9,8. Touch dynamics for each 9 point will be recorded.

So finally planning to collect up to 200 (20*10 records) pieces of touch gesture data and 200 pieces of facial features. These data will be stored in separate files.

3.3.2 Data Pre-processing

The purpose if having a pre-processing process is to improve computational efficiency as this authentication method will be used in mobile application authentication. It process, raw data to extract features and remove outliers in the raw data set to improve data quality and accuracy. Here outlier detection technique and dimension reduction technique will be used to get a representable data set out of all the raw data set.

In machine learning and pattern recognition domains, feature extraction is one of the main data pre-processing operations carried out to reduce a amount of resources required to perform data analysis on a large set of complex data. The feature extraction is mandatory to be carried out in both user registration and user authentication phase.

The feature extraction process starts from an initial set of raw data and constructs combination of variables to obtain unique set characteristics (features) which describe the data set with sufficient accuracy that could be used for subsequent learning and generalization steps. Moreover, it provides the benefits of dimensionality reduction. Because the analysis of large data set with a large number of variables requires more memory and computational power and also eventually it causing for over-fitting issues where classification algorithm over-fit to training sample but generalize poorly for new samples.

Here feature extraction will be used to extract unique characteristics (features) common to a user by processing the acquired raw data. This features could be used to distinguish one user from another. Thus these features will be used to train the authentication system.

The feature extraction process consists of two categories. One is **Facial Feature Extraction** and other is **Touch Gesture Extraction**. More information about feature extraction in each category will describe in following subsections.

Facial Feature Extraction

Feature extraction from collected raw face data is an important part of this approach. Here facial features are extracted using the size and relative position of important landmarks of the face, face activities and acceleration and rotational force along X, Y and Z axis.

As illustrate in Figure 3.9, eight distance matrices are calculated. those are;

- The distance between the left eye and right eye.
- The distance between the left cheek and right cheek.
- The distance between left mouth to right mouth.
- The distance between bottom mouth to nose.
- The distance between the right eye to nose.
- The distance between the left eye to nose.
- The distance between right cheek to nose.
- The distance between left cheek to nose.

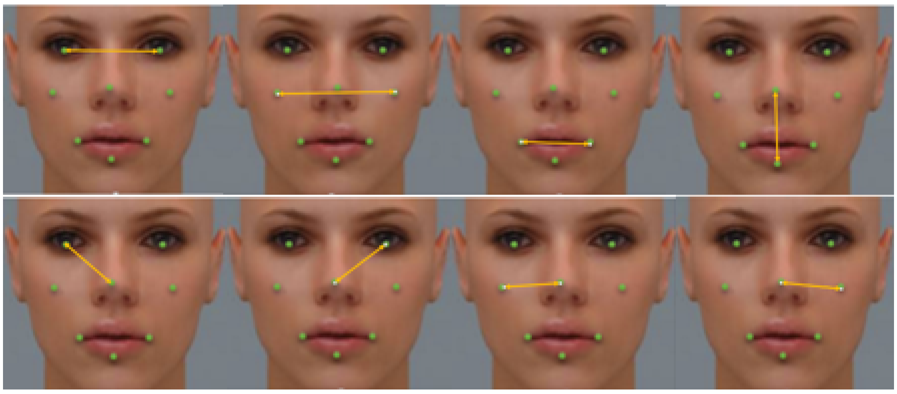
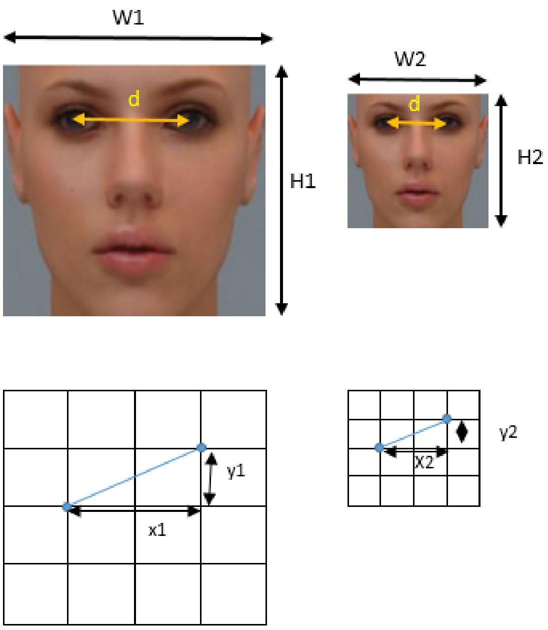


Figure 3.9: Face Feature Extraction



$$\text{Distance} = \sqrt{\left(\frac{x1}{w1}\right)^2 + \left(\frac{y1}{h1}\right)^2} = \sqrt{\left(\frac{x2}{w2}\right)^2 + \left(\frac{y2}{h2}\right)^2}$$

Figure 3.10: Landmark Distance Calculation

While face data collection, face size may vary based on the distance between the device and the user's face. So all these distances are calculated with relative to the size of the face as illustrate in Figure 3.10.

Here we made two assumptions.

- The distance between two landmarks along the X axis is proportional to the width of the face.
- The distance between two landmarks along the Y axis is proportional to the height of the face.

In addition to above-mentioned distance matrices; face activities, face orientation with respect to the X, Y and Z axis and acceleration and rotational forces along X, Y and Z axis also take place in the feature vector. Smiling probability, left eye open probability and right eye open probability are the vectors taken as face activities and EulerY (rotation around Y axis) and EulerZ (rotation around Z axis) are taken as face orientation features. So finally face feature vector is comprised of sixteen (16) features. This is the expected output from face feature extractor and the number of features uses for signature generation is determined after reducing features by applying some feature selection techniques like Information Gain Ratio and Gain Ratio.

Touch Feature Extraction

User touch dynamic pattern is consist of a set of unique features that can be useful in distinguishing users from one another. This touch feature extraction process is to extract these unique touch dynamic features by processing the raw touch data.

According to the literature, touch feature extraction can be done by using two different methods. Individual point (key) based feature extraction and overall key based extraction. In the first method, feature values of each individual touch point (key) is derived and feed into classifier for analysis. With the second method, it will derive average feature value of all the touch points before feeding into the classifier. Experimental results have shown that the point based feature extraction method perform better than the second methods regardless of the classification method used [26]. Ability to capture more fine-grained information from point based methods is identified as the reason for the outperformance.

The method using this approach can be identified as a combination of those two methods which are individual touch point based feature extraction and overall key based feature extraction. Because here we extract point based position information while calculating average values for other features like pressure, area based on the session. So touch feature vector will consist of twenty-five features. Those are Point1X, Point1Y, Point2X, Point2Y, Point3X, Point3Y, Point4X, Point4Y, Point5X, Point5Y, Point6X, Point6Y, Point7X, Point7Y, Point8X, Point8Y, Point9X, Point9Y, Average Touch Area Per Session, Average pressure per session, Total elapsed time, Average finger moving speed, Average acceleration along the X axis, Average acceleration along the Y axis, Average acceleration along the Z axis.

3.4 Data Classifier

Data classification is the most important operation in biometric basic authentication methods. The purpose of the classifier is to categorize (classify) collected data and extract features to generate authentication signature (template) and compare user feature data against the generated template (reference template). The outcome of this module is a matching score which is

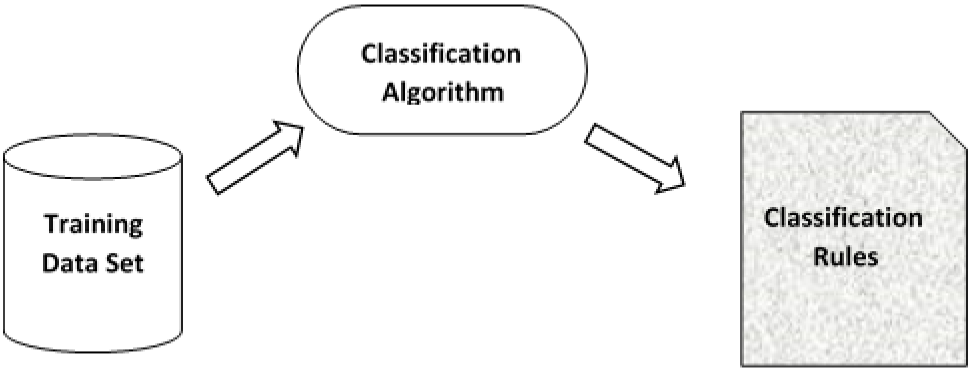


Figure 3.11: Data Classification

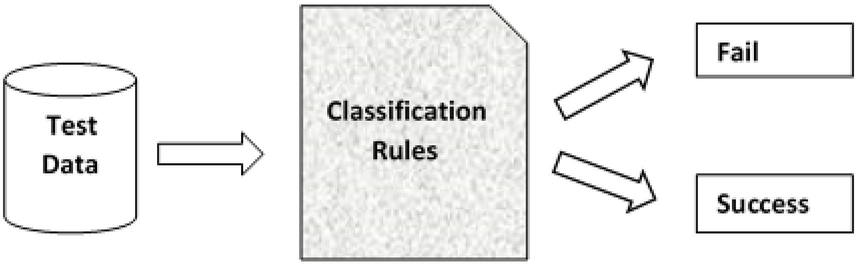


Figure 3.12: Test Classifier

used in decision making. Figure 3.11, Figure 3.12 and 3.13 respectively illustrate the abstract operational model of the data classifier.

The selected classification method is applied to both touch dynamic features and facial features to distinguish different users. Here prior evaluation of algorithms with extracted data will be performed using WEKA tool. WEKA is an open source software provided with a collection of machine learning algorithms. If the accuracy level of the classifier is acceptable it will be used to classify the new user behaviors in the future.

Data classification is usually done by using various machine learning algorithms (techniques) such as;

- Cluster Analysis Techniques.
- Statistical Techniques.

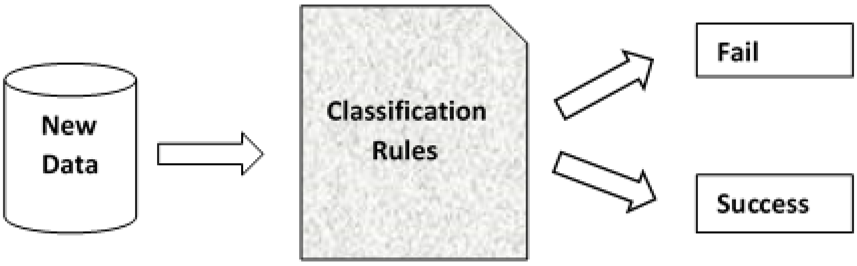


Figure 3.13: Validation

- Neural Networks.
- Support Vector Machine.
- Decision Tree.
- Probabilistic Modelling Techniques.
- Distance Measurement Techniques.

Among above techniques, cluster analysis technique and statistical technique is selected to be explored in this project. Here resource limitations and complexity factors were taken into consideration.

3.4.1 Cluster analysis

The Cluster analysis technique is using by assuming that samples of same subject exhibit similar behavior. Which mean it assumes that samples in same class should have similar properties. Thus the goal of using this technique is to group sample with similar properties to form a homogeneous cluster. According to literature, various types of cluster analysis techniques have been used throughout past years within this domain. k-means, k-star and k-Nearest Neighbour (KNN) [31] algorithms are popular among them. Hence KNN classification method will be explored in this approach.

KNN Classification Method The Nearest Neighbor classification method is a simple classification method. This method uses the distance measurement as the underline technique where it applies distance measurement between data and neighbors to calculate the distance value. According to this value, nearest is determined based on the K. There is no general optimum value for K and its value is usually obtained through trial and error approach.

The KNN classification method is considered as a simple classifier that can easily apply any distance measurement into the classification mechanism. The main benefit of using cluster analysis is that it can significantly increase the accuracy rate through increasing the matching efficiency.

Using KNN Classification Method In order to use KNN classification method, several training samples should be collected from each legitimate users. These each sample is considered as an n-feature vector. Then following steps are followed to perform user authentication based on KNN.

- **Generating User Signature (Profile)**: Here user's average feature vector is considered as the user profile. Therefore average n-feature vector is calculated for each user by using their sample matrices.

- **Clustering**: After generating user profiles, KNN classification method is applied to cluster profiles based on the distance measure.
- **Verification**: For new samples, KNN will be applied again to identify the cluster. Then check the cluster whether the user that sample is claimed to be is contained in the cluster. If the first condition is satisfied then the sample will be checked against that claiming user profile to determine whether the new sample is closer to claiming user profile.

3.4.2 Statistical Technique

Among different types of classification techniques used for user behavior classification, statistical techniques also have taken significant attention. Literature shows that there are several types of statistical techniques have been used in biometric research works [27] such as mean and standard deviation based techniques and deviation tolerance. Especially these techniques considered as more suitable for resource-constrained devices like mobile devices. In the survey done by Shen Teh et al [27], they have identified a number of advantages of using statistical technique over other machine learning techniques like Cluster Analysis, Distance Metrics, SVM, Neural Networks, etc.

- Less complex and easy to implement.
- Cost less computational time.
- Less resource consumption (battery power).

Therefore using a statistical technique to perform user authentication is considered here. Thus confidence interval is the selected method that will use in this project for model validation. Conceptually, a confidence interval is defined as an interval or range of values which is expected to contain an unknown population parameter with respect to some degree of confidence. This range is statistically derived from a sample drawn from the population.

The confidence interval statistical method is an interesting topic in the area of modeling and simulation because confidence intervals are commonly used in model validation. Typically a sample with all possible executions of the model is taken to calculate confidence intervals as an estimate of the population parameter (eg. mean) which would be the result of all possible execution of the model.

Using Confidence Interval Here the basic steps for calculating a confidence interval for a population mean. If the population is X and the sample is x_1, x_2, x_3, ..., x_n, following are the steps to calculate mean, standard deviation, confidence interval and finally generate the user signature.

- **Calculate Mean**: The first step is to calculate the sample mean $\bar{x}$ which can be easily calculated by using following equation 3.1.

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{n} \tag{3.1}$$

- **Calculate Sample Standard Deviation**: Then calculate the sample standard deviation s by using equation 3.2,

$$s = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{n-1}} \tag{3.2}$$

- **Calculate Confidence Interval**: Finally, confidence interval is calculated using the sample mean and the standard deviation. The confidence interval is a range of value and this range is denoted by [L, U]. Here L is the lower bound of the range and U is the upper bound of the range. Theoretically, this range is defined as,

$$[L,U] = [point_estimate - margine_error, point_estimate + margine_error]$$

When calculating the confidence interval for the population mean μ, sample mean $\bar{x}$ is taken as the point estimate. The calculation of margin error depends on the characteristics of the population where the sample is drawn from. If assume that the sample is drawn from a population is known to be normally distributed and the standard deviation σ of the population is known, the confidence interval is calculated by following formula.

$$[L,U] = [\bar{x} - z_c \frac{\sigma}{\sqrt{n}}, \bar{x} + z_c \frac{\sigma}{\sqrt{n}}] \tag{3.3}$$

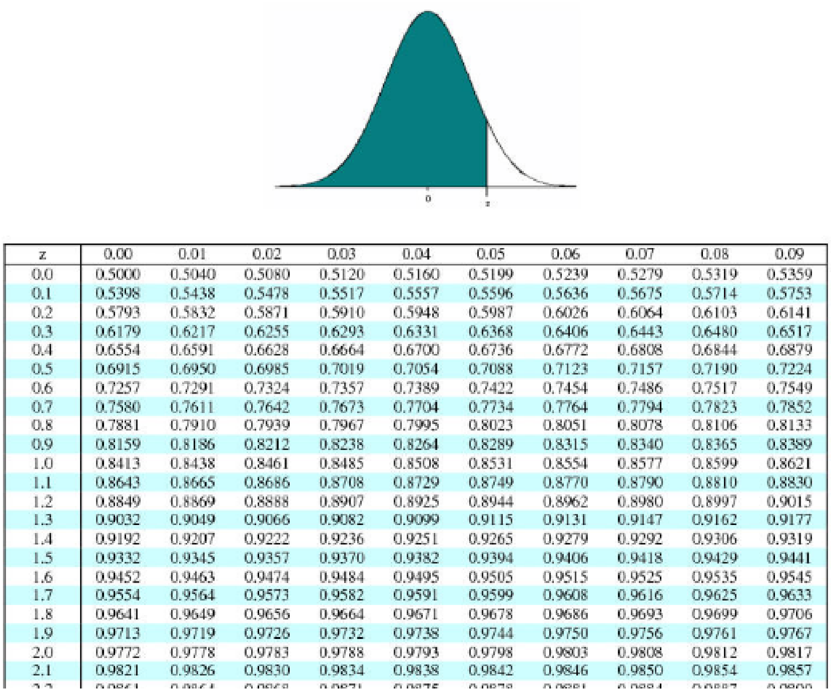
here z_c is identified as the critical value for the normal distribution. c is the confidence level of the distribution and it is defined as $c = (1-\alpha)$ where α is the level of significance which denotes the area under the distribution's probability density curve.

For an example, 95% confidence interval is calculated as,

$$Pr(-z \leq Z \leq +z) = 1 - \alpha = 0.95$$
$$Pr(Z \leq +z) = 1 - \frac{\alpha}{2} = 0.975$$
$$z = 1.96$$

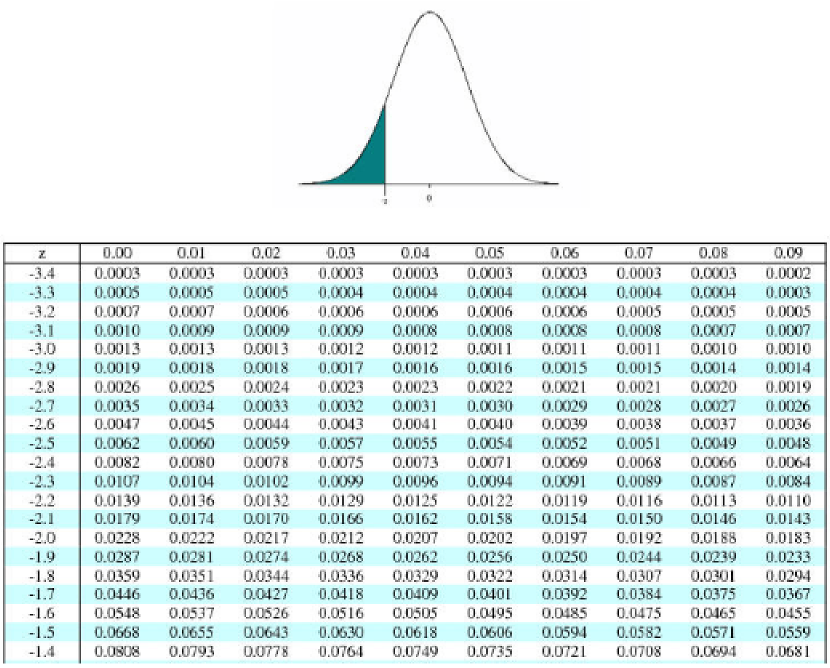
This z value can be found in Z-Score/ Z Transformation table as illustrate in Figure 3.14 and 3.15 [52].

Table 3.1 show some common critical values.



z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.2	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890

Figure 3.14: Standard Normal Probabilities for Positive Z-Score

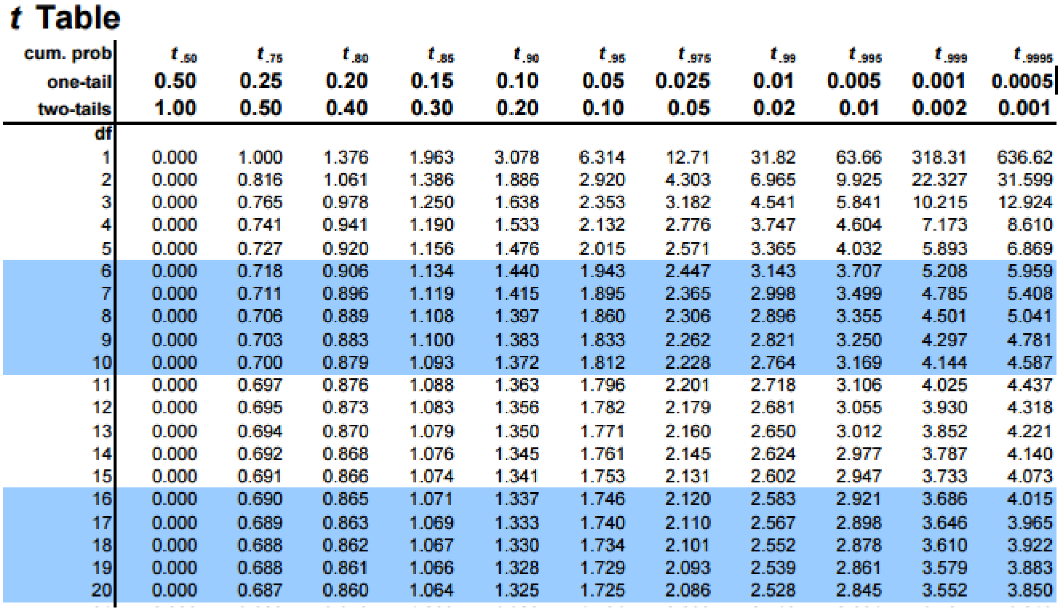


z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
-3.4	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0002
-3.3	0.0005	0.0005	0.0005	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0003
-3.2	0.0007	0.0007	0.0006	0.0006	0.0006	0.0006	0.0006	0.0005	0.0005	0.0005
-3.1	0.0010	0.0009	0.0009	0.0009	0.0008	0.0008	0.0008	0.0008	0.0007	0.0007
-3.0	0.0013	0.0013	0.0013	0.0012	0.0012	0.0011	0.0011	0.0011	0.0010	0.0010
-2.9	0.0019	0.0018	0.0018	0.0017	0.0016	0.0016	0.0015	0.0015	0.0014	0.0014
-2.8	0.0026	0.0025	0.0024	0.0023	0.0023	0.0022	0.0021	0.0021	0.0020	0.0019
-2.7	0.0035	0.0034	0.0033	0.0032	0.0031	0.0030	0.0029	0.0028	0.0027	0.0026
-2.6	0.0047	0.0045	0.0044	0.0043	0.0041	0.0040	0.0039	0.0038	0.0037	0.0036
-2.5	0.0062	0.0060	0.0059	0.0057	0.0055	0.0054	0.0052	0.0051	0.0049	0.0048
-2.4	0.0082	0.0080	0.0078	0.0075	0.0073	0.0071	0.0069	0.0068	0.0066	0.0064
-2.3	0.0107	0.0104	0.0102	0.0099	0.0096	0.0094	0.0091	0.0089	0.0087	0.0084
-2.2	0.0139	0.0136	0.0132	0.0129	0.0125	0.0122	0.0119	0.0116	0.0113	0.0110
-2.1	0.0179	0.0174	0.0170	0.0166	0.0162	0.0158	0.0154	0.0150	0.0146	0.0143
-2.0	0.0228	0.0222	0.0217	0.0212	0.0207	0.0202	0.0197	0.0192	0.0188	0.0183
-1.9	0.0287	0.0281	0.0274	0.0268	0.0262	0.0256	0.0250	0.0244	0.0239	0.0233
-1.8	0.0359	0.0351	0.0344	0.0336	0.0329	0.0322	0.0314	0.0307	0.0301	0.0294
-1.7	0.0446	0.0436	0.0427	0.0418	0.0409	0.0401	0.0392	0.0384	0.0375	0.0367
-1.6	0.0548	0.0537	0.0526	0.0516	0.0505	0.0495	0.0485	0.0475	0.0465	0.0455
-1.5	0.0668	0.0655	0.0643	0.0630	0.0618	0.0606	0.0594	0.0582	0.0571	0.0559
-1.4	0.0808	0.0793	0.0778	0.0764	0.0749	0.0735	0.0721	0.0708	0.0694	0.0681

Figure 3.15: Standard Normal Probabilities for Negative Z-Score

Confidence level c	Normal z
0.80	1.282
0.90	1.645
0.95	1.960
0.99	2.576

Table 3.1: Critical Values for Confidence Intervals



t Table

cum. prob	$t_{.50}$	$t_{.75}$	$t_{.80}$	$t_{.85}$	$t_{.90}$	$t_{.95}$	$t_{.975}$	$t_{.99}$	$t_{.995}$	$t_{.999}$	$t_{.9995}$
one-tail	0.50	0.25	0.20	0.15	0.10	0.05	0.025	0.01	0.005	0.001	0.0005
two-tails	1.00	0.50	0.40	0.30	0.20	0.10	0.05	0.02	0.01	0.002	0.001
df											
1	0.000	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	318.31	636.62
2	0.000	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	22.327	31.599
3	0.000	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	10.215	12.924
4	0.000	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	0.000	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	0.000	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	0.000	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	0.000	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	0.000	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.000	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	0.000	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	0.000	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	0.000	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	0.000	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	0.000	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	0.000	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	0.000	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	0.000	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	0.000	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	0.000	0.687	0.860	1.064	1.325	1.725	2.086	2.528	2.845	3.552	3.850

Figure 3.16: t Distribution Table

But these calculations are based on the assumption that the standard deviation σ of the population is known. So if assume that the standard deviation σ of the population is not known, the σ value is estimated using the sample standard deviation s. Then the t-distribution which is also known as Student t distribution is used instead of z distribution to calculate the confidence interval.

Thus the confidence interval for the population mean μ when σ is not known is calculated by equation 3.4.

$$[L,U] = [\overline{x} - t_c\frac{s}{\sqrt{n}}, \overline{x} + t_c\frac{s}{\sqrt{n}}] \tag{3.4}$$

Here t_c is the critical value of the t-distribution for c confidence level. This t_c value for different c values can be obtained from t-distribution table (Figure 3.16 [53].

As illustrate in Figure 3.16, for a given confidence level the critical value in t distribution is may vary based on the degree of freedom which is denoted by d.f. The **degree of freedom** is defined as the number of independent values on which the estimate is based. This value is equal to the number of values minus the number of parameters estimated. As population mean is not known here d.f value equals to the $n-1$ where n is the sample size.

Table 3.2 show some common critical values for different d.f values.

Confidence level c	**t Distribution**			
	d.f. = 5	**d.f. = 10**	**d.f. = 20**	**d.f. = 30**
0.80	1.476	1.372	1.325	1.310
0.90	2.015	1.812	1.725	1.697
0.95	2.571	2.228	2.086	2.042
0.99	4.032	3.169	2.845	2.750

Table 3.2: t Distribution Critical Values for Confidence Intervals

However deciding the distribution whether z distribution or t distribution to be used for confidence interval calculation is more complicated and it solely based on whether the population standard deviation σ is known or not. Therefore when selecting the distribution, following factors should be considered.

1. **Population Distribution**: Whether the distribution of population is known or unknown and if it is known whether it is a normal distribution or approximately normal distribution which means that the distribution is reasonably symmetric and mound-shape. This can be checked by plotting the histogram of the sample.
2. **Population Standard Deviation**: Whether the population standard deviation σ is known or unknown.
3. **Sample Size**: Whether the sample size (n) is ≥ 30 or not.

Then the distribution should be determined according to the guidelines in the following Table 3.3 [55] [56].

Option	**Population Standard Deviation** σ	**Sample Size** n	**Equation**
1	Known	≥ 30	3.3
2	Known	< 30	3.4
3	Unknown	–	3.4

Table 3.3: Guidelines to Select the Equation for Confidence Interval Calculation [59]

- **Generate User Signature**: Then the final step is to generate user signature by concatenating the sample mean $\bar{x}$, sample standard deviation s, and $[L, U]$ confidence intervals for each feature. If there are m number of features, then;

  Signature = "$\bar{x}_1, s_1, L_1, U_1 | \bar{x}_2, s_2, L_2, U_2 | \bar{x}_3, s_3, L_3, U_3 | ..... | \bar{x}_m, s_m, L_m, U_m$"

  Here two signatures are generated per users one by using face features and other is from touch features.

3.4.3 Improve Accuracy of the Classifier

Usually, sample data use for data classifications are either gathered by themselves or retrieve from the database which is collected before by some domain expertise. Then features are extracted from this sample dataset and classification model is generated based on features selected from these extracted features/attributes. Therefore the selection of features/attributes is critically important in classification regardless of the way sample data set is collected. Using a wrong set of features for modeling may mislead the outcome of the classifier and also it may lead over-fitting issue.

Mislead Results The root cause for the misleading issue is having redundant features among the selected feature vector. Having redundant features implies that one relevant feature gets redundant due to there is another feature which is strongly correlated. Such feature redundancies affect more on instance-based classifiers such as KNN. Because of the K-Nearest Neighbor algorithm uses a small set of neighbors in feature space there is a high possibility of getting affected by feature redundancy.

Over-fitting to Training Dataset If there are too many features then the model (hypothesis) may fit the training data set very well but may fail to generalize to new samples. This is considered as the over-fitting issue. So having irrelevant features in feature vector is identified as the main reason for over-fitting issue. Finally, the over-fitting of the training data can negatively affect the predictive accuracy since the model is not generalized.

Hence it is important to remove redundant and irrelevant features from feature vector prior to performing data classification. The removing redundant and irrelevant features is identified as the **feature selection**.

Feature Selection

Feature selection technique is also known as the attribute selection and variable selection. This technique is used in both machine learning and statistics domain to remove redundant and irrelevant features/attributes. The feature selection technique is a process to find the best subset of features from the data set. This is totally different from the process of feature extraction whereas feature extraction creates new features from existing features in the data set while feature selection select subset of most important features from that extracted feature set. The notion of ”best” is typically meant the highest accuracy. But this may change according to the application type.

Perform feature selection before classification provides a number of advantages.

- **Generalization**: Generalization is achieved by reducing overfitting by selecting most relevant features. Formally this will reduce the variance.
- **Improve Accuracy**: Improve accuracy by reducing the misleading data.
- **Less Training Time**: Lesson data implicitly increase the training speed through reducing the complexity of the model.

There are three types of feature selection methods such as filter method, wrapper method and embedded method [57].

Filter Method This method uses a statistical approach to rank features. First, it applies a statistical measure to calculate a score for each feature. Then it will return feature rankings which are ranked by the score. So this ranks can be used either to keep or remove low ranked features from data set. The Chi-Squared test, Information Gain, Gain Ratio and Correlation Coefficient are examples for filter methods. These methods are considered to be univariate and feature independent [57].

Wrapper Method Wrapper method is relies on a heuristic search of the space of all possible feature subsets. The most common approach is for heuristic search is hill climbing where keep adding features one at a time until there is no further improvement. This is called ”forward greedy wrapping”. One other alternative is ”Backward Greedy Wrapping” where it starts with a full set of predictors and removing features one at a time until no further improvement can be achieved. The third method is to interleave adding and removing phases either in forward or backward. This method is called ”Forward-Backward Wrapping” [57][58]. These different combinations of subsets are prepared, evaluated and compared to other combinations. A predictive model is used here to evaluate the combination and will assign a score based on the accuracy.

Embedded Method This technique is used during the model creation. Here is will check which features contributed more to give an accurate result. The generalization methods are known as the most common type of embedded feature selection method. Algorithms such as Elastic Net, Ridge Regression, and LASSO are examples for regularization algorithms. These methods introduce additional constraints such as regression algorithm into the predictive algorithm which make the model bias toward lower complexity [57].

According to the characteristic of this proposed approach ”Filter Method” will be used for feature selection.

3.5 Validator

This is the module to make the decision based on the output of classifier and Figure 3.13 is briefing the process. The validation module will perform a comparison between current user signature against with the target user’s signature who’s that the current user is pretended to be. The decision is made based on the dissimilarity score value generated my machine learning technique with respect to a predefined threshold value. This threshold value is determined through trial and error method.

However, this project apply fusion approach where combine two matching scores from different feature sets such as touch dynamics and facial features to make the final decision. The purpose of having the fusion approach is to increase the accuracy of the validator. There are two different approaches are used in literature to validate users. One is user verification where verify user against the claimed identity of the user. The second approach is user identification. In this approach, user will be identified from the set of legitimate users. The user verification approach which is shown in Figure 3.10 is considered to be the most suitable approach for authentication.

As mentioned before, two classification methods will be explored in this project one is a cluster analysis based approach while other is the statistical approach. In cluster analysis approach, decision-making process consists of both identification and verification approaches. Here identification is used by means of identifying the correct cluster to identify the cluster that claims user belongs to and then verification is used to verify the user is the actual user that claimed to be. In statistical model, only verification approach will be used to verify the user.

3.6 Summary

This chapter discussed in details about the methodology used in this project with including all the descriptive information of the design and the activity flow of the project. The proposed authentication system is consists of three main modules such as data collector, classifier and validator and two operation modes. The two operation modes are User Registration and User Authentication.

The first module of the system is data collector module which takes part of both operation modes of the authentication system. The data collector module comprises of other two sub-modules/units which are take care of data acquisition and data pre-processing. Data acquisition unit is responsible for acquisition of raw data by using different inbuilt sensors like camera, touch sensors, accelerometer etc. Then the pre-processor will process above raw data to extract some meaningful features to use in later stages to distinguish users from each other.

The second module is the data classifier which can be considered as the heart of the authentication system. By considering resource constraints of mobile devices, here two classification methods will be explored. One is cluster analysis approach and other is a statistical approach. In addition, feature selection techniques also used here to remove redundant and irrelevant features from feature vector to improve the performance by reducing complexity and increasing the accuracy of the system. This feature selection process will be done externally (manually) by using WEKA tool, which is an open source tool consists of different types of machine learning techniques. This is commonly used in the machine learning and statistical analysis domains.

The last component is the validator which is responsible for the final decision making. Here it uses fusion approach which combines information from multiple sources to improve the accuracy of the authentication system. Here it uses a type of Score level fusion (SLF) approach which is performed after the data classification operation. It combines two machine scores calculated on two different feature data to make the decision. The decision is made based on a predefined threshold value which is obtained through trial and error method.

Next chapter is allocated for the implementation of the proposed authentication methodology. It will describe how different techniques described in this chapter are used practically and challenges faced during the implementation.

Chapter 4

Implementation

4.1 Overview

The previous chapter brief out design and methodology of the project. This chapter will describe the process of the implementation. The implementation of this research will be done by focusing the Android mobile applications. Based on a literature study [26], Android is considered as the most popular development platform among mobile devices which is followed by iOS and Windows respectively. More information about selecting Android OS have been discussed in first two chapters.

Thus the first section of this chapter is allocated for the development environment preparation which includes information about the IDEs and other tools like WEKA, Octave that utilized during the implementation phase. Then next sections will focus on the implementation of the design and activities of the proposed system. So the arrangement of next sections as follows.

The second section will include information about the implementation of the application. Here it will describe the initial steps of the implementation with including a class diagram. Then next subsequent sections are allocated for each module of the application.

The third section will include implementation detail of the data collector module which one of the main three modules as described in the previous chapter. This section will describe data acquisition process with including information about the Android API and available sensors, data pre-processing process which includes feature extraction and how to build the two different feature vectors for face biometrics and touch dynamics and finally the data storing mechanism which can be used efficiently to share information among modules.

The fourth section of this chapter is allocated for the data classifier which is considered as the heart of this application. Here it will discuss the implementation of two different approaches. And also it will contain some other data analysis technique used to enhance the accuracy of the classifier. Feature selection technique can be identified as such technique that used within this process.

Finally, the fifth section is to describe the implementation details of the validator. It will consist of threshold value selection and other validation criteria.

The last section which is the sixth section will be the summary of this chapter which will brief all the information discussed in this chapter.

4.2 Development Environment

During the implementation, selection of the development platform also very important.Here Android platform was selected over iOS and Windows as the development platform. When selecting the development platform customizability, flexibility, cost and market share was taken into consideration. These are some common criteria that used in literature. More details about the background of the development platform have been discussed under the chapter 2, the literature survey.

Especially the android application development can be done on any of the following operation systems which are consists with other required tools like JDK5 or above and Android Studio or Eclipse.

OS support for android development:

- Microsoft Windows XP or later version
- Mac OS X 10.5.8 or later version with Intel chip.
- Linux including GNU C Library 2.7 or later

As mentioned earlier, the cost is one of the factors that considered when selecting the development platform. Hence Android is considered to be cost effective since all the required tools to develop Android applications are freely available to download from the web.

Moreover, there are many sophisticated technologies available for android application developments. Even though it is optional to use them, they will helpful to reduce the programmer's effort on android application development. Android Studio and Eclipse can be identified as the most popular and common IDEs that are used for android application developments. But with the stabilization of Android Studio, Eclipse IDE considered being deprecated in the case of android developments.

In this project, android application development is done on Microsoft Windows 10 installed with other required development tools like java JDK7 and Android Studio.

As this project mainly focuses on Android application development, next sub-sections are organized as; the first subsection is allocated for the IDE used for the developments. The second is about the resources available in android devices such as sensor and features. The last section will be discussed about tools that will be used for data analysis.

4.2.1 Android Studio

Android Studio is the official IDE (Integrated Development Environment) that recommended to use for Android platform developments. This is a replacement for the Google's primary IDE namely Eclipse Android Development Tools (ADT). Android Studio is developed based on JetBrains' IntelliJ IDEA and it is specifically designed for Android developments. The stable release of Android Studio was released in 2014 and it is freely available under the Apache License 2.0 for download on Windows, Mac OS X and Linux.

Android Studio provides the fastest tools for building Android applications on every type of Android devices. Also, it is equipped with capabilities of code editing, debugging, instant build/deployment, etc. Following are some of the important features available in Android Studio;

- Gradle-based build support
- Capability to catch performance and usability using Lint tools.
- A rich layout editor which supports drag and drop UI components.
- Provide an option to preview layouts on multiple screen configurations.
- Provided with an Android Virtual Device to deploy applications.

4.2.2 Resources

Android powered devices are consists of various types of built-in resources such as sensors to measure motion, orientation and different other environmental conditions. These sensors will provide user raw data with high precision and accuracy. Therefore this capability of android powered devices is helpful to monitor three-dimensional device movement, positioning and surrounding environmental factors like gravity, temperature and humidity. The sensors provided with Android platform can be divided into three broad categories as Motion Sensors, Position Sensors and Environmental Sensors [50].

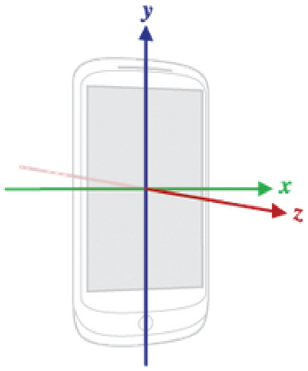


Figure 4.1: Coordination System of Accelerometer [50]

Motion Sensors Usually motion sensors are useful to monitor the device movements like shake, rotation, tilt and swing. This information reflect the user inputs as well as the physical environment. The motion sensor category includes accelerometers, gravity sensors, rotational vector sensors and gyroscope. The main purpose of motion sensors is to measure acceleration forces and rotational forces along X, Y and Z axes. Among the motion sensors, accelerometer and gyroscope are hardware-based sensors while others like gravity, linear acceleration and rotation vectors are either hardware based or software based.

Position Sensors Sensors which are used to measure the physical position of a device are categorized under Position Sensors. Orientation and Magnetometers the two categories of position sensors. The type of sensors provided in Android platform to determine the device position is called geomagnetic field sensor, the accelerometer sensor and proximity sensor. Among those position sensors, geomagnetic is a hardware-based sensor which is available in most of the handsets.Users can use geomagnetic sensor together with accelerometer to determine the device orientation with reference to the world frame such as determine device position relative to the north magnetic pole. The proximity sensor is used to identify whether the device is being held close to the user's face.

Environmental Sensors This is one of the three types of sensors provided with android platform. Usually, this type of sensors is used to measure various environmental parameters, such as ambient air temperature and pressure, illumination, and humidity. Android provides four types of sensors which include light, pressure, temperature and humidity. All of these sensors are hardware based and availability may vary according to the manufacturer.

The accelerometer is used in this project to read device orientation information while user entering username and password. It provides acceleration force data with respect to the standard 3-axis coordinate system. As same as most sensors, the coordinate system is defined relative to the device's screen when the device is in its default orientation as shown in Figure 4.1. Most importantly, this coordinate system axes won't change either on the device's screen orientation is changed or device movement [50].

4.2.3 Data Analysis Tools

The user authentication using biometrics on mobile devices converge more into the area of machine learning. Hence two data analytic tools will be used here to analyze raw data and process data. Those are WEKA 3.6.13 and Octave 4.0.3.

WEKA Weka (Waikato Environment for Knowledge Analysis) is an open source software issued under GNU General Public Licence. This workbench is enriched with a collection of machine learning algorithms and tools that can be used for data analysis and predictive modeling tasks such as visualization, data pre-processing, classification and clustering. Users can use either the graphical interface or a java application to apply algorithms to the dataset.

Octave Octave Software is one of the main free alternatives for MATLAB. This is an open source software featuring a high-level programming language issued under GNU General Public Licence. The main focus of Octave is mainly focused is on numerical computations such as solving linear and non-linear equations, numerical linear algebra, statistical analysis, etc. Also, it can be used as a batch-oriented language for automated data processing.

Moreover, Octave interpreter uses an OpenGL graphic engine to create plots, graphs and charts which are more similar to ”gnuplot”

4.3 Implementation

A mobile application need be used to collect face biometrics and touch dynamic data that required for this project. Therefore the application was developed as prove of concept (POC) and also with the intention of collecting user data. So the Main activity is consist of several options that navigate the user to different modes which are namely Training, Registration and Authentication. Figure 4.2 illustrate the main screen of the application.

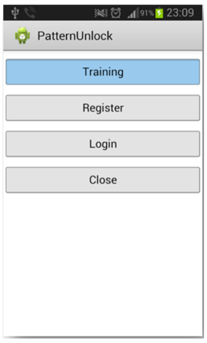


Figure 4.2: Main Display

For all of these three operational modes, users are directed to the page where they need to provide the username and then touch pattern respectively. The process is clearly shown in Figure 4.3.

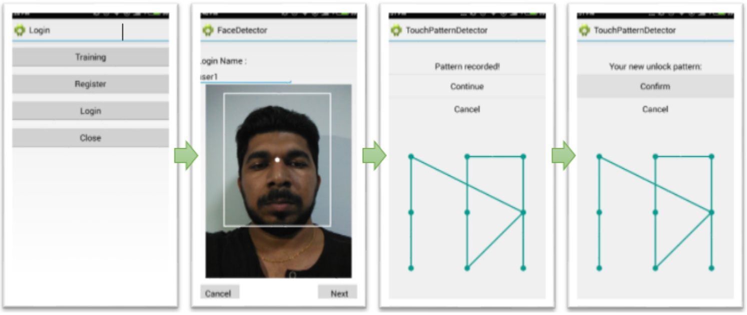


Figure 4.3: Data Acquisition Flow

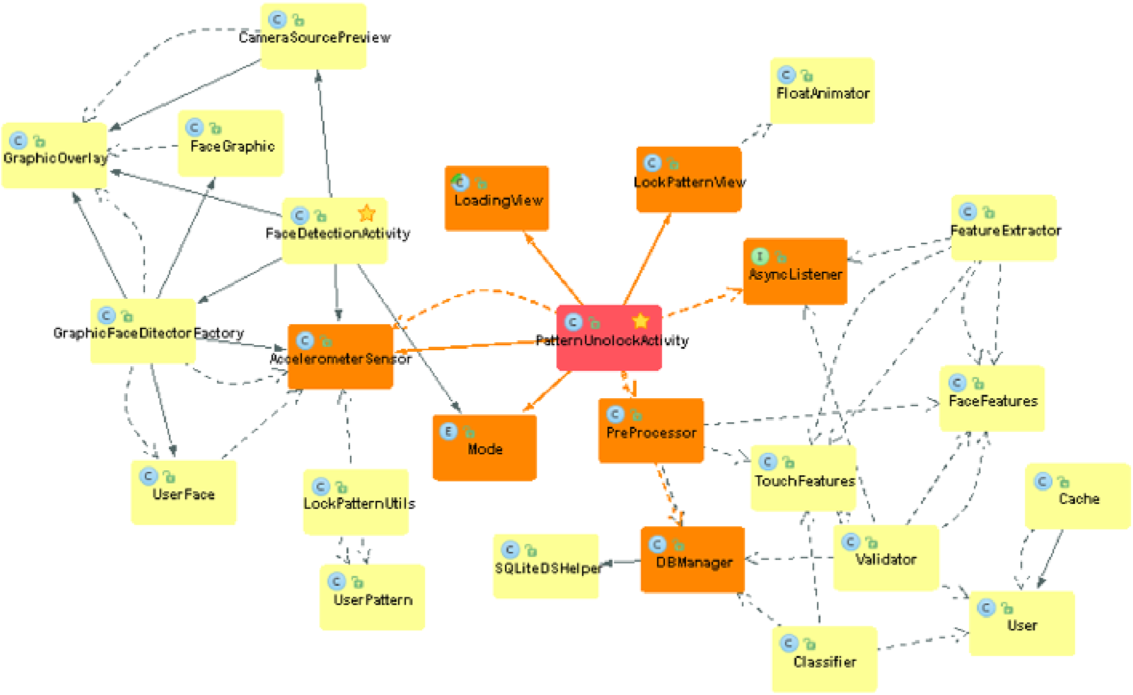


Figure 4.4: Class Diagram

During this process data pre-processing and storing operations are taking place as asynchronous operations. Figure 4.4 illustrates a high-level view of the class diagram details information will be discussed under sections that allocated for each module.

4.4 Data Collector

Basically, the proposed application perform in two operational modes called user registration and user authentication. In addition to these two modes, this application is used to collect the training dataset to be used to train the application. So data collection in all these three operational modes is handled by the data collector module. This module is one of the main three modules in the application.

The purpose of this module is to read input from different inbuilt resources of the device like front camera, touch sensors, accelerometer sensors etc., and store them into files to be used later and then pre-processing will be done in order to retrieve important features from collected raw data that need to be feed into data classifier for user classification. These pre-process data will be stored into an internal database. Here SQLite database is used as the data store.

4.4.1 Data Acquisition

As described before, data collector modules comprise of two submodules to handle data acquisition and data pre-processing operations. Within data acquisition operation, following four types of data will be collected.

1. Username
2. Face biometrics
3. Touch pattern
4. Touch behavior biometrics

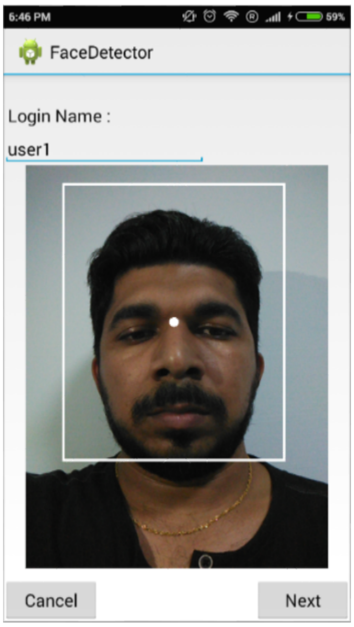
Hence the data acquisition is carried in two phases such as face data acquisition and touch data acquisition.

Face Data Acquisition

In the first phase, users are navigated to a screen that contains a field to enter the user name and the camera view. Figure 4.5 illustrate the face detector interface that used to collect username and facial features. Face data acquisition is a process containing two steps where the first step is to detect the face from the video stream and then extract landmark positions from the detected face. This process runs as a background process while user entering their username. Basically, this strategy for face data acquisition is used to make sure that user is focusing on the screen during the data acquisition period.

The Face Detection API is used to detect user face which was released with the Google Play Services 7.8. Face Detection API does not provide functionality for face recognition but for face detection. This API simply detects the presence of human faces in the image or video but it is not capable of determining whether two faces correspond to the same subject (user). It infers the faces with slight changes in position in consecutive frames of video are the same face, but if the face is disappeared from the field and re-entered then it is considered as a new face.

Especially this face detection API is capable of distinguishing faces at a different orientation and with different facial expressions. Therefore specific landmarks of detected face such as eyes, cheeks, nose and mouth can be collected even if the subject's head has slight turn sideways.



Login Name :
user1

Figure 4.5: Face Data Acquisition Screen

In order to use Google face detection API, Google Play Services should be added as a dependency as shown in Listing 4.1.

Listing 4.1: Google Play Services Configuration

```
dependencies {
    compile 'com.google.android.gms:play-services:7.8+'
}
```

In the onCreate method of the Face Detector Activity, both camera source and accelerometer should be initialized to capture face data and mobile orientation during the session. Listing 4.2 shows the initialization steps of the camera.

Listing 4.2: Create Camera Source

```
private void createCameraSource() {
        Log.i(CLASSNAME, "Create camera source");
        Context context = getApplicationContext();
        FaceDetector detector = new
           FaceDetector.Builder(context)
                 .setTrackingEnabled(true)
                 .setClassificationType(FaceDetector.ALL_LANDMARKS)
                 .build();


        fdFactory = new
           GraphicFaceDitectorFactory(mGraphicOverlay,
           getFilesDir(), accelerometer);
        detector.setProcessor(
                new
                   MultiProcessor.Builder<>(fdFactory).build());

        if (!detector.isOperational()) {
                Log.w(CLASSNAME, "Face detector dependencies
                   are not yet available.");
        }
        mCameraSource = new CameraSource.Builder(context,
           detector)
                .setRequestedPreviewSize(800, 800)
                .setFacing(CameraSource.CAMERA_FACING_FRONT)
                .setRequestedFps(20.0f)
                .build();
    }
```

The face detector is initialized with options for detecting faces with landmarks by setting the classification type to `FaceDetector.ALL_LANDMARKS` (Listing 4.3. Other important option is to set ”tracking enabled” to true. Setting ”tracking enabled” is important to get more accurate and faster result for detection on consecutive images such as video stream.

Listing 4.3: Initialize Face Detector

```
FaceDetector detector = new FaceDetector.Builder(context)
        .setTrackingEnabled(true)
        .setClassificationType(FaceDetector.ALL_LANDMARKS)
        .build();
```

The `Detector` can be identified as the base class for implementing face detector instance. Usually, the detector instance accepts a `Frame` as an input and return detected items as the output. But according to this project, it is required to process each frame received from the camera and detect the face on the frame. Hence in this project, `Detector` is used within a pipeline structure in conjunction with a `MultiProcessor` which continuously receives frames from camera source and then handles detected items (faces) via `GraphicFaceDitectorFactory.GraphicFaceTracker` Additionally, `GraphicFaceDitectorFactory` is fed with file location to store details of the detected faces and also an instance of accelerometer `AccelerometerSensor` to capture the device orientation.

The first time that an application using the Face API which is installed on a device, Google Mobile Services (GMS) will download the required library to the device. Usually, this is done by an installer before the application is run for the first time. But there is a possibility to fail this pre-installation process due to network issue or a device issue such as lack of storage. So `isOperational` method of the detector is used here to make sure that required library is available or not.

The `GraphicFaceDitectorFactory.GraphicFaceTracker` will listen on new item detections or item updates and will add/update the `UserFace` collection accordingly. This process will run as an asynchronous task, perform as a background operation and publish results on UI thread which make proper and easy use of UI thread. However number of frames and face detected during this period may vary and also the number of frames that need to be processed may take a higher value than required. Therefore the number of faces collect during one session has been limited to 5. This helps to reduce the memory consumption during the period.

Device Acceleration Information

In addition to face landmark information, acceleration sensor measures also used here to capture the acceleration applied to the device during the session. Thus Accelerometer sensor also needs to be initialized during the `onCreate` method call of the Face Detector Activity. Listing 4.7 shows how to get an instance of the acceleration sensor.

Listing 4.4: Initialize Acceleration

```
private void enableAccelerometer() {
        senSensorManager = (SensorManager)
            getSystemService(Context.SENSOR_SERVICE);
        senAccelerometer =
            senSensorManager.getDefaultSensor(Sensor.TYPE_ACCELERATION)
        senSensorManager.registerListener(this,
            senAccelerometer,
            SensorManager.SENSOR_DELAY_NORMAL);
}
```

The `getSystemService` method is used to get a reference to the sensor service of the system by passing the name, `Context.SENSOR_SERVICE`. Then a reference to the system's accelerometer need to be taken through the sensorManager which can be done by calling `getDefaultSensor( Sensor.TYPE_ACCELERATION)` with the type of the sensor going to be used. Finally acquired sensor must be registered as a listener to get readings. This is done by invoking `registerListener` public method of `SensorManager`. Here the expected sensor event delivery rate is given by `SensorManager.SENSOR_DELAY_NORMAL` which 3 microseconds(this value is the default value that suitable for screen orientation changes).

Theoretically, the acceleration applied to the device A_d is calculated by using the forces applied on the sensor (F_s) and the mass (m) of the device by using following equation 4.1. The relationship 4.2 show the gravitational influence on the acceleration measurement, measured by using equation 4.1 [50].

$$A_d = -\frac{\sum F_s}{m} \tag{4.1}$$

$$A_d = -g - \frac{\sum F_s}{m} \tag{4.2}$$

Therefore it is recommended to remove the contribution of the force of gravity while measuring the real acceleration of the device. There are two ways to read three-dimensional vector representing acceleration along each axis, excluding the gravity. One is to define to use normal accelerometer sensor and use different methods to filter sensor data such as use a simple high-pass filter to remove gravity. In order to apply a high-pass filter, it is required to isolate the force of gravity first. This can be achieved by applying a low-pass filter. Following example (Listing **??**) shows the extraction of actual acceleration on the device.

Listing 4.5: Initialize Acceleration

```java
private void readAcceleration(final SensorEvent event) {
        if(event.values != null) {
                final float alpha = event.timestamp
                   /SensorManager.SENSOR_DELAY_NORMAL;
                final float[] gravity = new
                   float[event.values.length];
                // Isolate the force of gravity with the
                   low-pass filter.
                gravity[0] = alpha * gravity[0] + (1 -
                   alpha) * event.values[0];
                gravity[1] = alpha * gravity[1] + (1 -
                   alpha) * event.values[1];
                gravity[2] = alpha * gravity[2] + (1 -
                   alpha) * event.values[2];

                // Remove the gravity contribution with the
                   high-pass filter.
                accelerometer.setX(event.values[0] -
                   gravity[0]);
                accelerometer.setY(event.values[1] -
                   gravity[1]);
                accelerometer.setZ(event.values[2] -
                   gravity[2]);
        }
}
```

The second method is to use linear accelerometer to read acceleration information as in Listing 4.6. the Linear acceleration sensor provides acceleration measures along the three axis without the influence of gravity. Therefore linear acceleration is usually used to perform gesture detection.

Listing 4.6: Initialize Linear Accelerometer

```
senAccelerometer =
   senSensorManager.getDefaultSensor(Sensor.TYPE_LINEAR_ACCELERATION);
```

Conceptually, the linear accelerometer sensor provides acceleration data according to the following relationship:

```
linear acceleration = acceleration - acceleration due to
   gravity
```

Therefore this second method is used in this project to capture device acceleration. Upon completion of the first face data acquisition phase, collected raw data will be stored into internal data store using appendToFile in InternalDSHelper class. These raw data will be stored as a comma separated list (A) into a file and also keep in cache.

Listing 4.7: Initialize Acceleration

```
public static void appendToFile(Context context, String
   data,String file) throws IOException {
                FileOutputStream fos =null;
                try {
                        fos = context.openFileOutput(file,
                           Context.MODE_APPEND);
                        fos.write(data.getBytes());
                } catch (Exception e) {
                        Log.i(CLASSNAME,
                           Constants.DATA_STORE + " file not
                           found");
            throw e;
                }finally {
                        if(fos != null) try {
                                fos.close();
                        } catch (IOException e) {
                                Log.i(CLASSNAME,
                                   Constants.DATA_STORE + "
                                   Error in closing stream
                                   ");
                        }
```

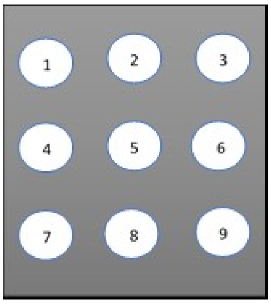


Figure 4.6: Pattern Numbering

```
                        }
}
```

Touch Data Acquisition

Touch data acquisition is done in the second phase. In this phase, users are provided with a pattern screen with nine spots (circles). Then each of these nine spots is assigned a number to identify the pattern. Figure 4.6 shows the numbers that have been assigned to each touch point.

An open source library called ”android-lockpattern” which is released under Apache Licensee v2.0 is used here to draw the pattern screen. Here `PatternUnolockActivity` is the responsive activity class that handles the second phase of data acquisition. So the pattern screen and the linear accelerometer is initialized during `onCreate` method call of the `PatternUnolockActivity`. The accelerometer initialization and registration is done as described in section 4.4.1.

In touch gesture acquisition process; spatial, timing and motion information of the user touch pattern will be taken. Once the user’s finger touched the edge of any circle, that circle will trigger and spatial,timing and motion information at that time will be recorded. Here touch size (touch area), Pressure and coordinates of the position are recorded as spatial information and timing information such as the time taken to enter each touch point also calculated by using the start time and the entry time.

Listing 4.8: Touch Data Acquisition

```
private void updateCellInfo(Cell hitCell, MotionEvent event)
    {
            if (hitCell != null) {
                    hitCell.x = event.getX();
                    hitCell.y = event.getY();
                    hitCell.area = event.getSize();
                    hitCell.pressure = event.getPressure();
                    hitCell.elapt = event.getEventTime()-
                        ptrnStart;
                    hitCell.user_id = getUser();
            }
}
```

Android OS API function calls are used here as shown in listing 4.9 to acquire spatial and timing information.

- **Pressure:** `MotionEvent.getPressure()` API call is used to obtain the touch pressure. This method returns the approximated force asserted upon each touch event. This value will be taken a value in the range 0 (soft touch) or 1 (hard touch). But this can be changed based on the device. Some devices return only either 0 or 1.

- **Touching Area:** This represents the contact area being touched which associated with a touch size. Therefore `MotionEvent.getSize()` android function is used to retrieve the touch size of each touch event.

- **Coordinates:** X and Y coordinates of the contact point are obtained by using the Android functions `MotionEvent.getX()` and `MotionEvent.getY()` with unit pixel (PX).

- **Elapsed Time:** Denotes the time taken to reach a particular point. This time is calculated by equation 4.3. Here the start time for the first node of the pattern is taken by `MotionEvent.getDownTime()` function call. This function returns the absolute start time of the touch event in milliseconds. Then the entry time which is also known as the event time is obtained from the `MotionEvent.getEventTime()` android function. It returns the time that the finger entered to the circle. Figure 3.6 depicts the computation of timing information.

  $$T_i = t_{i+1} - t_i \tag{4.3}$$

  Where

  - t_0 : is the absolute start time of the pattern.
  - t_i : is the time at node (i) on the unlock pattern.
  - t_{i+1} : is the time at node ($i+1$) on the unlock pattern.

In addition to spatial and timing information, accelerometer reading also is taken as motion information. Set-up and acquisition of accelerometer readings will be done as explained in section 4.4.1.

When storing pattern information, the encrypted pattern for each user will be stored in a separate file to be used later to validate the touch pattern. And raw touch dynamic data are stored as a comma separated list into a data file also in the cache.

Feature extraction from collected data is an important part of the proposed authentication method. This data pre-processing operation is done in the data preprocessing unit which is a part of the data collector module. Upon completion of one round of data collection, data collector will invoke pre-processor to extract features from collected raw data and store them into an internal database. Implementation of the pre-processor which include feature extraction is described in section 4.4.2.

4.4.2 Data Pre-processing

As described in the previous section, data pre-processor is invoked after each iteration of the data collection where one iteration include both face data acquisition and touch data acquisition phases. The result of the `PatternUnlockActivity` is monitored by the `FaceDetectionActivity.onActi` listener.

Listing 4.9: Touch Data Acquisition

```
@Override
    protected void onActivityResult(int requestCode, int
        resultCode, Intent data) {
          super.onActivityResult(requestCode, resultCode,
              data);
          if (resultCode == Constants.CustomStatus.RETRY) {

                trycount--;
                saveUserInfo();

                if ((mode == Mode.TRAIN || mode ==
                    Mode.REGISTER) && trycount == 0) {
                      bCancel.setEnabled(false);
                      mBtnOkCmd = Command.FINISH;
                      bOk.setText(R.string.finish);
                }

          } else if (resultCode ==
              Constants.CustomStatus.ERROR) {
                mMsg.setError(data.getExtras().getString(Constants.ERROR_
          }
    }
```

During the `FaceDetectionActivity.saveUserInfo()` method call, user's face and touch data get loaded from the cache and `FeatureExtractor` is invoked to extract important face and touch features from collected data.

Feature Extraction

The face feature vector consists of 16 features such as distance between eyes (DEYE), distance between left and right cheeks (DCHEEK), Width of the mouth (WMOUTH), distance between bottom mouth and nose (DMOUTHNOSE), distance between left eye and nose (DLEYENOSE), distance between right eye and nose (DREYENOSE), distance between left cheek and nose (DLCHEEKNOSE), distance between right cheek and nose (DRCHEEKNOSE), Acceleration along X axis (ACCELX), acceleration along Y axis(ACCELY), acceleration along Z axis (ACCELZ), Left eye open probability (LEO), right eye open probability(REO), smiling probability(SMILING), Euler Y(EY) and Euler Z(EZ). These face features are extracted from the raw data collected during the session and finally the average value of each extracted feature will be taken. The face data extraction process and calculations are described in section 3.3.2.

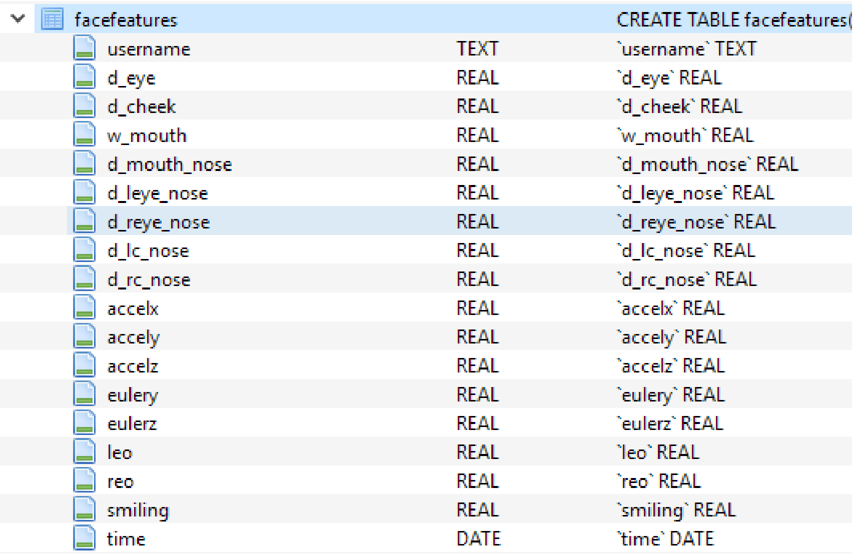
As shown in listing 4.10, the `FeatureExtractor.extractFaceFeatures` method is used for face feature extraction. This is an asynchronous task that runs in the background. Thus `AsyncListener` is used here to monitor the completion of the feature extraction process and upon completion of the task extracted face features are stored in the SQLite database through `SQLiteDSHelper`. These structured feature data are stored into ”facefeatures” table.

Listing 4.10: Feature Extraction

```
final String[] faces =
   Cache.getFacePatterns().split(System.lineSeparator());
         FeatureExtractor.extractFaceFeatures(new
            AsyncListener<Float>() {
             @Override
             public void callback(Float obj) {

             }

             @Override
             public void callback(Float[] list) {
                 if (list.length > 0) {
                     storeFaceFeatures(list, user);
                 }
             }
         }, faces);
```

Figure 4.7 contains the structure of the ”facefeatures” table.

A similar type of process is done for touch feature extraction. Here the touch feature vector (`TFV`) consists of 25 features. Those include X and Y coordinates for each nine touch points,



facefeatures		CREATE TABLE facefeatures(
username	TEXT	`username` TEXT
d_eye	REAL	`d_eye` REAL
d_cheek	REAL	`d_cheek` REAL
w_mouth	REAL	`w_mouth` REAL
d_mouth_nose	REAL	`d_mouth_nose` REAL
d_leye_nose	REAL	`d_leye_nose` REAL
d_reye_nose	REAL	`d_reye_nose` REAL
d_lc_nose	REAL	`d_lc_nose` REAL
d_rc_nose	REAL	`d_rc_nose` REAL
accelx	REAL	`accelx` REAL
accely	REAL	`accely` REAL
accelz	REAL	`accelz` REAL
eulery	REAL	`eulery` REAL
eulerz	REAL	`eulerz` REAL
leo	REAL	`leo` REAL
reo	REAL	`reo` REAL
smiling	REAL	`smiling` REAL
time	DATE	`time` DATE

Figure 4.7: SQLite Table Face Features

Average touch area per pattern (AVGAREA), Average touch pressure per pattern (AVGPRESSURE), Total elapsed time for pattern (TOTALELPTIME), Average finger moving speed (AVGMVSPEED), Average acceleration along X axis (AVGACCELX), Average acceleration along Y axis (AVGACCELY) and the Average acceleration along Z axis (AVGACCELZ). Extracted touch features are stored into ”touchfeatures” table.The table structure is shown in Figure 4.8.

4.4.3 Data Store

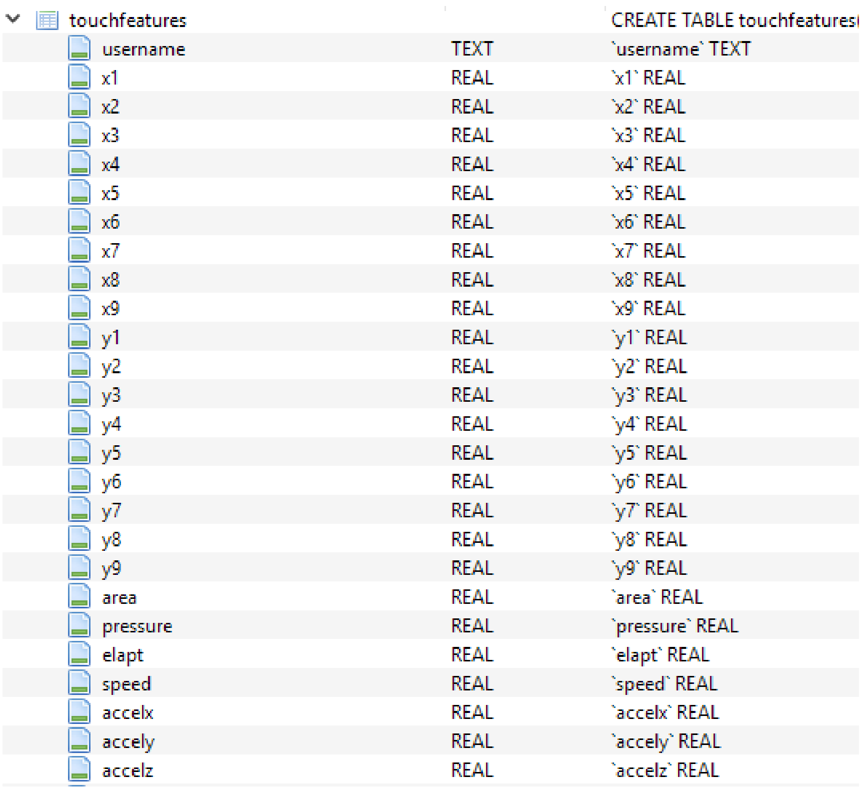
There are two types of data stores are used here to store data. One is internal data store where data stored in flat files. This internal data store is managed by InternalDSHelper which handles all the write ad read operations on internal data storage. The second data storage is SQLite database. This is managed by SQLiteDSHelper. It handles all the database operations like opening a connection, closing connection, insert, update, read, delete, etc.

Listing 4.11: Open SQLite Connection

```
public synchronized SQLiteDatabase getConnection(){
    return helper.getWritableDatabase();
}
```

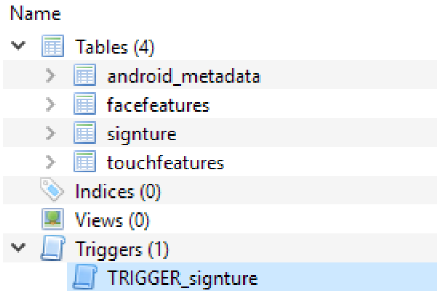
Here SQLiteOpenHelper.getWritableDatabase() method is used to open the database connection for both read and write operations. The first time this method is called, the database is created and connection is get opened by calling SQLiteDSHelper.onCreate(), SQLiteDSHelper.onUp and/or SQLiteDSHelper.onOpen() methods. Once the connection is opened successfully, the database is cached.

The database schema and structure and version are the key principles of using SQLite databases. DSContract is used to manage the layout of the schema. It contains names for URIs, tables,



touchfeatures		CREATE TABLE touchfeatures(
username	TEXT	`username` TEXT
x1	REAL	`x1` REAL
x2	REAL	`x2` REAL
x3	REAL	`x3` REAL
x4	REAL	`x4` REAL
x5	REAL	`x5` REAL
x6	REAL	`x6` REAL
x7	REAL	`x7` REAL
x8	REAL	`x8` REAL
x9	REAL	`x9` REAL
y1	REAL	`y1` REAL
y2	REAL	`y2` REAL
y3	REAL	`y3` REAL
y4	REAL	`y4` REAL
y5	REAL	`y5` REAL
y6	REAL	`y6` REAL
y7	REAL	`y7` REAL
y8	REAL	`y8` REAL
y9	REAL	`y9` REAL
area	REAL	`area` REAL
pressure	REAL	`pressure` REAL
elapt	REAL	`elapt` REAL
speed	REAL	`speed` REAL
accelx	REAL	`accelx` REAL
accely	REAL	`accely` REAL
accelz	REAL	`accelz` REAL

Figure 4.8: ”touchfeatures” SQLite Table



Name

- Tables (4)
  - android_metadata
  - facefeatures
  - signture
  - touchfeatures
- Indices (0)
- Views (0)
- Triggers (1)
  - TRIGGER_signture

Figure 4.9: SQLite Database

columns etc.

Figure 4.9 depicts the structure of the DB schema.

4.5 Data Classifier

Data classifier is the second module of the application which handles the user categorization and classification based on the extracted features. So as the first step, the classifier will generate a signature (template) from extracted features to be used later to identify the user. But if consider the size of the both face feature vector and touch feature vector, it is undesirable to use all of the available features to perform the classification. Usually, it is expected that more the number of features being used to represent the subject’s behavior, should increase the accuracy of the classifier. However, it has become unrealistic due to following reasons.

- High CPU usage: Using more features to train and test the classifier may cause in more processing time and CPU overhead which need to be taken into consideration because of this authentication method is focused on mobile devices.
- High memory utilization: Storing and handling more features may result in more memory usage.
- Discriminative capability: All the features do not have same discriminative capabilities where some of them have higher influence than others.
- Over-fitting: Using all features for classification may result in over-fitting to the training data set.

To address above problems, feature selection process is done before classification take place. This process is done manually by using the WEKA tool. The process of feature selection is described in next section.

4.5.1 Feature Selection

The purpose of having a feature selection process is to choose a smaller subset of features from feature set that can be used to represent the subject's behavior. Adequate selection of features may result in higher accuracy and efficiency of classifier methods. As explained in section 3.4.3, there are three main approaches for feature selection such as filter method, wrapper method and embedded method. Wrapper method will select features by using a classifier and it may obtain better performance with greater computational resources. The filter method performs selection of features independently from the classifier used. By considering the behavior of each approach and the requirement of this project, filter method is selected to be used for feature selection.

Filter method uses a statistical approach to rank features. First, it applies a statistical measure to compute n scores and then rank them by sorting the scores. Thus this method considered to be computationally efficient. Later these ranks can be used either to keep or remove low ranked features from data set. Chi Squared test, Information Gain, Gain Ratio and Correlation Coefficient are some examples of filter methods.

Information gain is one of the popular feature selection technique. Information gain (also called entropy) is calculated for each attribute for the output variable. This value can vary from 0 to 1 where 0 denotes no information and 1 denotes maximum information. Attributes contribute more information will gain a higher rank comparatively to the attributes which contribute less information. In WEKA, ”InfoGainAttributeEval” can be used to feature selection via information gain. Here Ranker search method must be used together with information gain evaluator.

The purpose of using Ranker method is to rank attributes by their individual evaluations. This method can be used in conjunction with (ReliefF, GainRatio, Entropy and etc.).

But information gain technique exhibits a bias towards attributes having a large number of values. Thus an extension known as the gain ratio has introduced to overcome the bias of information gain. Gain ratio technique applies a kind of normalization to information gain using a split information value. The gain ratio is defined as equation 4.4.

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)} \tag{4.4}$$

where A is the selected attribute and $Gain(A)$, $Info(D)$ and $Info_A(D)$ obtained by equation 4.5,4.6 and 4.7 respectively.

$$Gain(A) = Info(D) - Info_A(D) \tag{4.5}$$

$$Info(D) = -\sum_{i=1}^{m} p_i \log(p_i) \tag{4.6}$$

- m: number of classes
- p_i: the probability that an arbitrary tuple in D belongs to class C_i; estimated as $|C_{i,D}|/D$.

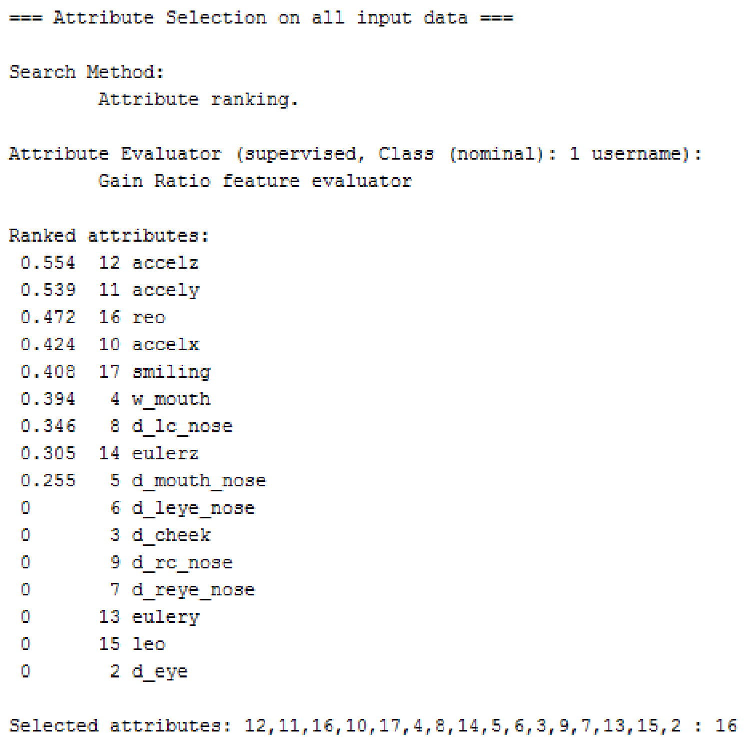
$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} Info(D_j) \tag{4.7}$$

- v: number of partitions
- $|D_j|/D$: weight of the j^{th} partition
- $Info(D_j)$: the entropy of partition D_j.

Therefore ”Gain Ratio” was selected as the preferred technique for feature selection. WEKA supports feature selection via gain ratio through ”GainRatioAttributeEval” evaluator. As same as for information gain evaluator, using Ranker search method is mandatory for the gain ratio as well. Figure 4.10 and 4.11 depict the face feature selection and touch feature selection results respectively.

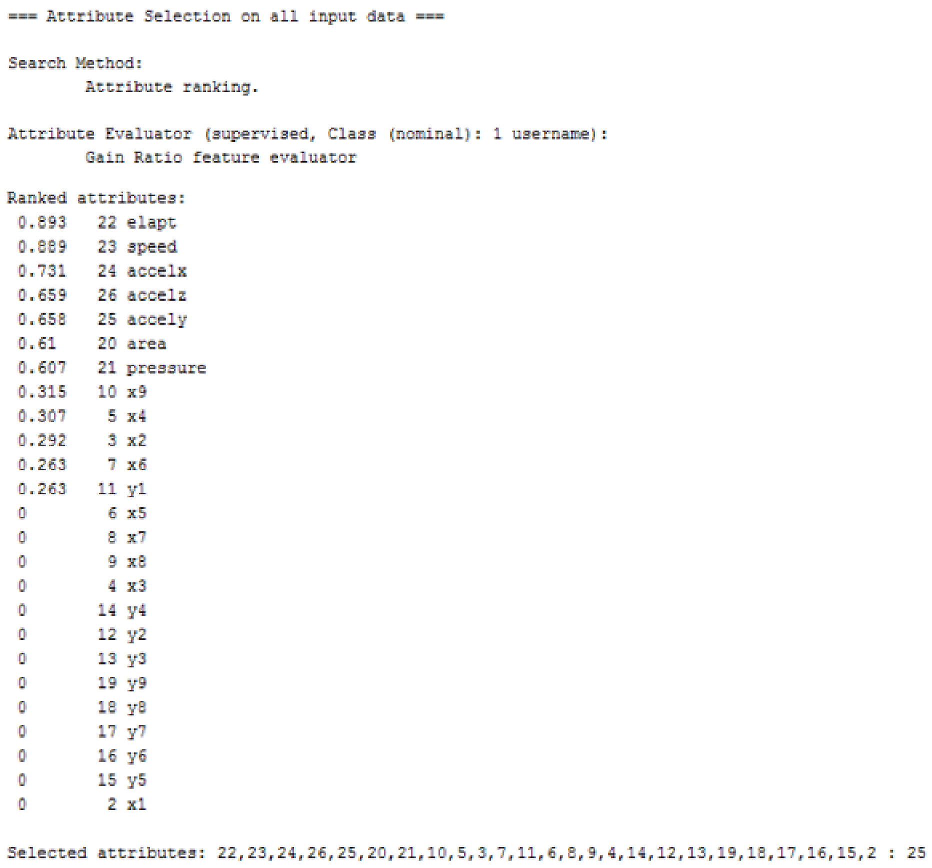
4.5.2 Classification

The statistical model is selected to perform the user classification. The main purpose of this operation is to generate a unique template (signature) to distinguish each user. Within this signature generation, it will transform above extracted features into a compact form which is called



```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 1 username):
        Gain Ratio feature evaluator

Ranked attributes:
 0.554  12 accelz
 0.539  11 accely
 0.472  16 reo
 0.424  10 accelx
 0.408  17 smiling
 0.394   4 w_mouth
 0.346   8 d_lc_nose
 0.305  14 eulerz
 0.255   5 d_mouth_nose
 0       6 d_leye_nose
 0       3 d_cheek
 0       9 d_rc_nose
 0       7 d_reye_nose
 0      13 eulery
 0      15 leo
 0       2 d_eye

Selected attributes: 12,11,16,10,17,4,8,14,5,6,3,9,7,13,15,2 : 16
```

Figure 4.10: Face Feature Selection



```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 1 username):
        Gain Ratio feature evaluator

Ranked attributes:
 0.893   22 elapt
 0.889   23 speed
 0.731   24 accelx
 0.659   26 accelz
 0.658   25 accely
 0.61    20 area
 0.607   21 pressure
 0.315   10 x9
 0.307    5 x4
 0.292    3 x2
 0.263    7 x6
 0.263   11 y1
 0        6 x5
 0        8 x7
 0        9 x8
 0        4 x3
 0       14 y4
 0       12 y2
 0       13 y3
 0       19 y9
 0       18 y8
 0       17 y7
 0       16 y6
 0       15 y5
 0        2 x1

Selected attributes: 22,23,24,26,25,20,21,10,5,3,7,11,6,8,9,4,14,12,13,19,18,17,16,15,2 : 25
```

Figure 4.11: Touch Feature Selection

user signature or template. In the statistical method, user signature is generated by concatenating the sample mean $\bar{x}$, sample standard deviation s, and $[L,U]$ confidence intervals for each feature.

If there are m number of features, then;

Signature = ”$\bar{x}_1, s_1, L_1, U_1 | \bar{x}_2, s_2, L_2, U_2 | \bar{x}_3, s_3, L_3, U_3 | ..... | \bar{x}_m, s_m, L_m, U_m$”

Signature generation is handled in `generateUserSignature` method (listing 4.12).

Listing 4.12: Generate User Signature

```
public void generateUserSignature(final String user) {
        Log.d(CLASSNAME, "Generate User Signature");
        AsyncTask<String, Void, Void> execute = new
           AsyncTask<String, Void, Void>() {
                @Override
                protected Void doInBackground(String...
                   params) {

                        final String user = params[0];

                // calculate mean
                float[] mean_f = getMeanFF(user);
                float[] mean_t = getMeanTF(user);

                // get feature matrix
                final float[][] fmatrix =
                   readFeatureMatrix(user);
                final float[][] tmatrix =
                   readTouchFeatureMatrix(user);
                // calculate sd & cis
                if (mean_f != null && mean_t != null &&
                   fmatrix != null && tmatrix != null) {
                    float[] sd_f = standardDeviation(mean_f,
                       fmatrix);
                    float[] sd_t = standardDeviation(mean_t,
                       tmatrix);

                    float[][] cc_f =
                       confidenceInterval(mean_f, sd_f);
                    float[][] cc_t =
                       confidenceInterval(mean_t, sd_t);
                    // Generate signatures
                    final String face_sign =
                       getSignature(mean_f, sd_f, cc_f);
                    final String touch_sign =
                       getSignature(mean_t, sd_t, cc_t);

                    // insert to DB
                    insertIntoDB(user, face_sign,
                       touch_sign);
                }
                return null;
                }

                @Override
                protected void onPostExecute(Void result) {
                        super.onPostExecute(result);
                }
        }.execute(user);

}
```

Before starting signature generation process it will load face feature matrix (M_f) and touch feature matrix (M_t) that stored into the database by pre-processor. Then it will calculate the mean vector for both M_f and M_t matrices which are v_{meanF} and v_{meanT} by using equation 3.1. Then the sample standard deviation s for both face and touch data collections are calculated by using equation 3.2.Finally confidence interval is calculated using student t distribution function depicts in 3.4. Here t-distribution function is used because of the standard deviation for the population is unknown.

Two templates/signatures will be generated per each user.One is generated by face features and other is by using touch features and these two signatures are stored in the database.

4.6 Validator

The validation module will perform a comparison between current user signature against with the target user's signature.The decision is made based on a predefined threshold value. However, this project apply fusion approach where combine two matching scores from different feature sets such as touch dynamics and facial features to make the final decision.

There are two validation approaches such as verification and identification. When considering the requirement of mobile application authentication, verification approach was selected in this project. In verification mode, login user is verified against the information of target user to make sure it is the same user that claimed to be.

Validator module is used only in user authentication mode and it is invoked upon user enter their credentials on application's login page. First, it will load target user's signatures from the database and will perform verification for each feature to check whether the new value belongs to the confidence interval of a particular feature. Then weight will apply to matches before calculating the dissimilarity score. These weights for each feature are obtained from the scores given by feature selection algorithm. Finally, two dissimilarity score values will be calculated and the decision is made with respect to the threshold value determined for each feature sets.

At the end of each successful user authentication, the database is updated to add a new feature set to enhance the accuracy of the authentication system.

4.7 Summary

This chapter discussed the implementation of the mobile application with including descriptive information for each of the three main modules. Basically, implementation is done for Android platform by using Android Studio and Java JDK 1.7. Besides Android Studio, WEKA and Octave were used as analytical tools for data analysis.

The main purpose of implementing this android application is to collect training data set and perform the evaluation. Each of these modules consists of background processes to process data that corresponding to the module and result is saved into either flat file or database. Then successive module will take required data from either database or cache. Here caching mechanism is use on top of the database to increase the performance. But while using cache memory usage also taken into account because memory usage is an important factor in mobile devices.

Finally, validator module makes the decision by combining the result of both face verification and touch verification processes. After each successful iteration, database updated with a feature to retrain the classifier in order to increase the accuracy.

Next chapter will include analytical information of this approach.

Chapter 5

Result and Evaluation

5.1 Overview

The previous chapter covered the implementation of the project with including technologies and tools that have been used. This chapter will include result and data analysis information.

5.2 Outlier Detection

Before performing feature selection, it is important to identify potential outliers and their impact on the final outcome. By definition, an outlier is an observation point that varies from other observations. Outliers may indicate bad data and may need to exclude from the data set. But in some cases, it is not possible to determine if an outlier point is bad data because there is a possibility that they may indicate something scientifically interesting. Therefore if the data contains significant outliers, it is recommended to consider the use of the robust statistical technique.

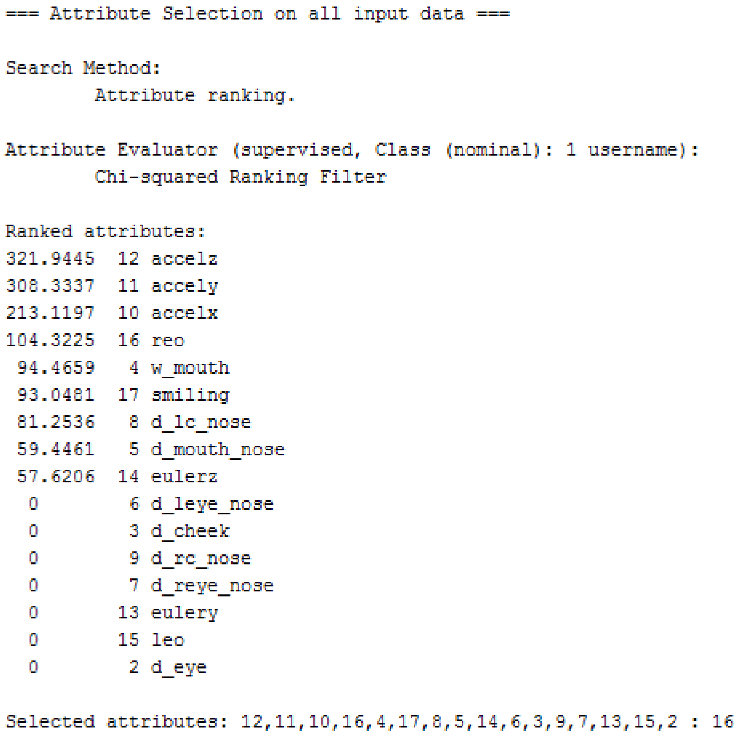
5.3 Feature Selection

Basically, the focus of machine learning techniques is to obtain an approximate relationship between an input feature vector $X_1,X_2,X_3,...X_N$ and output Y. Here N denotes the number of features. Usually, whole features set is not used to determine Y to avoid some issues like over-fitting issue and misleading issue. Instead, the output is decided only by a subset of most relevant features. The selection of this most relevant subset of features is called the feature selection process.

Hence feature selection was performed for both face and touch feature vectors to obtain the most relevant subset of features to used in the classification model. There are two objectives of performing feature selection in this project. One is to find the most relevant subset of features and other is to assign some weight to each feature based on their relevance to the output. Therefore filter method is used as the feature selection technique and it is performed by using WEKA tool which is an open source tool popular for data analysis in machine learning domain.

Filter Method There are four types of filter methods could be found in the literature such as Chi Squared Ranking Filter, Correlation Coefficient, Information Gain and Gain Ratio.

Figure 5.1 and Figure 5.2 depict the result of Chi Squared Ranking Filter. The `weka.attributeSelecti` evaluator is used to evaluate the worth of an attribute by computing the value of the chi-squared statistic with respect to the class.



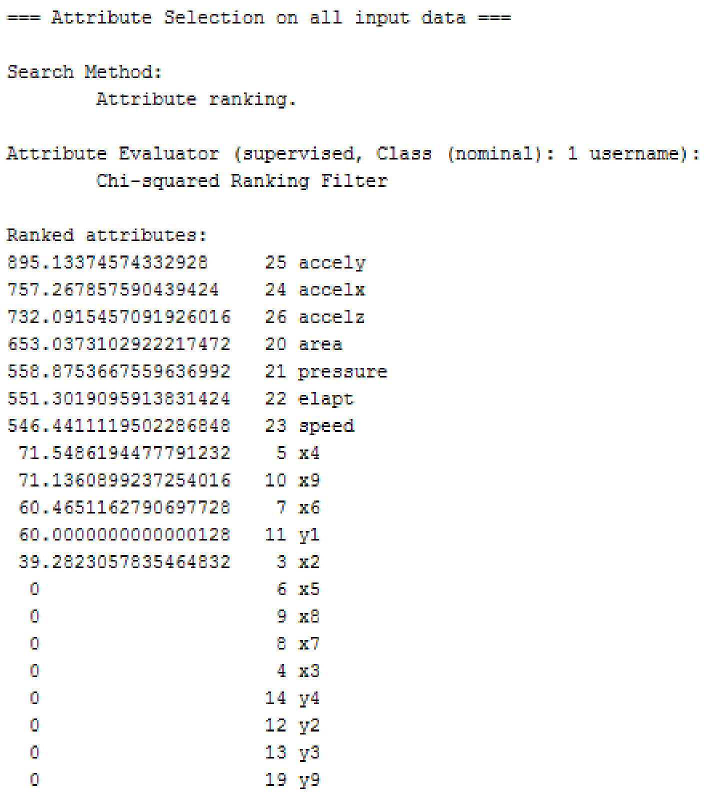
```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 1 username):
        Chi-squared Ranking Filter

Ranked attributes:
321.9445  12 accelz
308.3337  11 accely
213.1197  10 accelx
104.3225  16 reo
 94.4659   4 w_mouth
 93.0481  17 smiling
 81.2536   8 d_lc_nose
 59.4461   5 d_mouth_nose
 57.6206  14 eulerz
  0        6 d_leye_nose
  0        3 d_cheek
  0        9 d_rc_nose
  0        7 d_reye_nose
  0       13 eulery
  0       15 leo
  0        2 d_eye

Selected attributes: 12,11,10,16,4,17,8,5,14,6,3,9,7,13,15,2 : 16
```

Figure 5.1: Chi Squared Ranking Filter Results for Face Features

Figure 5.3 and Figure 5.4 show correlation coefficient based feature selection algorithm results. It evaluated the worth of a subset of attributes based on the individual predictive ability of each feature along with the degree of redundancy between them. Furthermore ”bestfirst” method is used as the search method where it searches the subset of attributes by using greedy hillclimbing augmented with backtracking facility. There are three types of operation modes in best-first technique. One is to start with the empty set of attributes and search forward and second is to start with the full set of attributes and search backward or search will start at any point and search in both directions by considering all possible single attribute additions and deletions. But this method does not return ranking values for selected attributes.

Next filter method is information gain. Figure 5.5 and 5.6 illustrate the result of face feature selection result and touch feature selection results respectively.

The last filter method to be analysed is the Gain Ratio. The Gain ratio is an extension to information gain filter method which is introduced to overcome the bias of information gain.



```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 1 username):
        Chi-squared Ranking Filter

Ranked attributes:
895.13374574332928     25 accely
757.267857590439424    24 accelx
732.0915457091926016   26 accelz
653.0373102922217472   20 area
558.8753667559636992   21 pressure
551.3019095913831424   22 elapt
546.4411119502286848   23 speed
 71.5486194477791232    5 x4
 71.1360899237254016   10 x9
 60.465116279069772 8    7 x6
 60.000000000000128    11 y1
 39.2823057835464832    3 x2
  0                     6 x5
  0                     9 x8
  0                     8 x7
  0                     4 x3
  0                    14 y4
  0                    12 y2
  0                    13 y3
  0                    19 y9
```

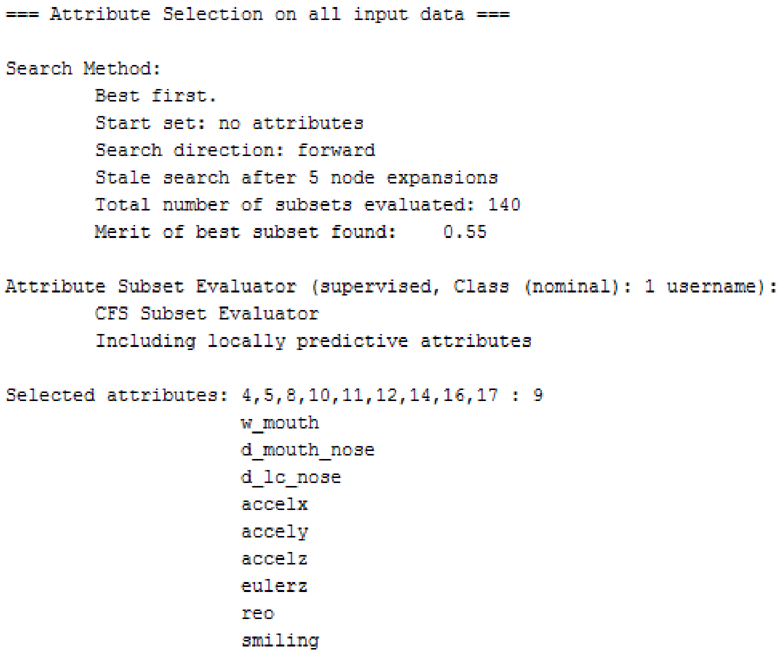
Figure 5.2: Chi Squared Ranking Filter Results for Touch Features

Gain ratio technique applies a kind of normalization to information gain using a split information value. Figure 5.7 and 5.8 contain the ranking results for both face feature set and touch feature set.

By considering the facts that mentioned in the previous chapter we will use Gain Ratio technique as the preferred feature selection method to select the subset of features that will be used for user classification and validation processes and finally authenticate the user. Therefore accelZ (acceleration force along Z axis, AccelY (acceleration force along Y axis), reo (right eye open probability), accelX (acceleration force along X axis), smiling (smiling probability), w_mount (width of mouth), d_lc_nose (distance between left cheek and nose), eulerZ (rotation around Z axis), d_mouth_nose (distance between bottom mouth and nose base) are selected as the subset of face feature vector. Also elapt(elapsed time), speed (finger moving speed), accelX (acceleration force along X axis), accelZ (acceleration force along Z axis), accelY (acceleration force along Y axis), area (touch area) and pressure (touch pressure) were selected as the subset of touch features that have more impact on the final outcome.

5.4 Data Analysis

User classification is done based on the statistical method proposed in previous chapters. Only selected features are considered while performing the classification. The first behavior of each feature is analyzed using WEKA tool. Figure 5.9 illustrate the result of the analysis.



```
=== Attribute Selection on all input data ===

Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 140
        Merit of best subset found:    0.55

Attribute Subset Evaluator (supervised, Class (nominal): 1 username):
        CFS Subset Evaluator
        Including locally predictive attributes

Selected attributes: 4,5,8,10,11,12,14,16,17 : 9
                     w_mouth
                     d_mouth_nose
                     d_lc_nose
                     accelx
                     accely
                     accelz
                     eulerz
                     reo
                     smiling
```

Figure 5.3: Correlation Coefficient Filter Results for Face Features

This illustrates the behavior of features for all the face data set. Next Figure 5.10 illustrate the behavior of touch data set.

Furthermore, behavior of selected attributes will be analyzed in detail in next section.

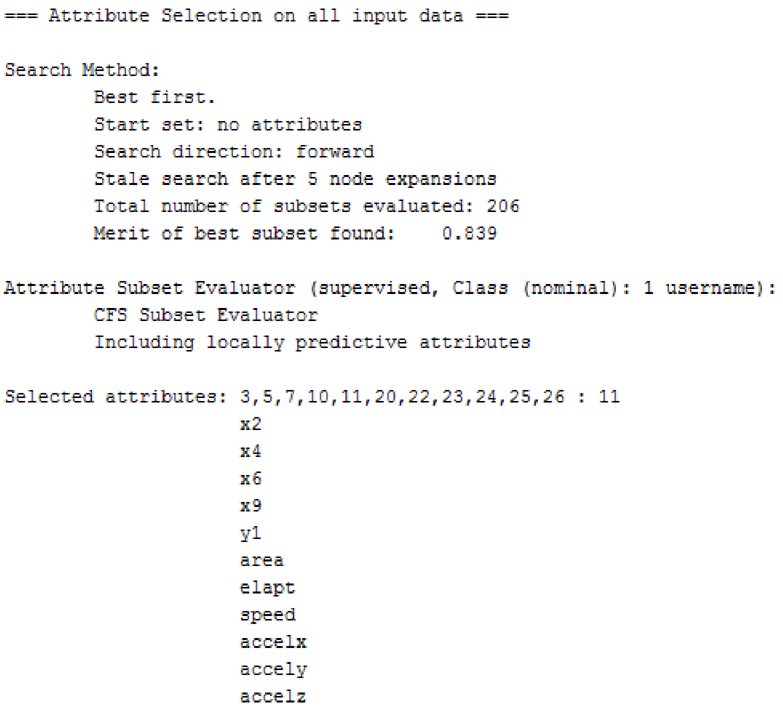
5.4.1 Face Feature Analysis

Initially, face feature vector contained 16 features and 9 features were selected by applying filter method feature selection technique. Here these each feature values were plotted against each user to understand their behavior. These plots were created using WEKA tool.

AccelZ (acceleration force along the Z axis) is the highest ranked feature among the selected nine features. It shows a significant difference between each user data set. In the same way two graphs have been plotted for all the other selected features. The second highest ranked feature is acceleration along the Y axis. Figure 5.12 show the variation of accelY values.

In plots referred in 5.12, the first user exhibits a considerable difference than other users. Also if consider average values it is clear that each value for different users is slightly differ from other users. Right eye open probability is the next in select feature subset. Figure 5.13 depict the behavior of this feature for different users.

The feature äccelxïs ranked into the 4th place according to Gain Ratio ranking technique. As illustrate in Figure 5.14 the value of mobile device acceleration force along the X axis, lies



```
=== Attribute Selection on all input data ===

Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 206
        Merit of best subset found:    0.839

Attribute Subset Evaluator (supervised, Class (nominal): 1 username):
        CFS Subset Evaluator
        Including locally predictive attributes

Selected attributes: 3,5,7,10,11,20,22,23,24,25,26 : 11
                     x2
                     x4
                     x6
                     x9
                     y1
                     area
                     elapt
                     speed
                     accelx
                     accely
                     accelz
```

Figure 5.4: Correlation Coefficient Filter Results for Touch Features

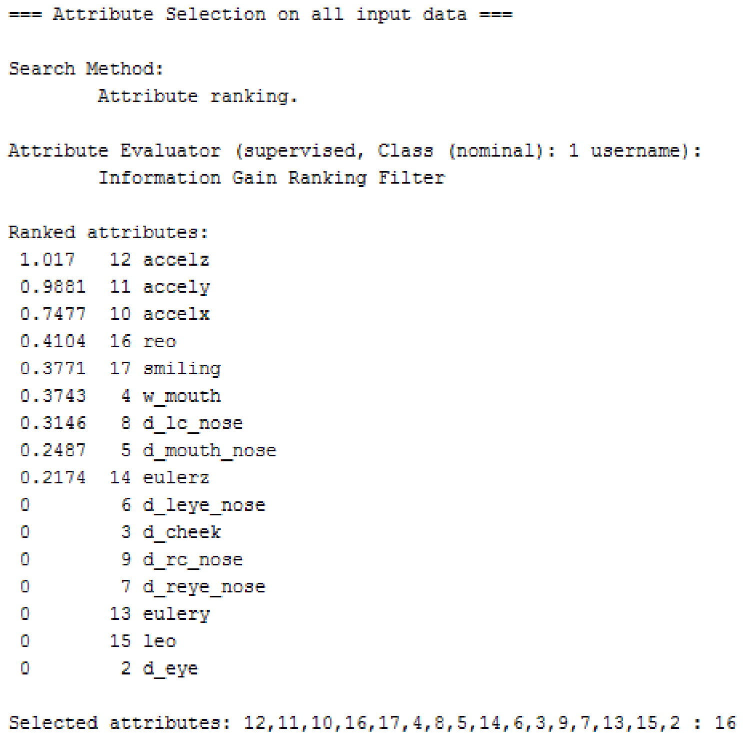
within long range when comparing to previous features and average value exhibit significant difference among each different users.

Smiling probability is a classified value return from Google^{TM}Services API. This is the next ranked feature in Gain Ratio feature selection results list. It contains both negative and positive values and exhibits significant difference among different users.

Figure 5.16 depicts the behavior of the feature ”width of mouth” for different users. According to the first plot, it is clear that there are some values which deviate from the cluster. The second plot contains the average width of mouth that calculated for each user. Here other users except third, fifth, seventh and ninth users exhibit significant difference. Therefore it is clear that this feature can be considered as a good candidate to use for user verification. But according to the ranking algorithm this feature is ranked into the sixth position. Therefore the affect of this feature on final outcome may be less than previously discussed features.

The distance between left cheek and nose have been plotted against different 10 users in Figure 5.17. As it illustrates in the figure values exhibit slight differences for different users.

The ËulerZänd D̈istance between the mouth and nosef̈eatures are the last in the selected feature list. In Figure 5.18, left hand side plot contains the raw extracted value plotted against each users. Here it is clearly illustrated that most of the values are converge to a single point and only a few values have been diverging from the center. When comparing with the plot that plot average value against each user, we can observe that outliers have made a considerable



```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 1 username):
        Information Gain Ranking Filter

Ranked attributes:
 1.017   12 accelz
 0.9881  11 accely
 0.7477  10 accelx
 0.4104  16 reo
 0.3771  17 smiling
 0.3743   4 w_mouth
 0.3146   8 d_lc_nose
 0.2487   5 d_mouth_nose
 0.2174  14 eulerz
 0        6 d_leye_nose
 0        3 d_cheek
 0        9 d_rc_nose
 0        7 d_reye_nose
 0       13 eulery
 0       15 leo
 0        2 d_eye

Selected attributes: 12,11,10,16,17,4,8,5,14,6,3,9,7,13,15,2 : 16
```

Figure 5.5: Information Gain Filter Results for Face Features

impact on the average value. Same can be seen in the Figure 5.19 which depicts the behavior of distance between mouth and nose.

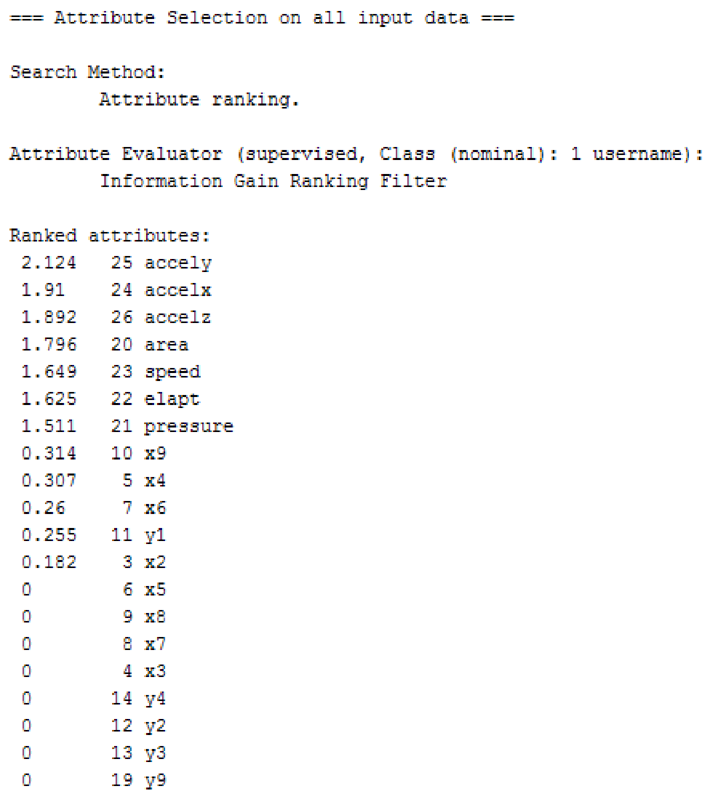
5.4.2 Touch Feature Analysis

The behavior of face features was analyzed in the previous subsection. Both raw values and average values of extracted features have been plotted against 10 different users to do the analysis. The same method will be used here for touch feature analysis and it is done based on the feature selection result list.

Among touch features, ”acceleration along the X-axis” is the highest ranked feature. Thus the behavior of touch accelX feature and average values are plotted in Figure 5.20. As shown in Figure 5.20, the behavior of acceleration along X axis feature for some users exhibits steady behavior than others. Also the average values per each user exhibits more diversity which can be useful in distinguishing different users.

The next feature is the elapsed time. Here total time that user spent to draw complete pattern was taken as the total elapsed time. Average value is calculated for all collected 20 samples per each user. Figure 5.21 depicts the behavior of raw elapsed time and average elapsed time for different 10 users.

The feature acceleration along the Z axis is the third ranked feature in the selected feature list. The plot on the left hand side in Figure 5.22 illustrates the behavior of raw values while



```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 1 username):
        Information Gain Ranking Filter

Ranked attributes:
 2.124   25 accely
 1.91    24 accelx
 1.892   26 accelz
 1.796   20 area
 1.649   23 speed
 1.625   22 elapt
 1.511   21 pressure
 0.314   10 x9
 0.307    5 x4
 0.26     7 x6
 0.255   11 y1
 0.182    3 x2
 0        6 x5
 0        9 x8
 0        8 x7
 0        4 x3
 0       14 y4
 0       12 y2
 0       13 y3
 0       19 y9
```

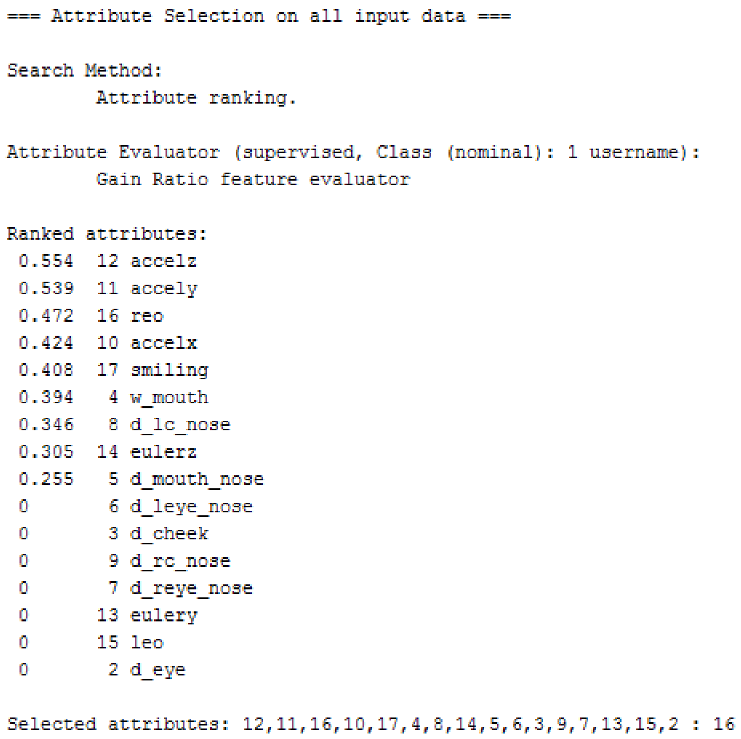
Figure 5.6: Information Gain Filter Results for Touch Features

the right hand side plot depicts the behavior of average values. The next Figure 5.23 illustrate the behavior of acceleration along Y axis for the 10 different users. According to the average graphs, it is clear that there is a significant difference in acceleration among different users. Therefore acceleration values can be considered as good candidates for user identification.

The next important feature is the finger moving speed. Here the speed between each node was calculated and average speed has been taken as the speed value for each iteration. The Figure 5.24 contains two plots that illustrate the variation of raw speed data average speed data for different users.

The touch area and pressure are two other important touch features and Figure 5.25 and 5.26 contains the behavior of area and pressure feature for different users respectively. Moreover if closely observe the behavior of these two features, values are converged around two different points for some users. When analyzing data it was clear that area and pressure corrected in two different sessions may diverge from the values from another session. Therefore by increasing the number of session for collect data may helpful to increase the performance.

After feature selection and analysis, next step is to apply classification algorithm. As explained in previous chapters classification algorithm is made up from the statistical method which uses mean, variation and confidence intervals. Next section will include the analysis of the classification and verification results.



```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 1 username):
        Gain Ratio feature evaluator

Ranked attributes:
 0.554  12 accelz
 0.539  11 accely
 0.472  16 reo
 0.424  10 accelx
 0.408  17 smiling
 0.394   4 w_mouth
 0.346   8 d_lc_nose
 0.305  14 eulerz
 0.255   5 d_mouth_nose
 0       6 d_leye_nose
 0       3 d_cheek
 0       9 d_rc_nose
 0       7 d_reye_nose
 0      13 eulery
 0      15 leo
 0       2 d_eye

Selected attributes: 12,11,16,10,17,4,8,14,5,6,3,9,7,13,15,2 : 16
```

Figure 5.7: Gain Ratio Filter Results for Face Features

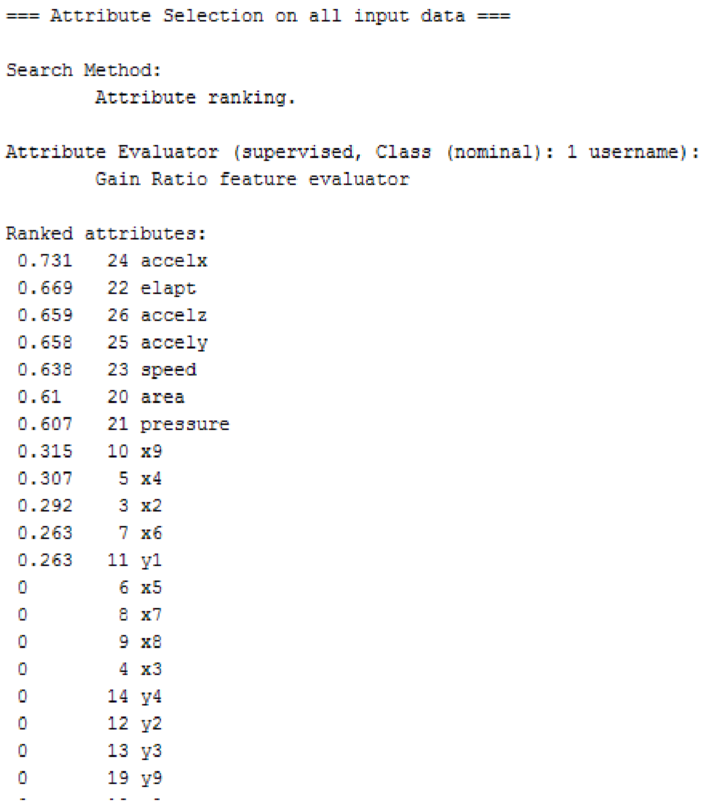
5.5 Classification & Verification

A statistical method is used here as the basis of the classification algorithm. According to the proposed algorithm confidence interval is used as the range to classify each feature. The main advantage of using this algorithm is it is not required keep lots of training data in the mobile application, but required only the particular user's data which is acquired during the user registration. First, classifier need be trained to classify users accurately. Thus threshold values are determined for face and touch feature set by using the acquired dataset for 10 different users.

In order to test and evaluate the proposed classification algorithm, 70% of samples from each user are taken as the training dataset and remainder (30%) are taken as testing dataset which will be used for the evaluation. In collected dataset, each user has 20 samples. Thus 14 randomly selected samples were taken as the training data set and the remainder is used for the testing.

Training During the training phase; mean, standard deviation and low boundary and the upper boundary of the confidence interval were calculated for each user and each feature. Following six confidence levels listed in Table 5.1 were taken into consideration. As the test sample size for each user is 14, the degree of freedom is 13 and thus confidence level that corresponds to d.f. 13 was considered. Refer appendix A for more details.

Then overall success rate is calculated for each confidence interval by using different threshold values. These threshold values were calculated using the weights that obtained from feature



```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 1 username):
        Gain Ratio feature evaluator

Ranked attributes:
 0.731    24 accelx
 0.669    22 elapt
 0.659    26 accelz
 0.658    25 accely
 0.638    23 speed
 0.61     20 area
 0.607    21 pressure
 0.315    10 x9
 0.307     5 x4
 0.292     3 x2
 0.263     7 x6
 0.263    11 y1
 0         6 x5
 0         8 x7
 0         9 x8
 0         4 x3
 0        14 y4
 0        12 y2
 0        13 y3
 0        19 y9
```

Figure 5.8: Gain Ratio Filter Results for Touch Features

Confidence level c (%)	**t value**
80	1.3502
90	1.7709
95	2.1604
98	2.6503
99	3.0123
99.9	4.2208

Table 5.1: t-Distribution Confidence Intervals for d.f 13

selection algorithm. Weight for selected features is shown in Table 5.2.

Then threshold value for $X\%$ success rate is calculated by using Equation 5.1. In order to analyze the success and failure rates, seven threshold values are calculated as shown in Table 5.3.

$$T = 3.697 * \frac{X}{100} \tag{5.1}$$

As shown in Table 5.4 success rate is calculated. The success rate means the percentage of identifying a legitimate user accurately.

In this approach, identifying most relevant confidence interval and the threshold value is more important. The false acceptance rate (FAR) and false rejection rate (FRR) are used to select the

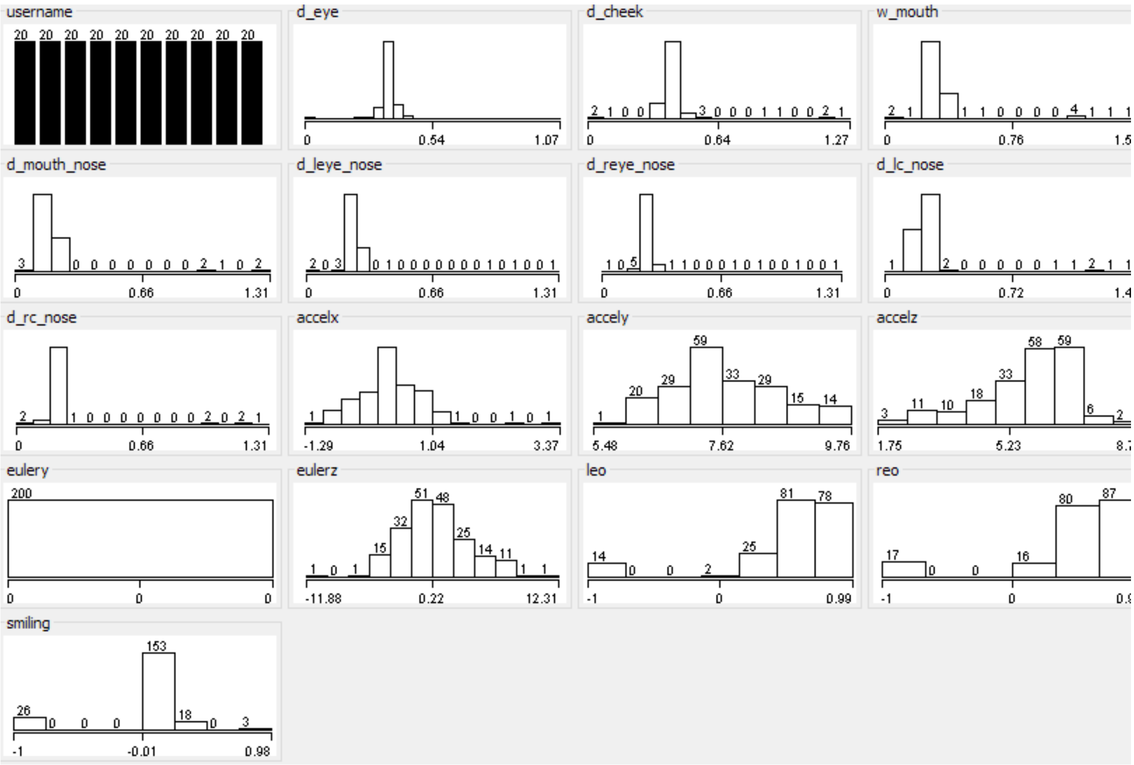


Figure 5.9: Face Feature Analysis

relevant confidence interval and threshold value because FRR and FAR can be used ensure the accuracy and user acceptance of the proposed method which is the main goal of this project. The calculation of FAR and FRR for each confidence intervals are shown in Table 5.5 and 5.6 respectively.

The FAR rate is calculated by selecting a one user and assume that other nine users are trying to access this selected user's account. Here FAR is calculated as the same way we calculate the acceptance rate for legitimate users and will check how many attempts were falsely accepted as the legitimate users. This process is executed for each user (10) and obtained the average acceptance rate as the FAR.

FRR is the inverse of the success rate for each user. So FRR values were easily calculated by using the true acceptance rates which include in Table 5.4.

As shown in Figure 5.27, false acceptance rates are plotted in a single graph to identify the behavior of different confidence intervals. Here threshold value id has been taken as the X axis and percentage value (FAR) as the Y axis.

According to the Figure 5.27, it is clear that when confidence interval is higher, the FAR is also got increased. Therefore it implies that in order to reduce false acceptance rate we need to select less confidence interval which is more converge to the center of the distribution. But this information is not sufficient enough to make the decision. Thus FRR also graphed as shown

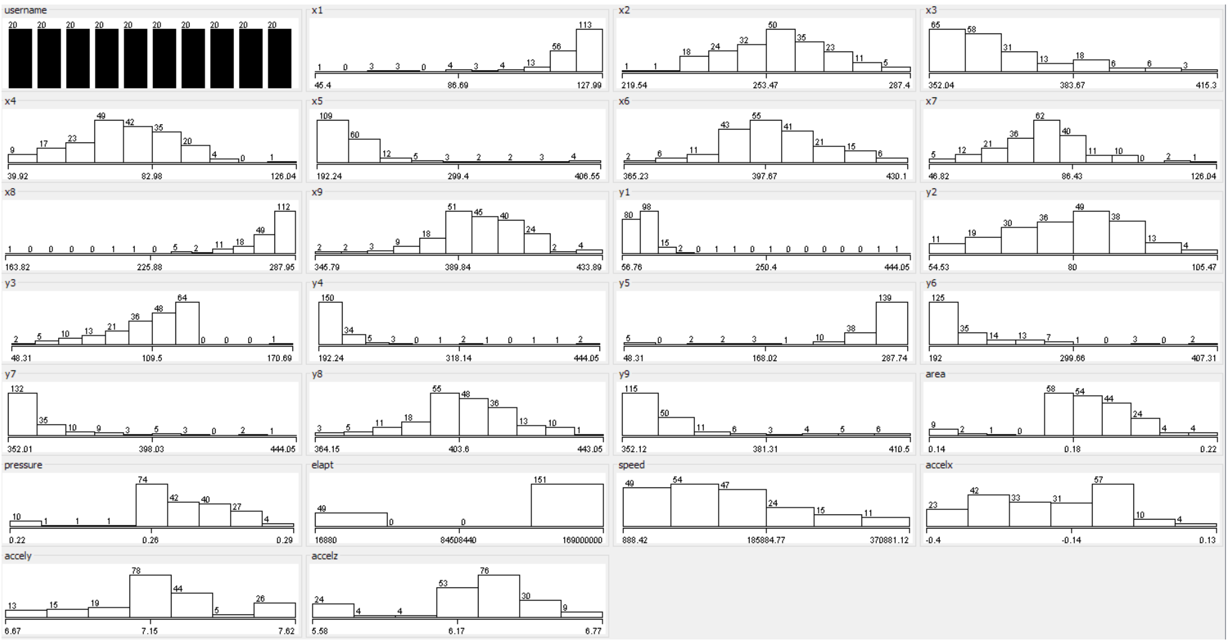


Figure 5.10: Touch Feature Analysis

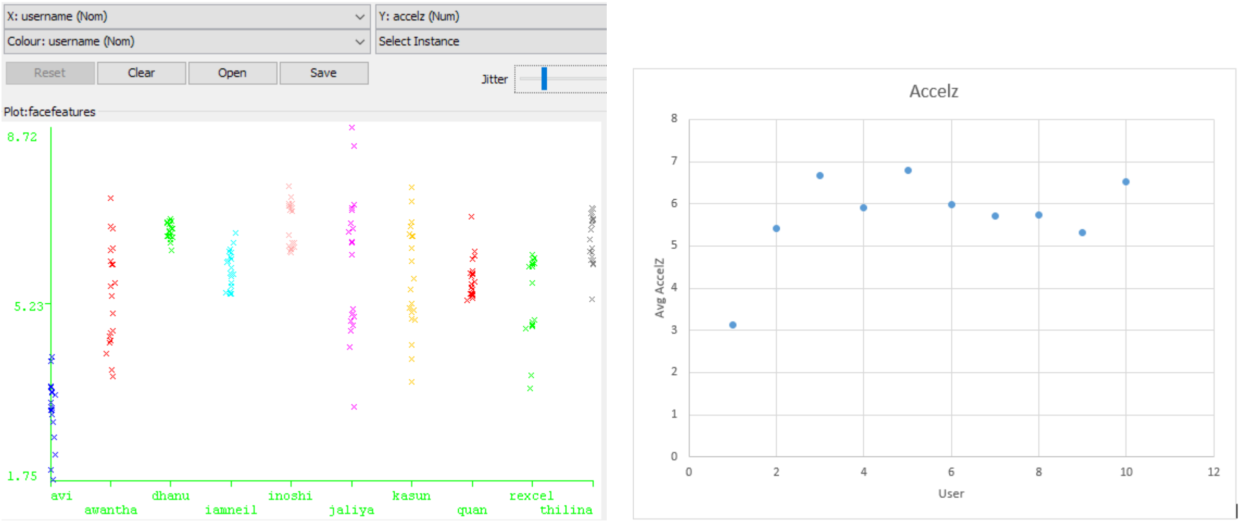
Feature	Weight
AccelZ	0.554
AccelY	0.539
REO	0.472
AccelX	0.424
Smile	0.408
W-MOUTH	0.394
D-LC-NOSE	0.346
EulerZ	0.305
D-MOUTH-NOSE	0.255
Total	**3.697**

Table 5.2: Face Feature Weights

in Figure 5.28 by using a threshold as X axis and FRR as Y axis. In this graph, it shows that when confidence interval takes a less value the false rejection rate has gone up. Hence the evaluation metrics EER (equal error rate) which is used to measure the accuracy of computer authentication methods need to used here as well to select the best confidence interval and the corresponding threshold value.

Basically the equal error rate (EER) metric is the interception of FAR and FRR graphs. This metric is highly used to evaluate the accuracy of different authentication methods. The accuracy of authentication method is inversely proportional to the value of EER which means lower the EER, the accuracy of the particular method will be high. Figure 5.29 depicts the EER values for different confidence intervals.

According to the Figure 5.29, the equal error rate takes lower value when both confidence interval and the threshold value is higher. Thus in this study, we will use 99.9 confidence interval

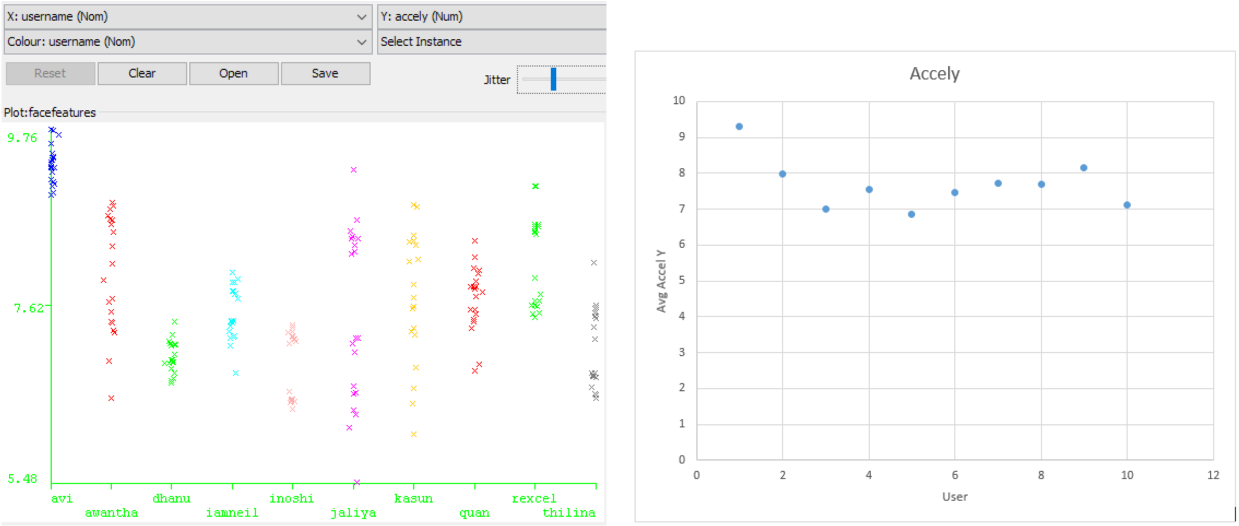


X: username (Nom)

Y: accelz (Num)

Colour: username (Nom)

Figure 5.11: Acceleration Force along Z Axis



X: username (Nom)

Y: accely (Num)

Colour: username (Nom)

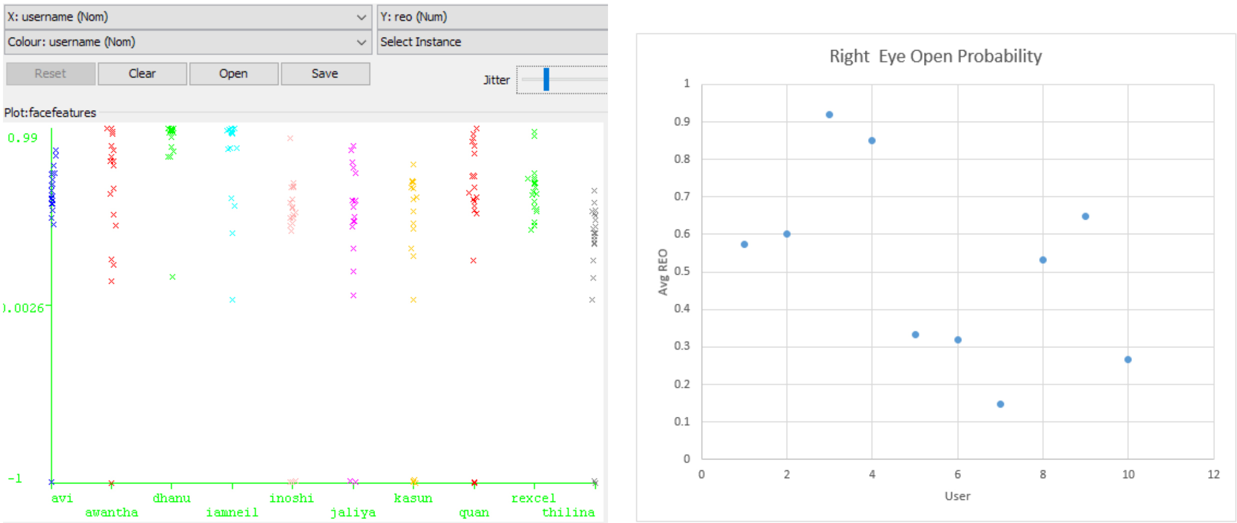
Figure 5.12: Acceleration Force along Y Axis

and T5 threshold value which is 70% (2.5879) while verifying the user's face.

A similar approach is followed for the touch dynamic verification process as well. Same confidence intervals listed in Table 5.1 will be used. For touch dynamic based user verification, seven features were selected and their weights are listed in Table 5.7. Refer B for more details.

Thus if all the touch features get matched, highest score that can be achieved is 4.572. Therefore threshold values for touch data verification are calculated using following equation 5.2. Here X denotes the percentage value and list of threshold values are contained in Table 5.8.

$$T = 4.572 * \frac{X}{100} \tag{5.2}$$



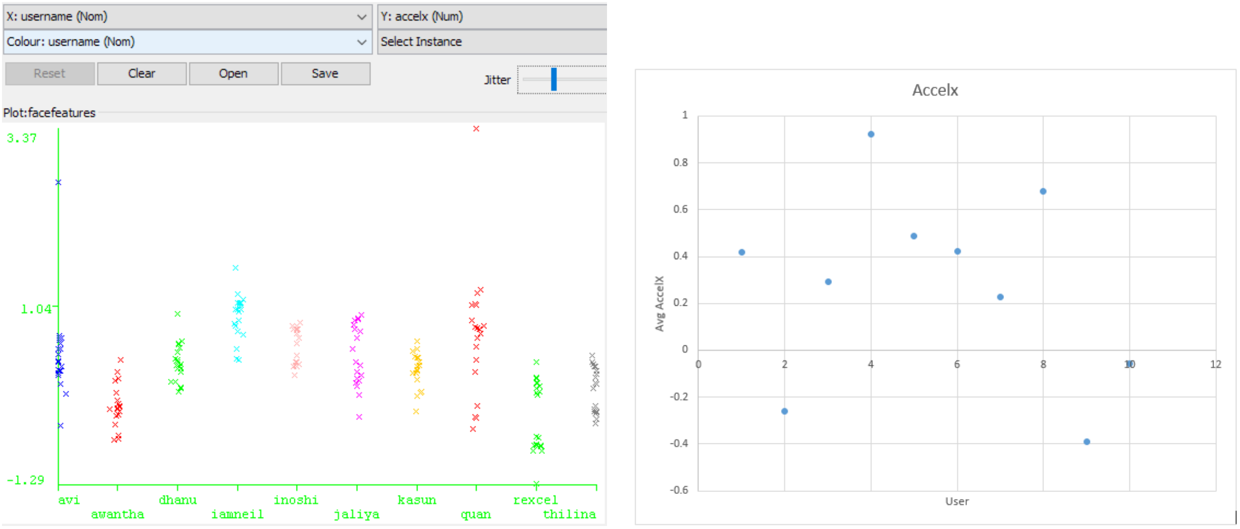
X: username (Nom)

Colour: username (Nom)

Y: reo (Num)

Plot:facefeatures

Figure 5.13: Right Eye Open Probability



X: username (Nom)

Colour: username (Nom)

Y: accelx (Num)

Plot:facefeatures

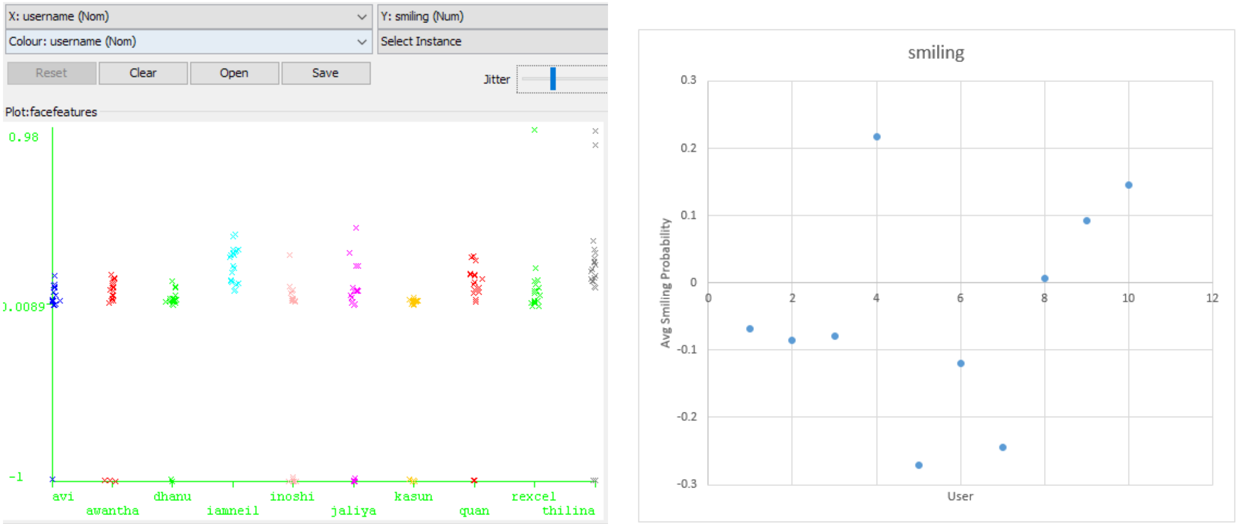
Figure 5.14: Acceleration Force along X Axis

Then success rate for each legitimate user have been calculated by using 14 training samples and the result is listed in Table 5.9.

Then FAR and FRR is calculated to analyze the behavior of touch dynamics for different users. Calculation was done in the same that explained under face verification.Results of FAR and FRR are listed in Table 5.10 and Table 5.11 respectively.

Furthermore, Figure 5.30 illustrate the graph that plots false acceptance rate against different threshold values for different confidence intervals. the graph shows that the FAR is lower when comparing to the value obtained for face verification.

The Figure 5.31 contains the graph that describe the false rejection rate against different threshold values like 30%, 40%, 50%, 60%, 70%, 80% and 90% for different confidence inter-

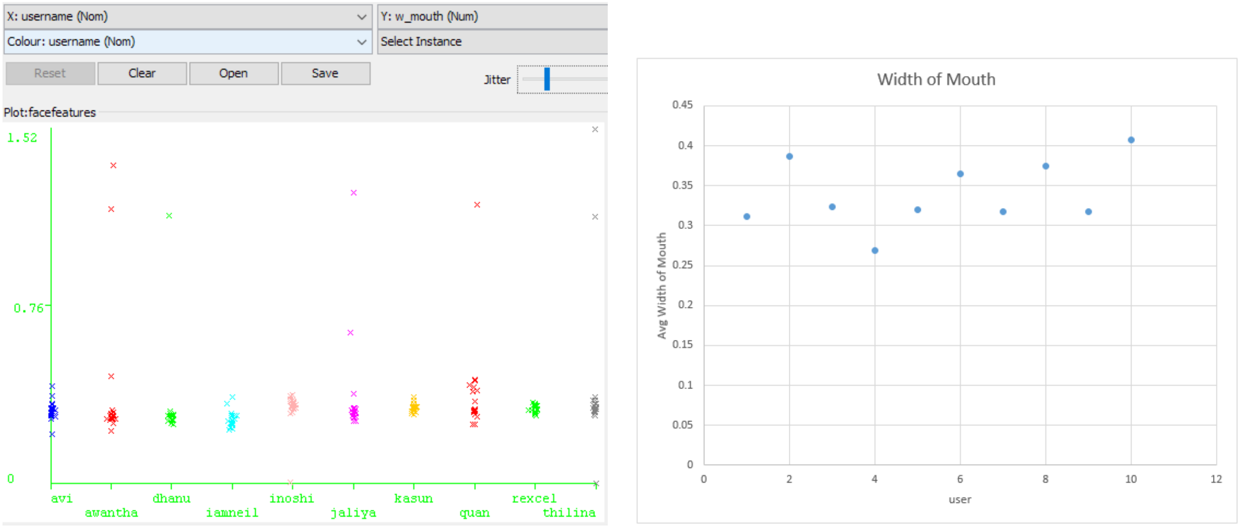


X: username (Nom)

Y: smiling (Num)

Colour: username (Nom)

Figure 5.15: Smiling Probability



X: username (Nom)

Y: w_mouth (Num)

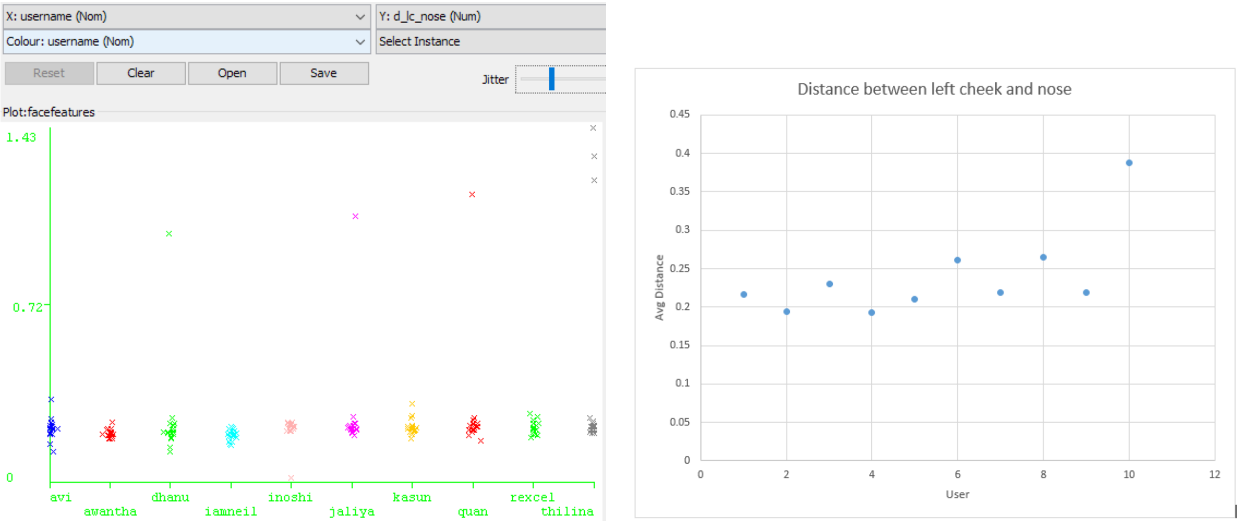
Colour: username (Nom)

Figure 5.16: Width of Mouth

vals. When the confidence interval is higher the rejection rate is getting lower and this value eventually gets increased when increasing the threshold limit.

Basically, these information are not enough as explained earlier. Hence EER graph is drawn by using FAR and FRR values as shown in Figure 5.32 to determine the suitable confidence interval and the threshold value for touch data verification.

As shown in the EER graph plot against different threshold values, FAR and FRR lines for all the confidence interval values except 99.9 confidence interval do not cross each other. Thus 99.9 is selected as the confidence interval for touch data verification and a corresponding threshold value which is 46% (2.10312) selected as the threshold value.

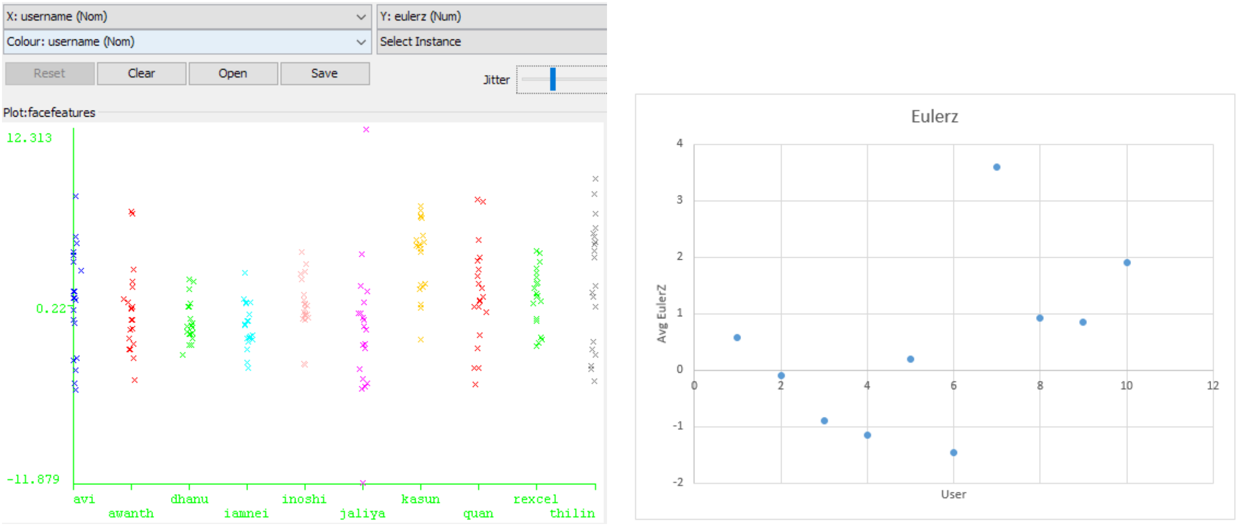


X: username (Nom)

Y: d_lc_nose (Num)

Colour: username (Nom)

Figure 5.17: Distance between Left Cheek and Nose



X: username (Nom)

Y: eulerz (Num)

Colour: username (Nom)

Figure 5.18: Euler Z

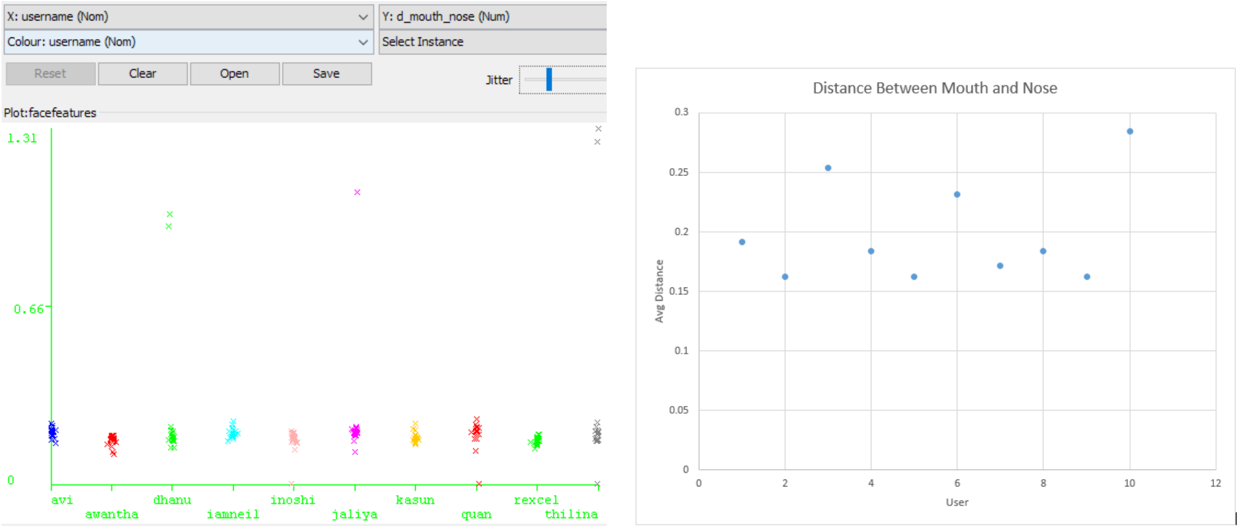
Finally, the decision is made by considering the success rate of both face and touch feature verification result.

5.5.1 Decision Making

The final decision is made by considering the acceptance rate of both face and touch features.For decision making, we will use two approaches as listed below.

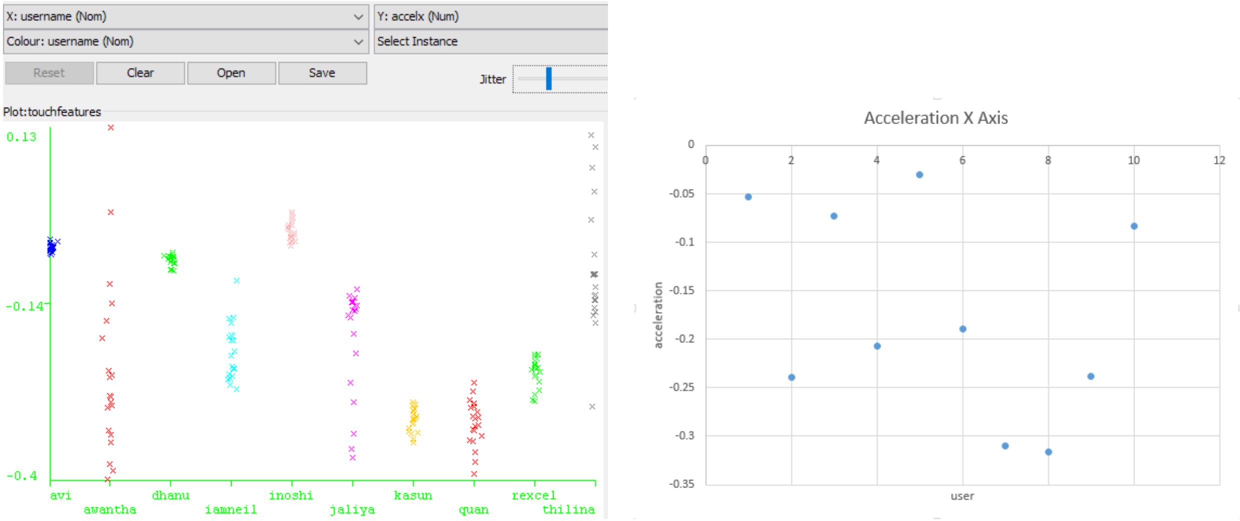
1. Either of the face or touch dynamics features need to be successfully verified.
2. Both face and touch dynamics features need to be successfully verified.

The analysis of result for the first approach is as follows. For face feature verification, 99.9 confidence level and 0.7 threshold value are used and touch data verified by using 99.9 confi-



X: username (Nom)
Y: d_mouth_nose (Num)
Colour: username (Nom)

Figure 5.19: Distance between Mouth & Nose



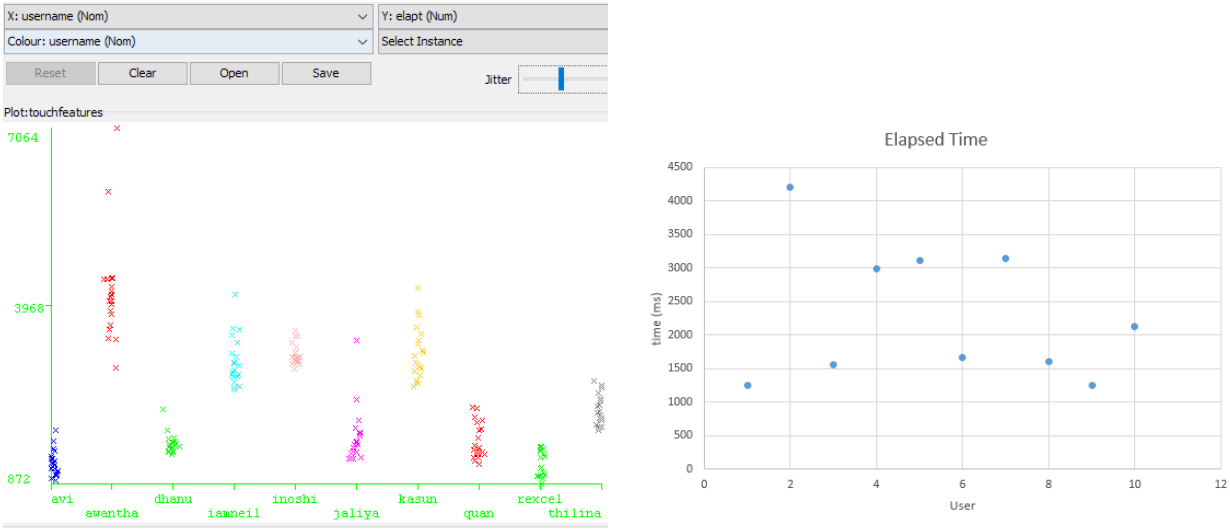
X: username (Nom)
Y: accelx (Num)
Colour: username (Nom)

Figure 5.20: Acceleration along X axis

dence level 0.46 threshold values as calculated above. The consolidated result of this analysis is listed in the Table 5.12 and success rate for each user have been taken into a graph in Figure 5.33. More details about the statistics are included in Appendix C.

In the first approach, it shows higher success rate which is around 92.86%. Therefore false rejection rate is 7.14%. If analyze the false acceptance rate (FAR) for the first approach. Table 5.13 shows the consolidated results obtained for each user.

Moreover, graph in Figure 5.34 help provides an overview of the false acceptance rates for the first approach. This overall FAR value is around 38.89% which is a comparatively a higher number.



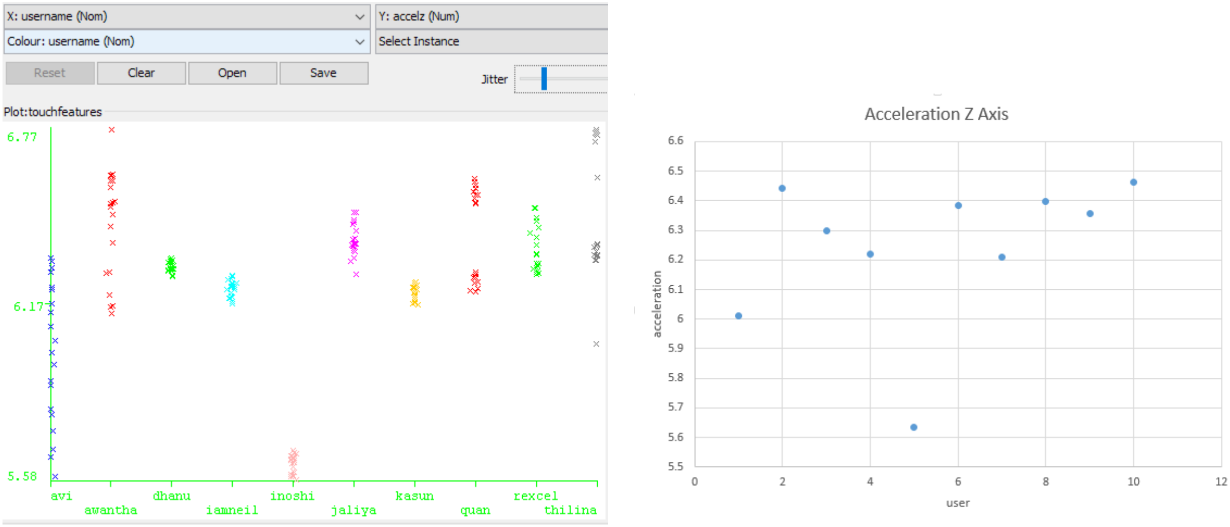
X: username (Nom)

Colour: username (Nom)

Y: elapt (Num)

Select Instance

Figure 5.21: Total elapsed time



X: username (Nom)

Colour: username (Nom)

Y: accelz (Num)

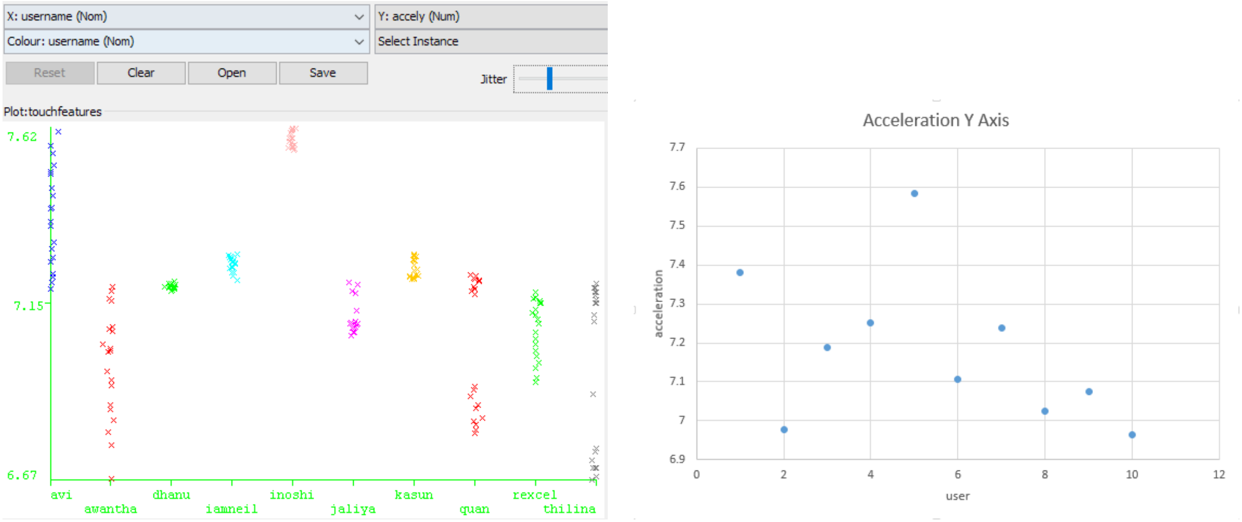
Select Instance

Figure 5.22: Acceleration along Z axis

Then we will focus on the second approach. In the second approach, we consider the success of both face and touch data verification results. Table 5.14 contains the list of success rates per each user.

Furthermore, Figure 5.35 depicts the success rates for each user plotted by using the training dataset. According to these results, it is clear that the second approach has a very less success rate when comparing to the first approach. This value is taken 57.14% in average.

As same as the success rate, false acceptance rate also very important to selecting a better approach. Following Table 5.13 and Figure 5.34 shows the consolidated false acceptance results obtained for the second approach.



X: username (Nom)

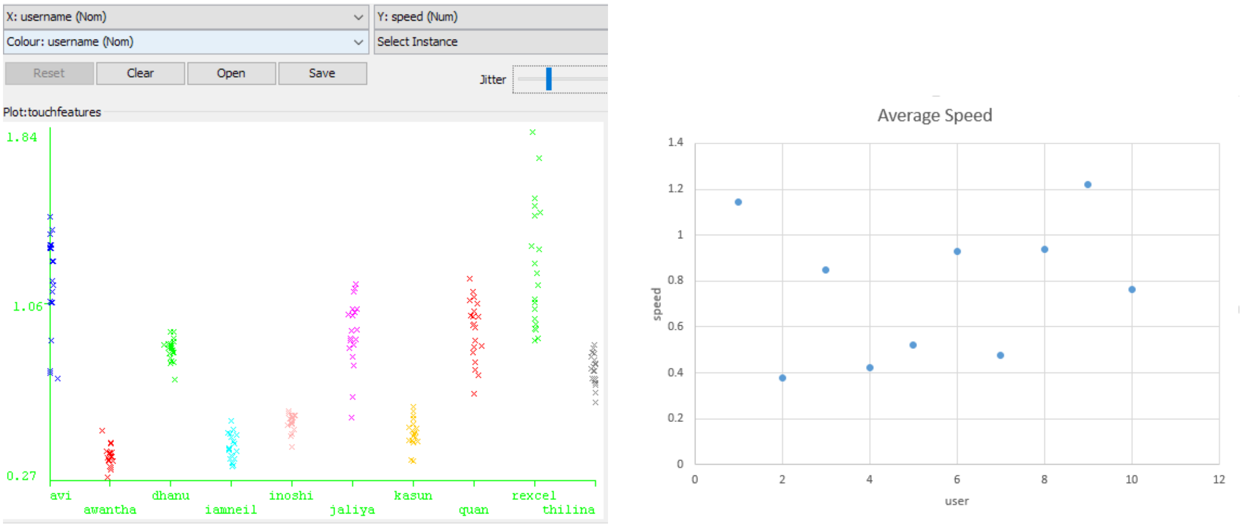
Y: accely (Num)

Colour: username (Nom)

Select Instance

Plot:touchfeatures

Figure 5.23: Acceleration along Y axis



X: username (Nom)

Y: speed (Num)

Colour: username (Nom)

Select Instance

Plot:touchfeatures

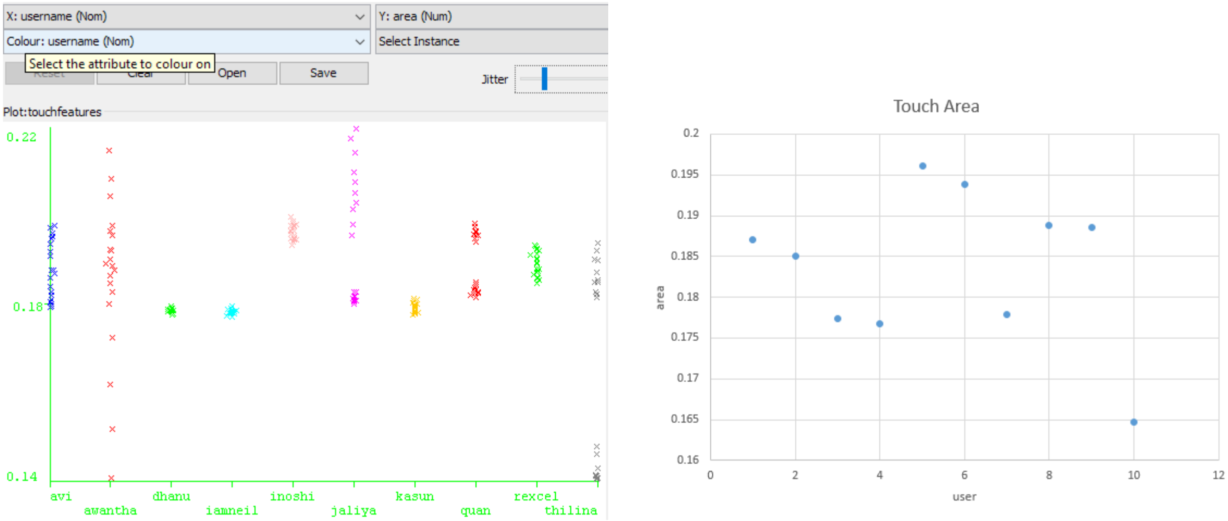
Figure 5.24: Finger Moving Speed

According to the statistics, the second approach is more suitable to gain more security since it's overall false acceptance rate is around 11.98% which is a less value. But the success rate is highly important for the usability. Thus the first approach can be selected as the more suitable method for the final decision making. Next section will focus on the evaluation of the test result which obtains by using test data set.

5.6 Evaluation

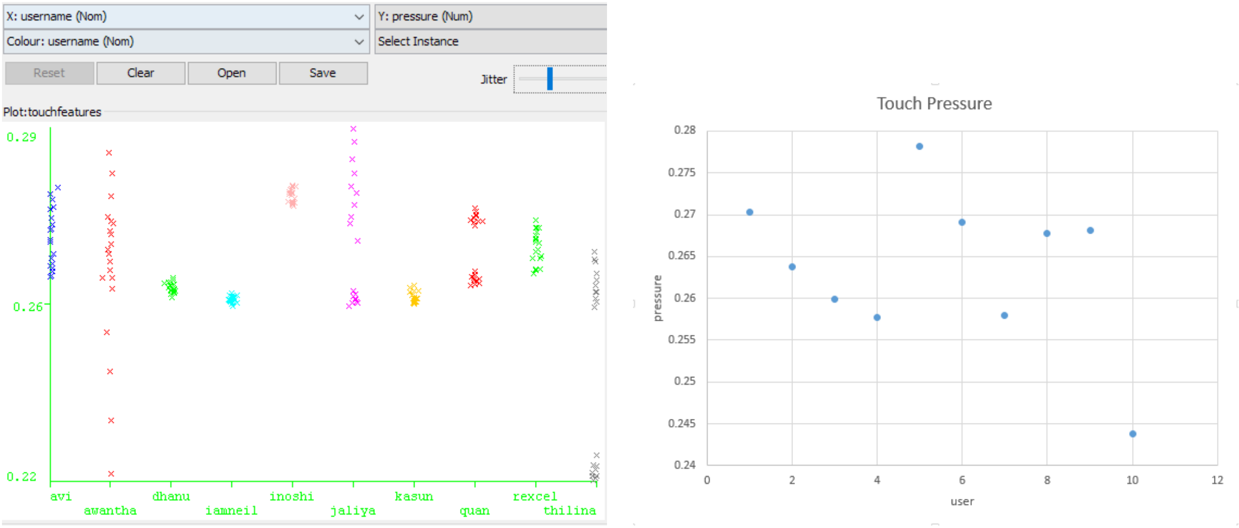
In order to evaluate the system, quantitative evaluation approach is used. The metrics that are commonly used to evaluate the verification accuracy of a given authentication method are false rejection rate (FRR), false acceptance rate (FAR) and equal error rate (EER).

- False Rejection Rate



X: username (Nom)
Y: area (Num)
Colour: username (Nom)
Select Instance

Figure 5.25: Touch Area



X: username (Nom)
Y: pressure (Num)
Colour: username (Nom)
Select Instance

Figure 5.26: Touch Pressure

- False Acceptance Rate
- Equal Error Rate

False Rejection Rate (FRR) False rejection rate is the percentage of the number of legitimate users, who falsely rejected against the total number of legitimate user tries. Having lower value as FRR means that fewer number of legitimate users being falsely rejected which implies a high usability level.

False Acceptance Rate (FAR) False acceptance rate is the percentage of the number of illegitimate users who are falsely accepted against the total number of illegitimate user tries. Accepted behavior is to have low FAR value. Having high FAR value indicates that system has a lower security level.

Id	Percentage %	Value
T1	30	1.1091
T2	40	1.4788
T3	50	1.8485
T4	60	2.2182
T5	70	2.5879
T6	80	2.9576
T7	90	3.3273

Table 5.3: Threshold Values for Face Verification

Confidence Interval (%)	T1	T2	T3	T4	T5	T6	T7
80	40.71%	26.43%	15.00%	6.43%	1.43%	0.00%	0.00%
90	71.43%	44.29%	27.86%	14.29%	6.43%	2.14%	0.71%
95	85.71%	68.57%	50.00%	27.86%	14.29%	7.14%	1.43%
98	92.86%	85.71%	70.00%	42.14%	27.86%	14.29%	3.57%
99	95.00%	89.29%	79.29%	63.57%	47.14%	27.14%	7.86%
99.9	99.29%	95.71%	89.29%	84.29%	77.86%	59.29%	35.71%

Table 5.4: Success Rate of Recognising Face

Equal Error Rate (ERR) This is a single number performance metric and EER is commonly used to measure and compare the overall accuracy level of different biometrics authentication methods. EER is obtained by finding the interception point of two graphs, one for FRR and the other for FAR. So having lower ERR means having lower FRR and FAR which indicates a better accuracy performance of a biometric based authentication method

5.6.1 Face Verification Result Evaluation

For the evaluation, test dataset which consists of 60 records is used. First, both face biometrics and touch behavior biometrics are evaluated separately to check the success rate, false rejection rate and false acceptance rate. Here 99.9 confidence level is used as the confidence level and threshold values for face and touch data verifications are 0.7 and 0.46 respectively. These values were determined based on the analysis of training dataset as described under clas-

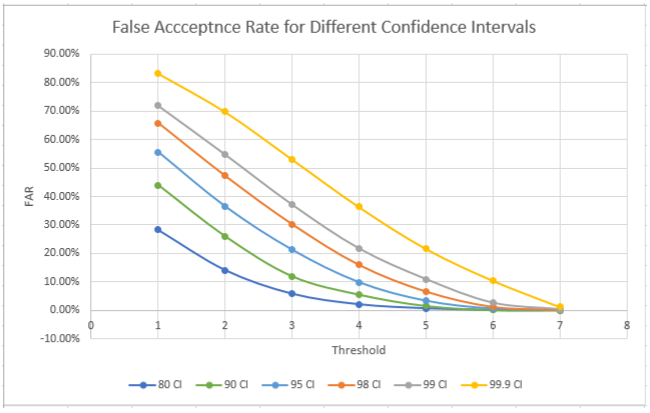


Figure 5.27: FAR for Different Confidence Intervals

Confidence Interval (%)	T1	T2	T3	T4	T5	T6	T7
80	28.33%	14.13%	5.79%	2.06%	0.63%	0.08%	0.00%
90	43.89%	25.95%	11.98%	5.56%	1.59%	0.08%	0.00%
95	55.63%	36.51%	21.27%	9.84%	3.33%	0.56%	0.08%
98	65.79%	47.22%	30.24%	15.95%	6.59%	1.19%	0.08%
99	71.98%	54.76%	37.14%	21.67%	10.87%	2.54%	0.24%
99.9	83.25%	69.76%	53.10%	36.35%	21.67%	10.48%	1.27%

Table 5.5: False Acceptance Rate for Different Confidence Intervals

Confidence Interval (%)	T1	T2	T3	T4	T5	T6	T7
80	59.29%	73.57%	85.00%	93.57%	98.57%	100.00%	100.00%
90	28.57%	55.71%	72.14%	85.71%	93.57%	97.86%	99.29%
95	14.29%	31.43%	50.00%	72.14%	85.71%	92.86%	98.57%
98	7.14%	14.29%	30.00%	57.86%	72.14%	85.71%	96.43%
99	5.00%	10.71%	20.71%	36.43%	52.86%	72.86%	92.14%
99.9	0.71%	4.29%	10.71%	15.71%	22.14%	40.71%	64.29%

Table 5.6: False Rejection Rate for Different Confidence Intervals

sification & Verification section in this chapter.

This subsection focuses on the face biometric verification and Table 5.16 contains the evaluation results for face biometric verification.

The graphical view of above result can be seen in the Figure 5.37. This graph shows that for some users FRR take a higher value and for some other users FAR values are higher. However, the overall success rate for face recognition is 80% which is comparatively a higher value and FRR rate is 20% and the FAR rate is 14.17%.

5.6.2 Touch Verification Result Evaluation

This section will discuss the evaluation of the touch verification results. Here success rate, false rejection rate and false acceptance rates are calculated by using the touch test dataset. The result for each user are listed in Table 5.17 and graphically illustrate in Figure 5.38.

According to the statistics, proposed method can achieve 66.67% success rate which can be considered as an average success rate. Also if consider the false acceptance rate, it is 14.14 % which is a less value.

However, the final objective of this project is to combine these results to obtain better accuracy rate. Next subsection is allocated to evaluate the final result set.

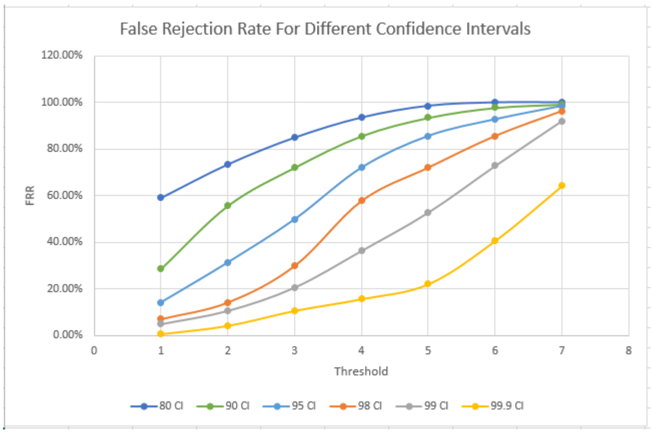


Figure 5.28: FRR for Different Confidence Intervals

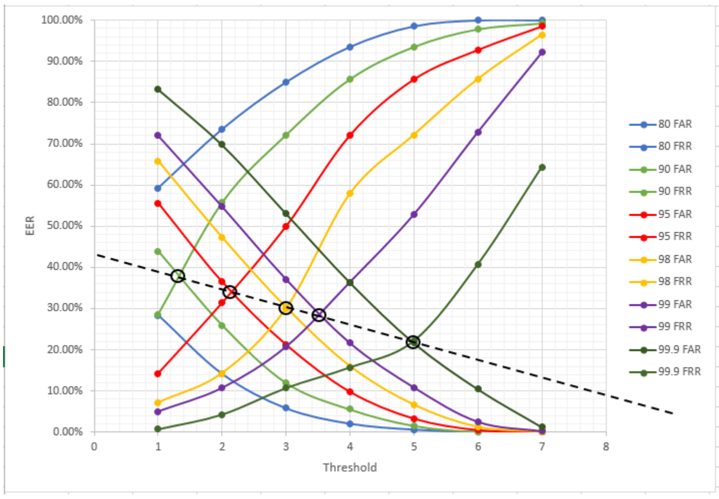


Figure 5.29: EER for Different Confidence Intervals

5.6.3 Final Result Evaluation

The final result is obtained based on both face and touch verification results and decision is made by using the first approach described in previous sections. Here, if one of the verification results is true, then it considered as a successful attempt. Table 5.18 contains the results obtained for each user.

The consolidated value of results achieved by this method is listed in Table 5.19.

When analyzing the result, proposed authentication method provide higher success rate which affects the usability of the mobile application. But it also exhibits higher false acceptance rate which affects the security level of the application. Therefore more tuning should be done to reduce the amount of FAR to achieve higher security level.

5.6.4 Device Performance

In addition to the result statistics, resource utilization of proposed application also very important factor. Memory usage and CPU usage of the android application has been captured in

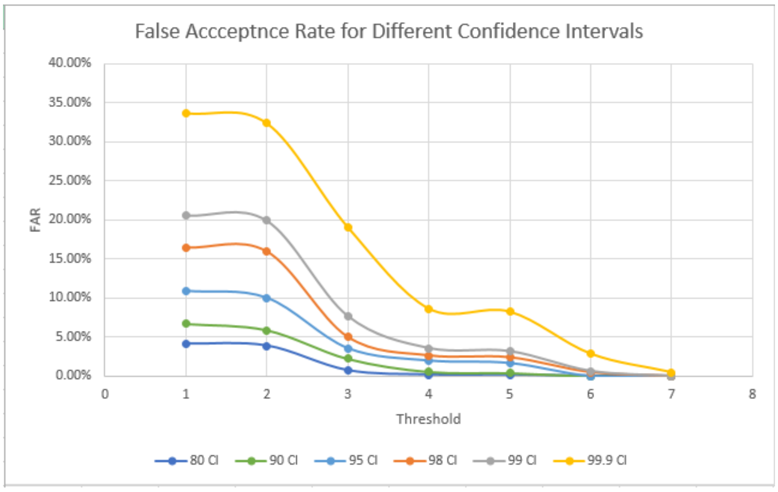


Figure 5.30: FAR of Touch Verification for Different Confidence Intervals

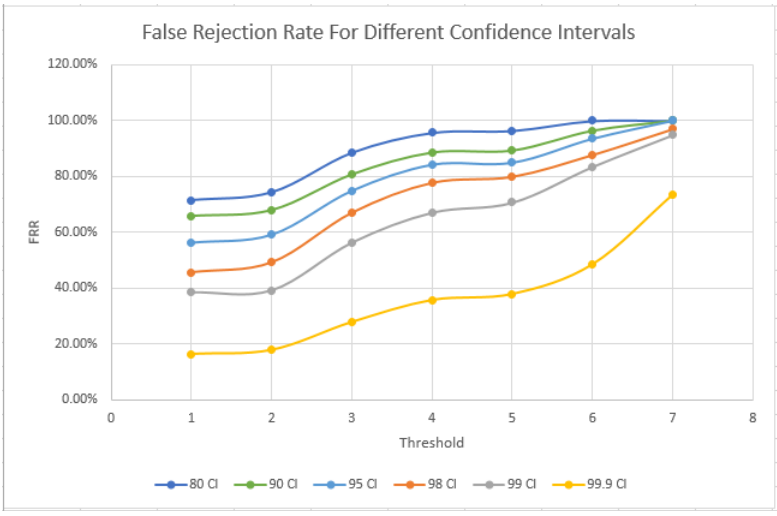


Figure 5.31: FRR of Touch Verification for Different Confidence Intervals

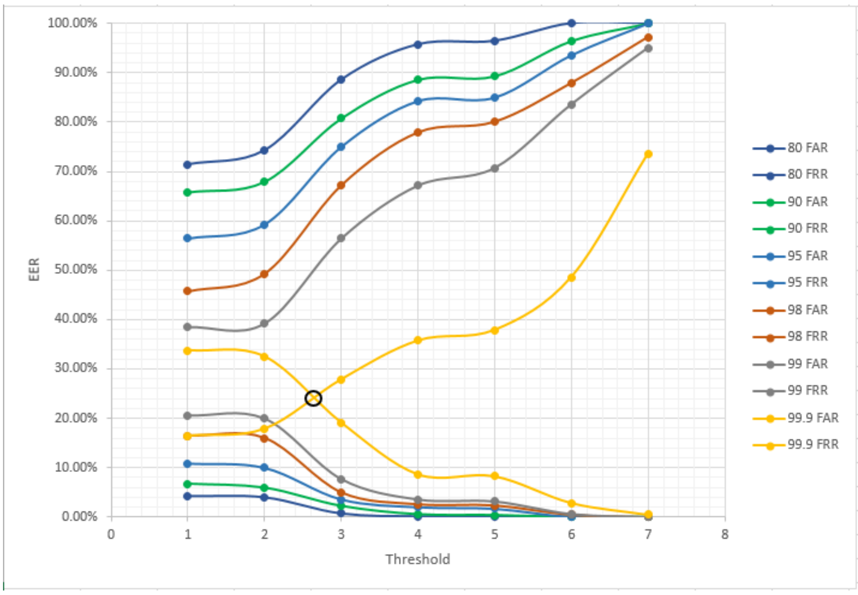


Figure 5.32: EER of Touch Verification for Different Confidence Intervals

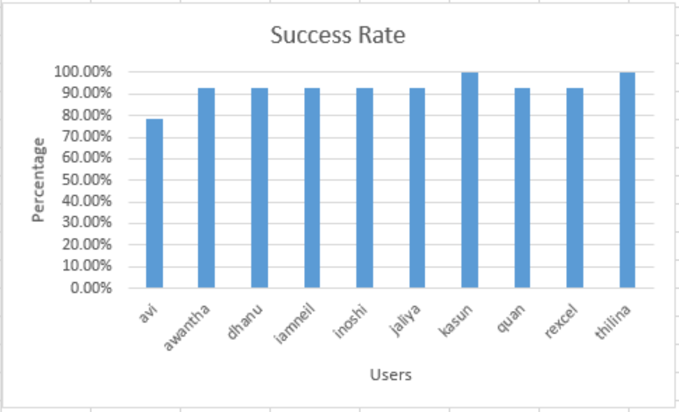


Figure 5.33: User Verification Result Graph for Training Data

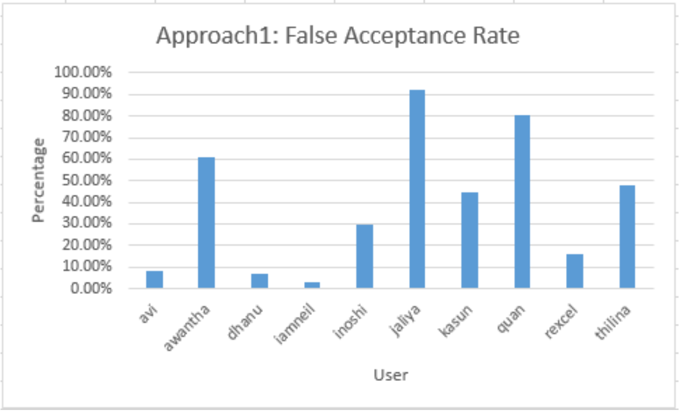


Figure 5.34: False User Acceptance Results for Training Data

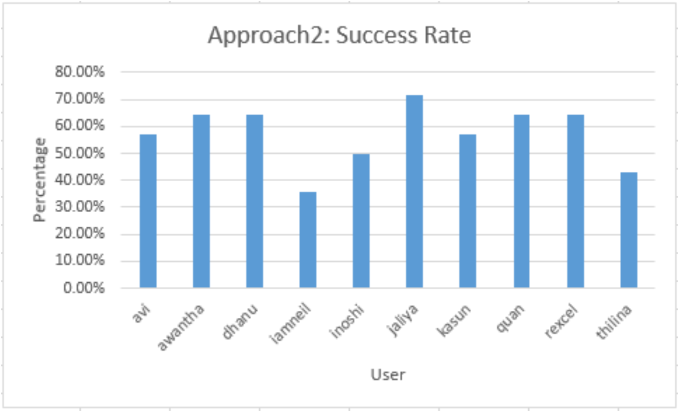


Figure 5.35: User Verification Result Graph for Training Data (Approach2)

Feature	Weight
AccelX	0.731
Elapsed Time	0.669
AccelZ	0.659
AccelY	0.658
Speed	0.638
Touch Area	0.61
Touch Pressure	0.607
Total	**4.572**

Table 5.7: Face Feature Weights

Id	Percentage %	Value
T1	30	1.3716
T2	40	1.8288
T3	50	2.286
T4	60	2.7432
T5	70	3.2004
T6	80	3.6576
T7	90	4.1148

Table 5.8: Threshold Values for Face Verification

Figure 5.39 and Figure 5.40 respectively.

5.7 Summary

The purpose of this chapter is to analyze and evaluate the result of the proposed authentication method. The proposed authentication method is using a statistical approach for user classification and verification and a final decision is determined based on a threshold value. In the statistical method, mean values, standard deviation, lower and higher limits for different two tail confidence levels are taken. This information is obtained for both face features and touch features for different confidence levels which are 80%, 90%, 95%, 98%, 99% and 99.9%. Then the best confidence level was selected by analyzing the results obtained for the training dataset.

According to the analysis, 99.9 confidence level was selected as the most appropriate confidence level for both face and touch verification. In the same way, suitable threshold values also determined by using the training dataset. Here EER is used as the criteria to select best confidence level and threshold values. Based on the analysis 0.7 (2.5879) has been selected as the threshold value for the face verification and 0.46 (2.10312) has been selected as the threshold for touch data verification.

Finally, the overall success rate, false rejection rate and false acceptance rates were calculated using the test dataset and final result is listed in Table 5.19.

Confidence Interval (%)	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**
80	28.57%	25.71%	11.43%	4.29%	3.57%	0.00%	0.00%
90	34.29%	32.14%	19.29%	11.43%	10.71%	3.57%	0.00%
95	43.57%	40.71%	25.00%	15.71%	15.00%	6.43%	0.00%
98	54.29%	50.71%	32.86%	22.14%	20.00%	12.14%	2.86%
99	61.43%	60.71%	43.57%	32.86%	29.29%	16.43%	5.00%
99.9	83.57%	82.14%	72.14%	64.29%	62.14%	51.43%	26.43%

Table 5.9: Success Rate of Touch Verification for Different Confidence Intervals

Confidence Interval (%)	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**
80	4.13%	3.89%	0.71%	0.16%	0.16%	0.00%	0.00%
90	6.67%	5.87%	2.22%	0.56%	0.40%	0.00%	0.00%
95	10.87%	10.00%	3.57%	1.98%	1.67%	0.00%	0.00%
98	16.43%	15.95%	5.00%	2.62%	2.38%	0.48%	0.00%
99	20.56%	19.92%	7.70%	3.57%	3.17%	0.63%	0.00%
99.9	33.73%	32.46%	19.13%	8.65%	8.25%	2.86%	0.48%

Table 5.10: FAR of Touch Verification for Different Confidence Intervals

Moreover, the resource utilization of the device also captured here to see whether this method is compatible with mobile devices. Next chapter will conclude the result of the proposed with comparing to other research works in the field.

Confidence Interval (%)	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**
80	71.43%	74.29%	88.57%	95.71%	96.43%	100.00%	100.00%
90	65.71%	67.86%	80.71%	88.57%	89.29%	96.43%	100.00%
95	56.43%	59.29%	75.00%	84.29%	85.00%	93.57%	100.00%
98	45.71%	49.29%	67.14%	77.86%	80.00%	87.86%	97.14%
99	38.57%	39.29%	56.43%	67.14%	70.71%	83.57%	95.00%
99.9	16.43%	17.86%	27.86%	35.71%	37.86%	48.57%	73.57%

Table 5.11: FRR of Touch Verification for Different Confidence Intervals

User	**Success Rate (%)**
avi	78.57%
awantha	92.86%
dhanu	92.86%
iamneil	92.86%
inoshi	92.86%
jaliya	92.86%
kasun	100.00%
quan	92.86%
rexcel	92.86%
thilina	100.00%

Table 5.12: User Verification Results for Training Data

User	**FAR (%)**
avi	7.94%
awantha	61.11%
dhanu	7.14%
iamneil	3.17%
inoshi	29.37%
jaliya	92.06%
kasun	44.44%
quan	80.16%
rexcel	15.87%
thilina	47.62%

Table 5.13: False User Acceptance Results for Training Data

User	**Success Rate (%)**
avi	57.14%
awantha	64.29%
dhanu	64.29%
iamneil	35.71%
inoshi	50.00%
jaliya	71.43%
kasun	57.14%
quan	64.29%
rexcel	64.29%
thilina	42.86%

Table 5.14: User Verification Results for Training Data (Approach2)

User	FAR (%)
avi	0.00%
awantha	27.78%
dhanu	0.00%
iamneil	0.00%
inoshi	0.00%
jaliya	30.16%
kasun	16.67%
quan	30.95%
rexcel	0.00%
thilina	14.29%

Table 5.15: False User Acceptance Results for Training Data (Approach2)

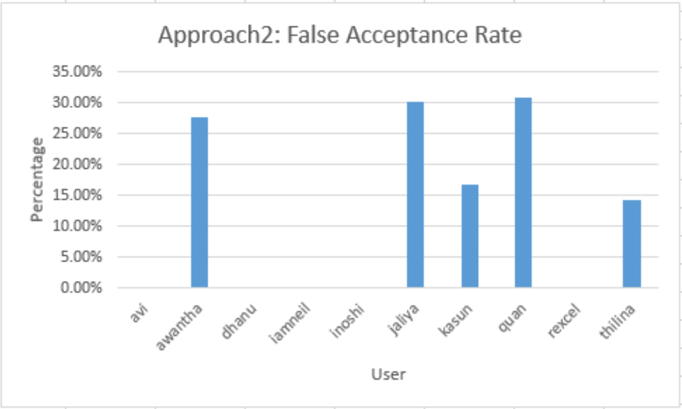


Figure 5.36: False User Acceptance Results for Training Data (Approach2)

User	Success Rate (%)	FRR (%)	FAR(%)
avi	66.67%	33.33%	3.70%
awantha	66.67%	33.33%	44.44%
dhanu	83.33%	16.67%	8.93%
iamneil	66.67%	33.33%	1.85%
inoshi	100.00%	0.00%	7.78%
jaliya	100.00%	0.00%	27.22%
kasun	33.33%	66.67%	6.11%
quan	100.00%	0.00%	22.22%
rexcel	100.00%	0.00%	3.89%
thilina	83.33%	16.67%	15.56%

Table 5.16: Results Face Biometrics

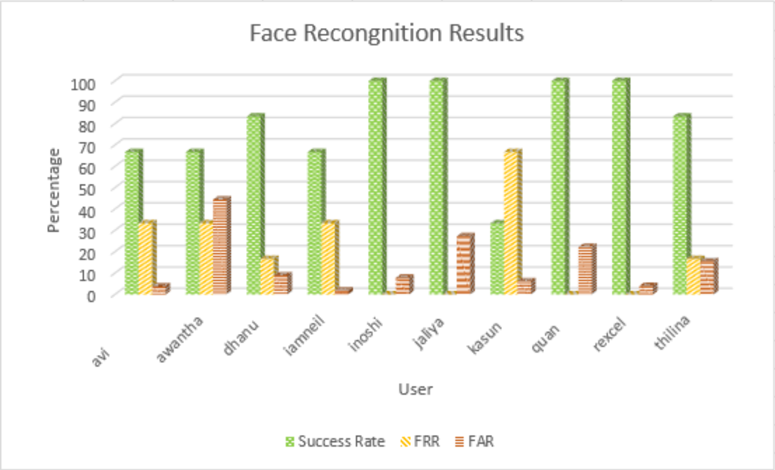


Figure 5.37: Face Verification Results

User	Success Rate (%)	FRR (%)	FAR(%)
avi	66.67%	33.33%	11.11%
awantha	66.67%	33.33%	38.89%
dhanu	66.67%	33.33%	0.00%
iamneil	66.67%	33.33%	0.00%
inoshi	66.67%	33.33%	0.00%
jaliya	83.33%	16.67%	16.67%
kasun	66.67%	33.33%	24.07%
quan	66.67%	33.33%	20.37%
rexcel	66.67%	33.33%	0.00%
thilina	50.00%	50.00%	33.33%

Table 5.17: Results Touch Behavior Biometrics

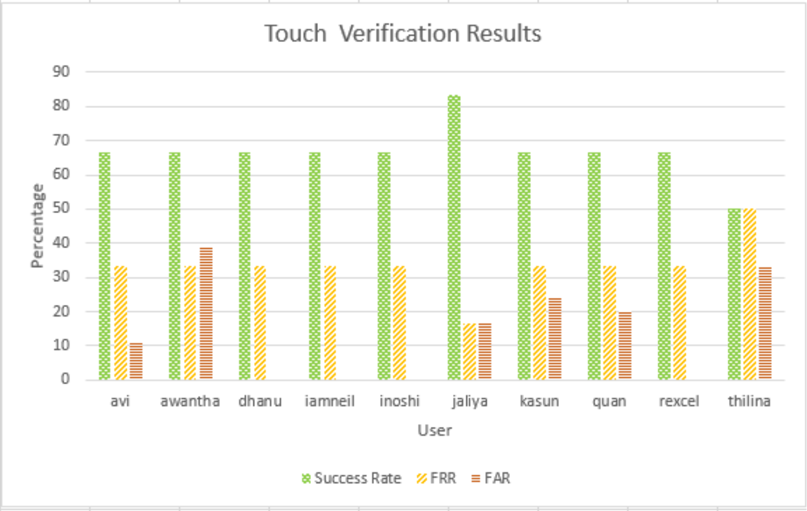


Figure 5.38: Touch Data Verification Results

User	**Success Rate (%)**	**FRR (%)**	**FAR(%)**
avi	83.33	16.67	14.81
awantha	83.33	16.67	57.41
dhanu	83.33	16.67	9.26
iamneil	100	0	1.85
inoshi	100	0	25.93
jaliya	100	0	92.59
kasun	66.67	33.33	35.19
quan	100	0	75.93
rexcel	100	0	12.96
thilina	100	0	68.52

Table 5.18: Final Result

Success Rate (%)	**FRR (%)**	**FAR(%)**
91.67	8.33	39.44

Table 5.19: Final Result

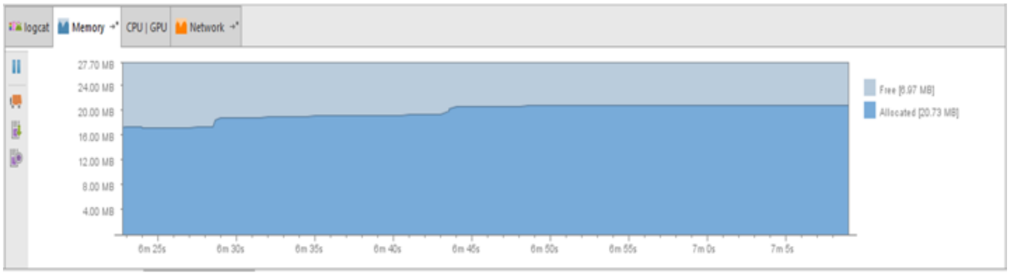


Figure 5.39: Memory Usage

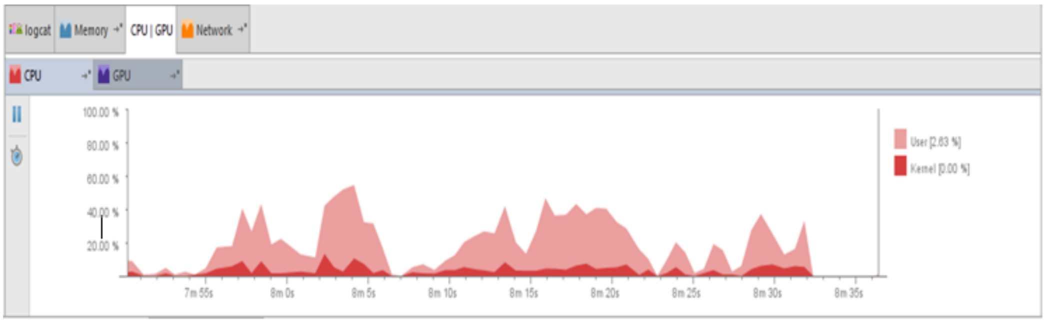


Figure 5.40: CPU Usage

Chapter 6

Conclusion

The gold of this project is to introduce an implicit approach to improve mobile application authentication where it authenticates users instantly without connecting to any back-end server. The basic idea of this project is to empower user authentication facility by using existing resources of common smartphone devices. Therefore the combination of face features and touch behavior while user entering the password pattern was evaluated in this project. The result and finding of this project will be discussed here.

As shown Table 6.1, proposed statistical method was able to achieve a higher success rate and lower false rejection rate when using both face and touch behavior biometrics. But if use only touch dynamics for user authentication, the success rate is taking less value than a previous value such as 66.67. This result is depicted in Table 6.2.

Success Rate (%)	**FRR (%)**	**FAR(%)**
91.67	8.33	39.44

Table 6.1: Final Result When Using Both Face and Touch Behavior Biometrics

Success Rate (%)	**FRR (%)**	**FAR(%)**
66.67	33.33	14.44

Table 6.2: Final Result When Using Touch Behavior Biometrics

The accuracy rate for these approaches are calculated by using following equation **??**;

$$Accuracy = \frac{\sum TrueNegative + \sum TruePostive}{\sum TrueNegative + \sum TruePostive + \sum FalseNegative + \sum FalsePostive} \tag{6.1}$$

True Negative: Correctly rejected True Positive: Correctly Accepted False Negative: Falsely Rejected False Positive: Falsely Accepted

According to above equation, the accuracy rate for both approaches is 76.11%.

Pattern	FAR	FRR	Accuracy%
1	0.18	0.135	85.5
2	0.18	0.08	90.9
3	0.136	0.07	92.3

Table 6.3: Measurements of UA-UPTD Performance [46]

Before concluding the results, result comparison is done with the experiment done by Waheeh et al (2015) which is shown in Table 6.3 and the experiment result of the research work done by Luca et al (2012) (listed in Table 6.2). Both of these experiments done by using only touch dynamics for user authentication.

Therefore comparatively proposed statical method show less performance than these two experiments.

Accuracy (%)	FRR (%)	FAR(%)
77	19	21

Table 6.4: Measurements of DTW Performance [4]

Moreover following are the other findings of this research.

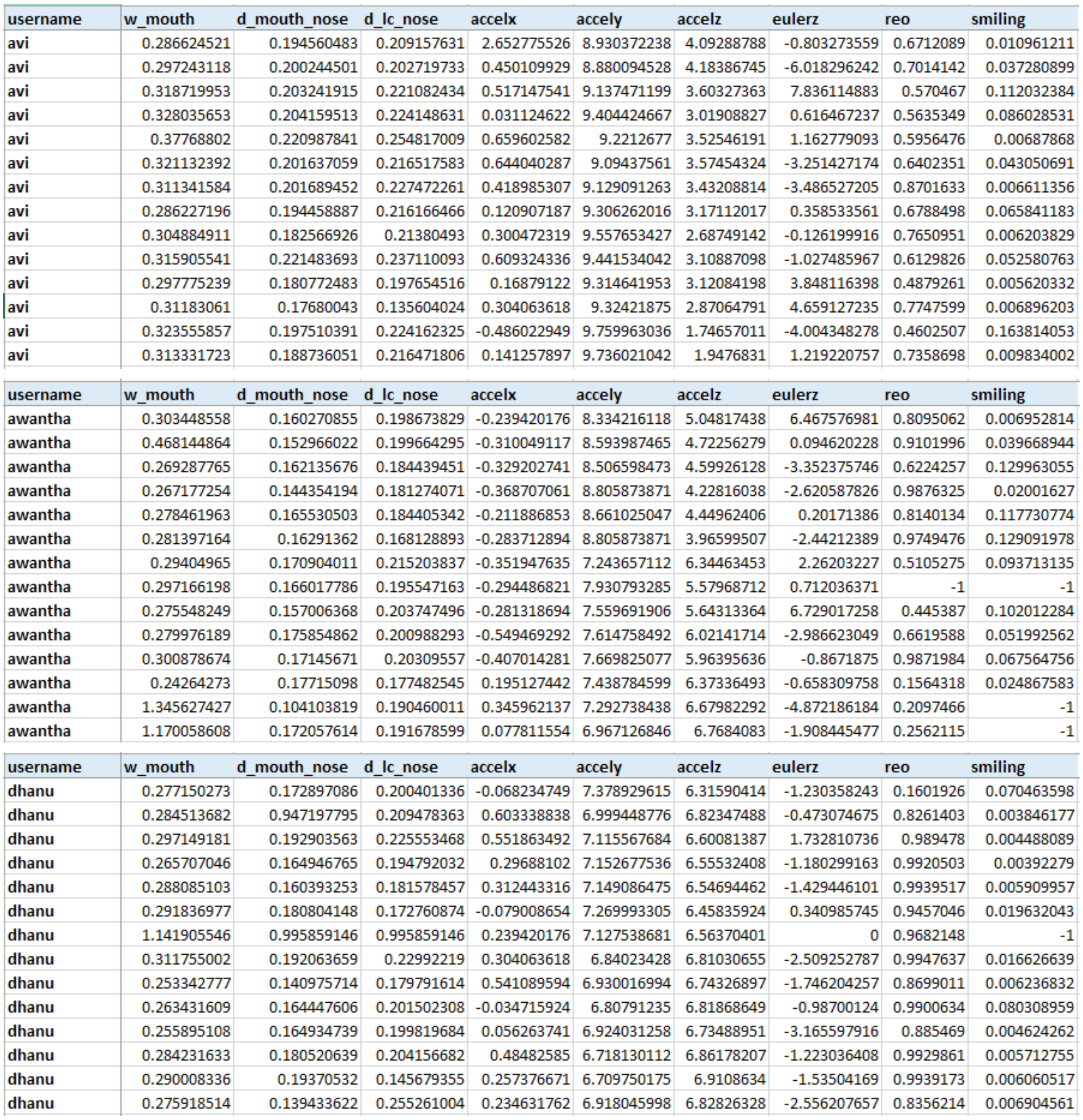
- Using user verification instead user identification method is more suitable for user authentication on the mobile application due to it needs to verify user against data of a particular user that claimed to be. Thus fewer data will be processed and resource utilization will be less.

- In this research only selected features have been used to perform user classification and verification. Thus, removing less effective features from feature will cause to reduce the dimension and will avoid the over-fitting problem.

Still there are few open points remaining for future work. First, a long-term study is required to measure the performance of measured approach by using different patterns with different lengths. The second point is, it is required to improve the accuracy of the method by decreasing the false acceptance rate. Because still there is a room for improve face recognition part.

Appendix A

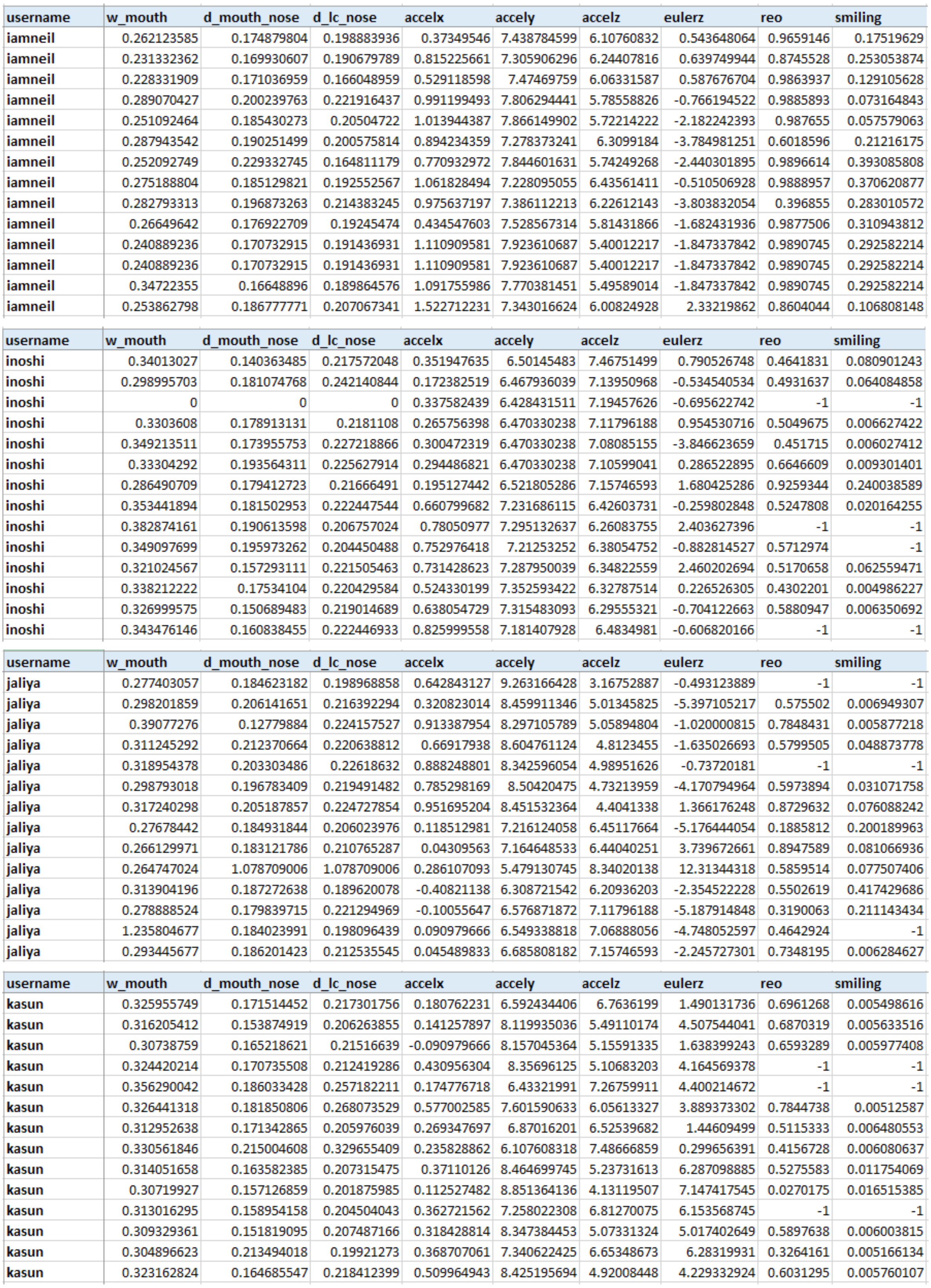
Face Training Dataset



username	w_mouth	d_mouth_nose	d_lc_nose	accelx	accely	accelz	eulerz	reo	smiling
avi	0.286624521	0.194560483	0.209157631	2.652775526	8.930372238	4.09288788	-0.803273559	0.6712089	0.010961211
avi	0.297243118	0.200244501	0.202719733	0.450109929	8.880094528	4.18386745	-6.018296242	0.7014142	0.037280899
avi	0.318719953	0.203241915	0.221082434	0.517147541	9.137471199	3.60327363	7.836114883	0.570467	0.112032384
avi	0.328035653	0.204159513	0.224148631	0.031124622	9.404424667	3.01908827	0.616467237	0.5635349	0.086028531
avi	0.37768802	0.220987841	0.254817009	0.659602582	9.2212677	3.52546191	1.162779093	0.5956476	0.00687868
avi	0.321132392	0.201637059	0.216517583	0.644040287	9.09437561	3.57454324	-3.251427174	0.6402351	0.043050691
avi	0.311341584	0.201689452	0.227472261	0.418985307	9.129091263	3.43208814	-3.486527205	0.8701633	0.006611356
avi	0.286227196	0.194458887	0.216166466	0.120907187	9.306262016	3.17112017	0.358533561	0.6788498	0.065841183
avi	0.304884911	0.182566926	0.21380493	0.300472319	9.557653427	2.68749142	-0.126199916	0.7650951	0.006203829
avi	0.315905541	0.221483693	0.237110093	0.609324336	9.441534042	3.10887098	-1.027485967	0.6129826	0.052580763
avi	0.297775239	0.180772483	0.197654516	0.16879122	9.314641953	3.12084198	3.848116398	0.4879261	0.005620332
avi	0.31183061	0.17680043	0.135604024	0.304063618	9.32421875	2.87064791	4.659127235	0.7747599	0.006896203
avi	0.323555857	0.197510391	0.224162325	-0.486022949	9.759963036	1.74657011	-4.004348278	0.4602507	0.163814053
avi	0.313331723	0.188736051	0.216471806	0.141257897	9.736021042	1.9476831	1.219220757	0.7358698	0.009834002

username	w_mouth	d_mouth_nose	d_lc_nose	accelx	accely	accelz	eulerz	reo	smiling
awantha	0.303448558	0.160270855	0.198673829	-0.239420176	8.334216118	5.04817438	6.467576981	0.8095062	0.006952814
awantha	0.468144864	0.152966022	0.199664295	-0.310049117	8.593987465	4.72256279	0.094620228	0.9101996	0.039668944
awantha	0.269287765	0.162135676	0.184439451	-0.329202741	8.506598473	4.59926128	-3.352375746	0.6224257	0.129963055
awantha	0.267177254	0.144354194	0.181274071	-0.368707061	8.805873871	4.22816038	-2.620587826	0.9876325	0.02001627
awantha	0.278461963	0.165530503	0.184405342	-0.211886853	8.661025047	4.44962406	0.20171386	0.8140134	0.117730774
awantha	0.281397164	0.16291362	0.168128893	-0.283712894	8.805873871	3.96599507	-2.44212389	0.9749476	0.129091978
awantha	0.29404965	0.170904011	0.215203837	-0.351947635	7.243657112	6.34463453	2.26203227	0.5105275	0.093713135
awantha	0.297166198	0.166017786	0.195547163	-0.294486821	7.930793285	5.57968712	0.712036371	-1	-1
awantha	0.275548249	0.157006368	0.203747496	-0.281318694	7.559691906	5.64313364	6.729017258	0.445387	0.102012284
awantha	0.279976189	0.175854862	0.200988293	-0.549469292	7.614758492	6.02141714	-2.986623049	0.6619588	0.051992562
awantha	0.300878674	0.17145671	0.20309557	-0.407014281	7.669825077	5.96395636	-0.8671875	0.9871984	0.067564756
awantha	0.24264273	0.17715098	0.177482545	0.195127442	7.438784599	6.37336493	-0.658309758	0.1564318	0.024867583
awantha	1.345627427	0.104103819	0.190460011	0.345962137	7.292738438	6.67982292	-4.872186184	0.2097466	-1
awantha	1.170058608	0.172057614	0.191678599	0.077811554	6.967126846	6.7684083	-1.908445477	0.2562115	-1

username	w_mouth	d_mouth_nose	d_lc_nose	accelx	accely	accelz	eulerz	reo	smiling
dhanu	0.277150273	0.172897086	0.200401336	-0.068234749	7.378929615	6.31590414	-1.230358243	0.1601926	0.070463598
dhanu	0.284513682	0.947197795	0.209478363	0.603338838	6.999448776	6.82347488	-0.473074675	0.8261403	0.003846177
dhanu	0.297149181	0.192903563	0.225553468	0.551863492	7.115567684	6.60081387	1.732810736	0.989478	0.004488089
dhanu	0.265707046	0.164946765	0.194792032	0.29688102	7.152677536	6.55532408	-1.180299163	0.9920503	0.00392279
dhanu	0.288085103	0.160393253	0.181578457	0.312443316	7.149086475	6.54694462	-1.429446101	0.9939517	0.005909957
dhanu	0.291836977	0.180804148	0.172760874	-0.079008654	7.269993305	6.45835924	0.340985745	0.9457046	0.019632043
dhanu	1.141905546	0.995859146	0.995859146	0.239420176	7.127538681	6.56370401	0	0.9682148	-1
dhanu	0.311755002	0.192063659	0.22992219	0.304063618	6.84023428	6.81030655	-2.509252787	0.9947637	0.016626639
dhanu	0.253342777	0.140975714	0.179791614	0.541089594	6.930016994	6.74326897	-1.746204257	0.8699011	0.006236832
dhanu	0.263431609	0.164447606	0.201502308	-0.034715924	6.80791235	6.81868649	-0.98700124	0.9900634	0.080308959
dhanu	0.255895108	0.164934739	0.199819684	0.056263741	6.924031258	6.73488951	-3.165597916	0.885469	0.004624262
dhanu	0.284231633	0.180520639	0.204156682	0.48482585	6.718130112	6.86178207	-1.223036408	0.9929861	0.005712755
dhanu	0.290008336	0.19370532	0.145679355	0.257376671	6.709750175	6.9108634	-1.53504169	0.9939173	0.006060517
dhanu	0.275918514	0.139433622	0.255261004	0.234631762	6.918045998	6.82826328	-2.556207657	0.8356214	0.006904561

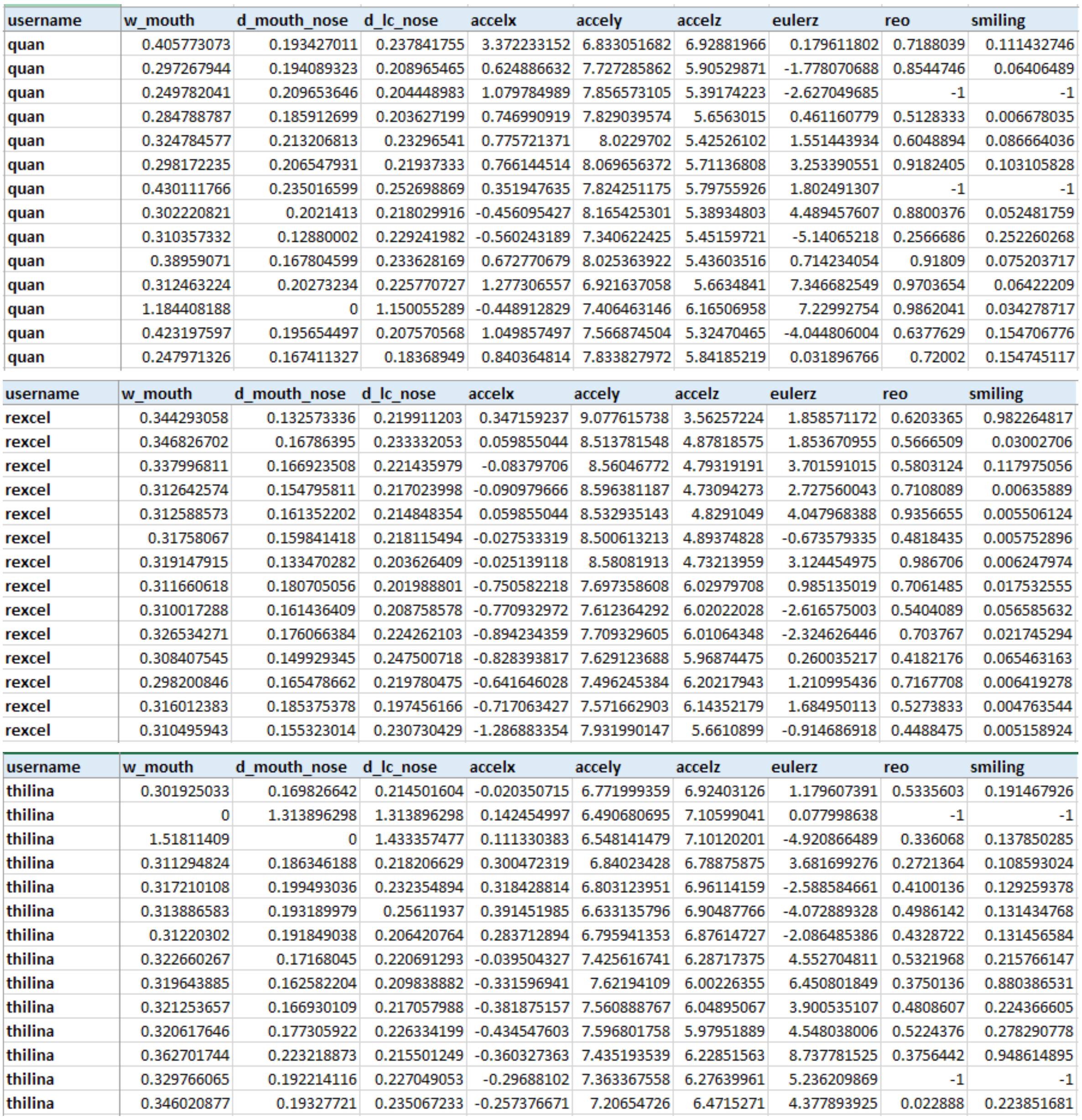


username	w_mouth	d_mouth_nose	d_lc_nose	accelx	accely	accelz	eulerz	reo	smiling
iamneil	0.262123585	0.174879804	0.198883936	0.37349546	7.438784599	6.10760832	0.543648064	0.9659146	0.17519629
iamneil	0.231332362	0.169930607	0.190679789	0.815225661	7.305906296	6.24407816	0.639749944	0.8745528	0.253053874
iamneil	0.228331909	0.171036959	0.166048959	0.529118598	7.47469759	6.06331587	0.587676704	0.9863937	0.129105628
iamneil	0.289070427	0.200239763	0.221916437	0.991199493	7.806294441	5.78558826	-0.766194522	0.9885893	0.073164843
iamneil	0.251092464	0.185430273	0.20504722	1.013944387	7.866149902	5.72214222	-2.182242393	0.987655	0.057579063
iamneil	0.287943542	0.190251499	0.200575814	0.894234359	7.278373241	6.3099184	-3.784981251	0.6018596	0.21216175
iamneil	0.252092749	0.229332745	0.164811179	0.770932972	7.844601631	5.74249268	-2.440301895	0.9896614	0.393085808
iamneil	0.275188804	0.185129821	0.192552567	1.061828494	7.228095055	6.43561411	-0.510506928	0.9888957	0.370620877
iamneil	0.282793313	0.196873263	0.214383245	0.975637197	7.386112213	6.22612143	-3.803832054	0.396855	0.283010572
iamneil	0.26649642	0.176922709	0.19245474	0.434547603	7.528567314	5.81431866	-1.682431936	0.9877506	0.310943812
iamneil	0.240889236	0.170732915	0.191436931	1.110909581	7.923610687	5.40012217	-1.847337842	0.9890745	0.292582214
iamneil	0.240889236	0.170732915	0.191436931	1.110909581	7.923610687	5.40012217	-1.847337842	0.9890745	0.292582214
iamneil	0.34722355	0.16648896	0.189864576	1.091755986	7.770381451	5.49589014	-1.847337842	0.9890745	0.292582214
iamneil	0.253862798	0.186777771	0.207067341	1.522712231	7.343016624	6.00824928	2.33219862	0.8604044	0.106808148

username	w_mouth	d_mouth_nose	d_lc_nose	accelx	accely	accelz	eulerz	reo	smiling
inoshi	0.34013027	0.140363485	0.217572048	0.351947635	6.50145483	7.46751499	0.790526748	0.4641831	0.080901243
inoshi	0.298995703	0.181074768	0.242140844	0.172382519	6.467936039	7.13950968	-0.534540534	0.4931637	0.064084858
inoshi	0	0	0	0.337582439	6.428431511	7.19457626	-0.695622742	-1	-1
inoshi	0.3303608	0.178913131	0.2181108	0.265756398	6.470330238	7.11796188	0.954530716	0.5049675	0.006627422
inoshi	0.349213511	0.173955753	0.227218866	0.300472319	6.470330238	7.08085155	-3.846623659	0.451715	0.006027412
inoshi	0.33304292	0.193564311	0.225627914	0.294486821	6.470330238	7.10599041	0.286522895	0.6646609	0.009301401
inoshi	0.286490709	0.179412723	0.21666491	0.195127442	6.521805286	7.15746593	1.680425286	0.9259344	0.240038589
inoshi	0.353441894	0.181502953	0.222447544	0.660799682	7.231686115	6.42603731	-0.259802848	0.5247808	0.020164255
inoshi	0.382874161	0.190613598	0.206757024	0.78050977	7.295132637	6.26083755	2.403627396	-1	-1
inoshi	0.349097699	0.195973262	0.204450488	0.752976418	7.21253252	6.38054752	-0.882814527	0.5712974	-1
inoshi	0.321024567	0.157293111	0.221505463	0.731428623	7.287950039	6.34822559	2.460202694	0.5170658	0.062559471
inoshi	0.338212222	0.17534104	0.220429584	0.524330199	7.352593422	6.32787514	0.226526305	0.4302201	0.004986227
inoshi	0.326999575	0.150689483	0.219014689	0.638054729	7.315483093	6.29555321	-0.704122663	0.5880947	0.006350692
inoshi	0.343476146	0.160838455	0.222446933	0.825999558	7.181407928	6.4834981	-0.606820166	-1	-1

username	w_mouth	d_mouth_nose	d_lc_nose	accelx	accely	accelz	eulerz	reo	smiling
jaliya	0.277403057	0.184623182	0.198968858	0.642843127	9.263166428	3.16752887	-0.493123889	-1	-1
jaliya	0.298201859	0.206141651	0.216392294	0.320823014	8.459911346	5.01345825	-5.397105217	0.575502	0.006949307
jaliya	0.39077276	0.12779884	0.224157527	0.913387954	8.297105789	5.05894804	-1.020000815	0.7848431	0.005877218
jaliya	0.311245292	0.212370664	0.220638812	0.66917938	8.604761124	4.8123455	-1.635026693	0.5799505	0.048873778
jaliya	0.318954378	0.203303486	0.22618632	0.888248801	8.342596054	4.98951626	-0.73720181	-1	-1
jaliya	0.298793018	0.196783409	0.219491482	0.785298169	8.50420475	4.73213959	-4.170794964	0.5973894	0.031071758
jaliya	0.317240298	0.205187857	0.224727854	0.951695204	8.451532364	4.4041338	1.366176248	0.8729632	0.076088242
jaliya	0.27678442	0.184931844	0.206023976	0.118512981	7.216124058	6.45117664	-5.176444054	0.1885812	0.200189963
jaliya	0.266129971	0.183121786	0.210765287	0.04309563	7.164648533	6.44040251	3.739672661	0.8947589	0.081066936
jaliya	0.264747024	1.078709006	1.078709006	0.286107093	5.479130745	8.34020138	12.31344318	0.5859514	0.077507406
jaliya	0.313904196	0.187272638	0.189620078	-0.40821138	6.308721542	6.20936203	-2.354522228	0.5502619	0.417429686
jaliya	0.278888524	0.179839715	0.221294969	-0.10055647	6.576871872	7.11796188	-5.187914848	0.3190063	0.211143434
jaliya	1.235804677	0.184023991	0.198096439	0.090979666	6.549338818	7.06888056	-4.748052597	0.4642924	-1
jaliya	0.293445677	0.186201423	0.212535545	0.045489833	6.685808182	7.15746593	-2.245727301	0.7348195	0.006284627

username	w_mouth	d_mouth_nose	d_lc_nose	accelx	accely	accelz	eulerz	reo	smiling
kasun	0.325955749	0.171514452	0.217301756	0.180762231	6.592434406	6.7636199	1.490131736	0.6961268	0.005498616
kasun	0.316205412	0.153874919	0.206263855	0.141257897	8.119935036	5.49110174	4.507544041	0.6870319	0.005633516
kasun	0.30738759	0.165218621	0.21516639	-0.090979666	8.157045364	5.15591335	1.638399243	0.6593289	0.005977408
kasun	0.324420214	0.170735508	0.212419286	0.430956304	8.35696125	5.10683203	4.164569378	-1	-1
kasun	0.356290042	0.186033428	0.257182211	0.174776718	6.43321991	7.26759911	4.400214672	-1	-1
kasun	0.326441318	0.181850806	0.268073529	0.577002585	7.601590633	6.05613327	3.889373302	0.7844738	0.00512587
kasun	0.312952638	0.171342865	0.205976039	0.269347697	6.87016201	6.52539682	1.44609499	0.5115333	0.006480553
kasun	0.330561846	0.215004608	0.329655409	0.235828862	6.107608318	7.48666859	0.299656391	0.4156728	0.006080637
kasun	0.314051658	0.163582385	0.207315475	0.37110126	8.464699745	5.23731613	6.287098885	0.5275583	0.011754069
kasun	0.30719927	0.157126859	0.201875985	0.112527482	8.851364136	4.13119507	7.147417545	0.0270175	0.016515385
kasun	0.313016295	0.158954158	0.204504043	0.362721562	7.258022308	6.81270075	6.153568745	-1	-1
kasun	0.309329361	0.151819095	0.207487166	0.318428814	8.347384453	5.07331324	5.017402649	0.5897638	0.006003815
kasun	0.304896623	0.213494018	0.19921273	0.368707061	7.340622425	6.65348673	6.28319931	0.3264161	0.005166134
kasun	0.323162824	0.164685547	0.218412399	0.509964943	8.425195694	4.92008448	4.229332924	0.6031295	0.005760107



username	w_mouth	d_mouth_nose	d_lc_nose	accelx	accely	accelz	eulerz	reo	smiling
quan	0.405773073	0.193427011	0.237841755	3.372233152	6.833051682	6.92881966	0.179611802	0.7188039	0.111432746
quan	0.297267944	0.194089323	0.208965465	0.624886632	7.727285862	5.90529871	-1.778070688	0.8544746	0.06406489
quan	0.249782041	0.209653646	0.204448983	1.079784989	7.856573105	5.39174223	-2.627049685	-1	-1
quan	0.284788787	0.185912699	0.203627199	0.746990919	7.829039574	5.6563015	0.461160779	0.5128333	0.006678035
quan	0.324784577	0.213206813	0.23296541	0.775721371	8.0229702	5.42526102	1.551443934	0.6048894	0.086664036
quan	0.298172235	0.206547931	0.21937333	0.766144514	8.069656372	5.71136808	3.253390551	0.9182405	0.103105828
quan	0.430111766	0.235016599	0.252698869	0.351947635	7.824251175	5.79755926	1.802491307	-1	-1
quan	0.302220821	0.2021413	0.218029916	-0.456095427	8.165425301	5.38934803	4.489457607	0.8800376	0.052481759
quan	0.310357332	0.12880002	0.229241982	-0.560243189	7.340622425	5.45159721	-5.14065218	0.2566686	0.252260268
quan	0.38959071	0.167804599	0.233628169	0.672770679	8.025363922	5.43603516	0.714234054	0.91809	0.075203717
quan	0.312463224	0.20273234	0.225770727	1.277306557	6.921637058	5.6634841	7.346682549	0.9703654	0.06422209
quan	1.184408188	0	1.150055289	-0.448912829	7.406463146	6.16506958	7.22992754	0.9862041	0.034278717
quan	0.423197597	0.195654497	0.207570568	1.049857497	7.566874504	5.32470465	-4.044806004	0.6377629	0.154706776
quan	0.247971326	0.167411327	0.18368949	0.840364814	7.833827972	5.84185219	0.031896766	0.72002	0.154745117

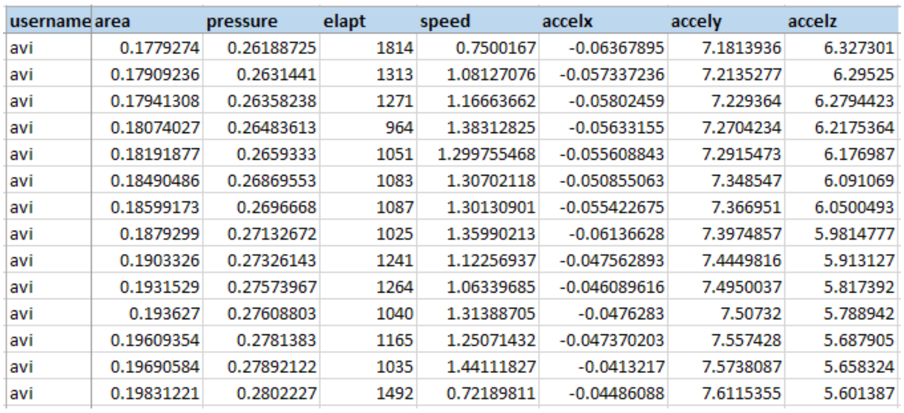
username	w_mouth	d_mouth_nose	d_lc_nose	accelx	accely	accelz	eulerz	reo	smiling
rexcel	0.344293058	0.132573336	0.219911203	0.347159237	9.077615738	3.56257224	1.858571172	0.6203365	0.982264817
rexcel	0.346826702	0.16786395	0.233332053	0.059855044	8.513781548	4.87818575	1.853670955	0.5666509	0.03002706
rexcel	0.337996811	0.166923508	0.221435979	-0.08379706	8.56046772	4.79319191	3.701591015	0.5803124	0.117975056
rexcel	0.312642574	0.154795811	0.217023998	-0.090979666	8.596381187	4.73094273	2.727560043	0.7108089	0.00635889
rexcel	0.312588573	0.161352202	0.214848354	0.059855044	8.532935143	4.8291049	4.047968388	0.9356655	0.005506124
rexcel	0.31758067	0.159841418	0.218115494	-0.027533319	8.500613213	4.89374828	-0.673579335	0.4818435	0.005752896
rexcel	0.319147915	0.133470282	0.203626409	-0.025139118	8.58081913	4.73213959	3.124454975	0.986706	0.006247974
rexcel	0.311660618	0.180705056	0.201988801	-0.750582218	7.697358608	6.02979708	0.985135019	0.7061485	0.017532555
rexcel	0.310017288	0.161436409	0.208758578	-0.770932972	7.612364292	6.02022028	-2.616575003	0.5404089	0.056585632
rexcel	0.326534271	0.176066384	0.224262103	-0.894234359	7.709329605	6.01064348	-2.324626446	0.703767	0.021745294
rexcel	0.308407545	0.149929345	0.247500718	-0.828393817	7.629123688	5.96874475	0.260035217	0.4182176	0.065463163
rexcel	0.298200846	0.165478662	0.219780475	-0.641646028	7.496245384	6.20217943	1.210995436	0.7167708	0.006419278
rexcel	0.316012383	0.185375378	0.197456166	-0.717063427	7.571662903	6.14352179	1.684950113	0.5273833	0.004763544
rexcel	0.310495943	0.155323014	0.230730429	-1.286883354	7.931990147	5.6610899	-0.914686918	0.4488475	0.005158924

username	w_mouth	d_mouth_nose	d_lc_nose	accelx	accely	accelz	eulerz	reo	smiling
thilina	0.301925033	0.169826642	0.214501604	-0.020350715	6.771999359	6.92403126	1.179607391	0.5335603	0.191467926
thilina	0	1.313896298	1.313896298	0.142454997	6.490680695	7.10599041	0.077998638	-1	-1
thilina	1.51811409	0	1.433357477	0.111330383	6.548141479	7.10120201	-4.920866489	0.336068	0.137850285
thilina	0.311294824	0.186346188	0.218206629	0.300472319	6.84023428	6.78875875	3.681699276	0.2721364	0.108593024
thilina	0.317210108	0.199493036	0.232354894	0.318428814	6.803123951	6.96114159	-2.588584661	0.4100136	0.129259378
thilina	0.313886583	0.193189979	0.25611937	0.391451985	6.633135796	6.90487766	-4.072889328	0.4986142	0.131434768
thilina	0.31220302	0.191849038	0.206420764	0.283712894	6.795941353	6.87614727	-2.086485386	0.4328722	0.131456584
thilina	0.322660267	0.17168045	0.220691293	-0.039504327	7.425616741	6.28717375	4.552704811	0.5321968	0.215766147
thilina	0.319643885	0.162582204	0.209838882	-0.331596941	7.62194109	6.00226355	6.450801849	0.3750136	0.880386531
thilina	0.321253657	0.166930109	0.217057988	-0.381875157	7.560888767	6.04895067	3.900535107	0.4808607	0.224366605
thilina	0.320617646	0.177305922	0.226334199	-0.434547603	7.596801758	5.97951889	4.548038006	0.5224376	0.278290778
thilina	0.362701744	0.223218873	0.215501249	-0.360327363	7.435193539	6.22851563	8.737781525	0.3756442	0.948614895
thilina	0.329766065	0.192214116	0.227049053	-0.29688102	7.363367558	6.27639961	5.236209869	-1	-1
thilina	0.346020877	0.19327721	0.235067233	-0.257376671	7.20654726	6.4715271	4.377893925	0.022888	0.223851681

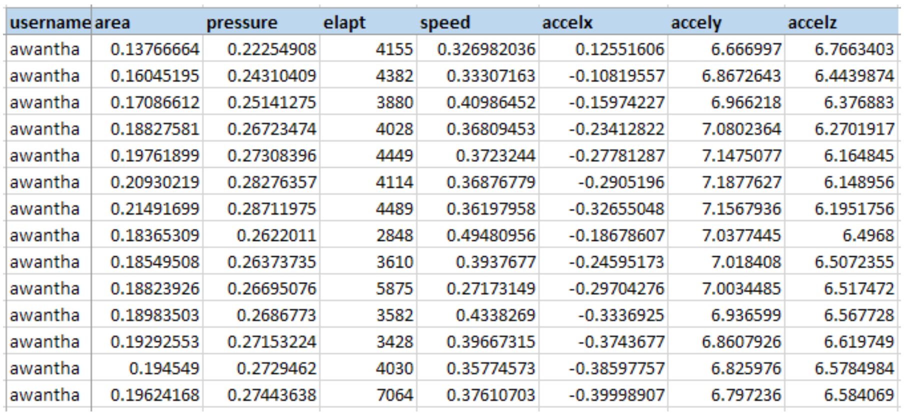
Please refer ”2013MIS015-CD-1” for more details (”train/face”)

Appendix B

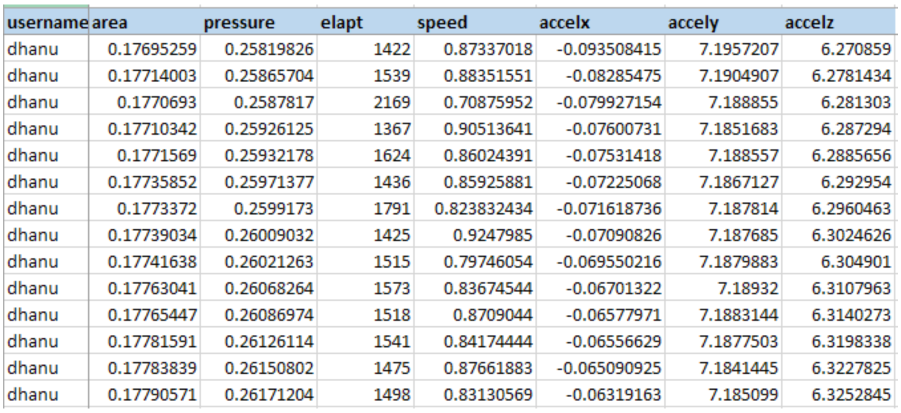
Touch Training Dataset



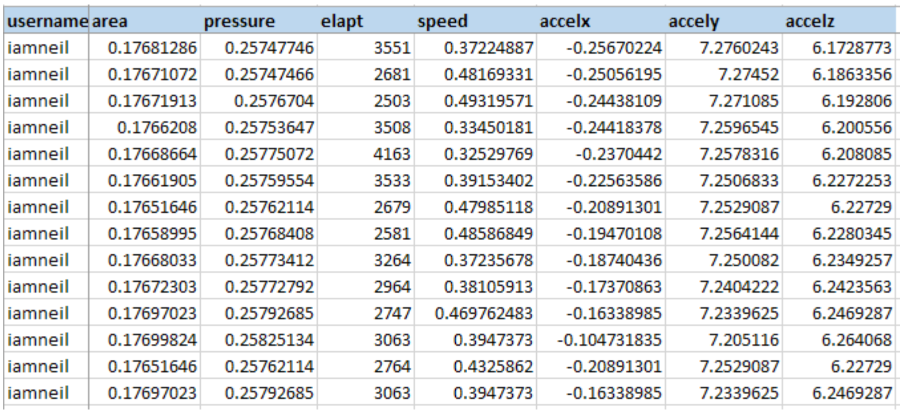
username	area	pressure	elapt	speed	accelx	accely	accelz
avi	0.1779274	0.26188725	1814	0.7500167	-0.06367895	7.1813936	6.327301
avi	0.17909236	0.2631441	1313	1.08127076	-0.057337236	7.2135277	6.29525
avi	0.17941308	0.26358238	1271	1.16663662	-0.05802459	7.229364	6.2794423
avi	0.18074027	0.26483613	964	1.38312825	-0.05633155	7.2704234	6.2175364
avi	0.18191877	0.2659333	1051	1.299755468	-0.055608843	7.2915473	6.176987
avi	0.18490486	0.26869553	1083	1.30702118	-0.050855063	7.348547	6.091069
avi	0.18599173	0.2696668	1087	1.30130901	-0.055422675	7.366951	6.0500493
avi	0.1879299	0.27132672	1025	1.35990213	-0.06136628	7.3974857	5.9814777
avi	0.1903326	0.27326143	1241	1.12256937	-0.047562893	7.4449816	5.913127
avi	0.1931529	0.27573967	1264	1.06339685	-0.046089616	7.4950037	5.817392
avi	0.193627	0.27608803	1040	1.31388705	-0.0476283	7.50732	5.788942
avi	0.19609354	0.2781383	1165	1.25071432	-0.047370203	7.557428	5.687905
avi	0.19690584	0.27892122	1035	1.44111827	-0.0413217	7.5738087	5.658324
avi	0.19831221	0.2802227	1492	0.72189811	-0.04486088	7.6115355	5.601387



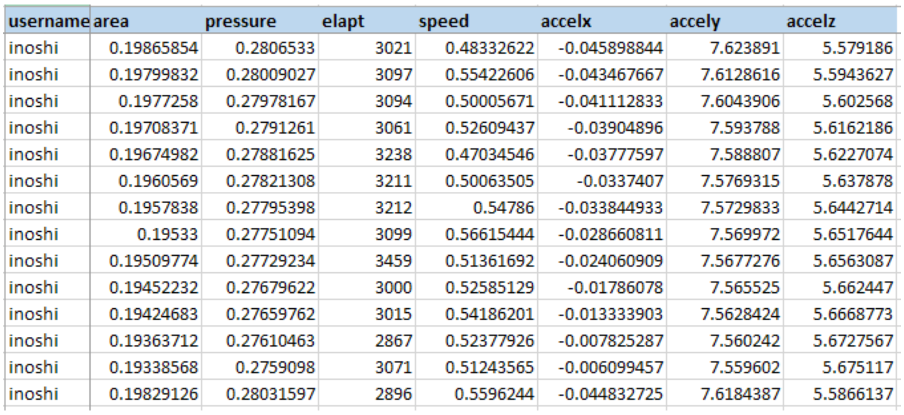
username	area	pressure	elapt	speed	accelx	accely	accelz
awantha	0.13766664	0.22254908	4155	0.326982036	0.12551606	6.666997	6.7663403
awantha	0.16045195	0.24310409	4382	0.33307163	-0.10819557	6.8672643	6.4439874
awantha	0.17086612	0.25141275	3880	0.40986452	-0.15974227	6.966218	6.376883
awantha	0.18827581	0.26723474	4028	0.36809453	-0.23412822	7.0802364	6.2701917
awantha	0.19761899	0.27308396	4449	0.3723244	-0.27781287	7.1475077	6.164845
awantha	0.20930219	0.28276357	4114	0.36876779	-0.2905196	7.1877627	6.148956
awantha	0.21491699	0.28711975	4489	0.36197958	-0.32655048	7.1567936	6.1951756
awantha	0.18365309	0.2622011	2848	0.49480956	-0.18678607	7.0377445	6.4968
awantha	0.18549508	0.26373735	3610	0.3937677	-0.24595173	7.018408	6.5072355
awantha	0.18823926	0.26695076	5875	0.27173149	-0.29704276	7.0034485	6.517472
awantha	0.18983503	0.2686773	3582	0.4338269	-0.3336925	6.936599	6.567728
awantha	0.19292553	0.27153224	3428	0.39667315	-0.3743677	6.8607926	6.619749
awantha	0.194549	0.2729462	4030	0.35774573	-0.38597757	6.825976	6.5784984
awantha	0.19624168	0.27443638	7064	0.37610703	-0.39998907	6.797236	6.584069



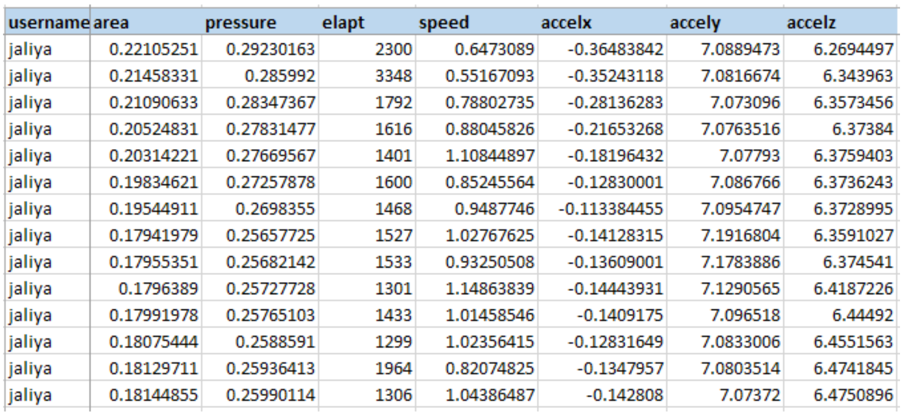
username	area	pressure	elapt	speed	accelx	accely	accelz
dhanu	0.17695259	0.25819826	1422	0.87337018	-0.093508415	7.1957207	6.270859
dhanu	0.17714003	0.25865704	1539	0.88351551	-0.08285475	7.1904907	6.2781434
dhanu	0.1770693	0.2587817	2169	0.70875952	-0.079927154	7.188855	6.281303
dhanu	0.17710342	0.25926125	1367	0.90513641	-0.07600731	7.1851683	6.287294
dhanu	0.1771569	0.25932178	1624	0.86024391	-0.07531418	7.188557	6.2885656
dhanu	0.17735852	0.25971377	1436	0.85925881	-0.07225068	7.1867127	6.292954
dhanu	0.1773372	0.2599173	1791	0.823832434	-0.071618736	7.187814	6.2960463
dhanu	0.17739034	0.26009032	1425	0.9247985	-0.07090826	7.187685	6.3024626
dhanu	0.17741638	0.26021263	1515	0.79746054	-0.069550216	7.1879883	6.304901
dhanu	0.17763041	0.26068264	1573	0.83674544	-0.06701322	7.18932	6.3107963
dhanu	0.17765447	0.26086974	1518	0.8709044	-0.06577971	7.1883144	6.3140273
dhanu	0.17781591	0.26126114	1541	0.84174444	-0.06556629	7.1877503	6.3198338
dhanu	0.17783839	0.26150802	1475	0.87661883	-0.065090925	7.1841445	6.3227825
dhanu	0.17790571	0.26171204	1498	0.83130569	-0.06319163	7.185099	6.3252845



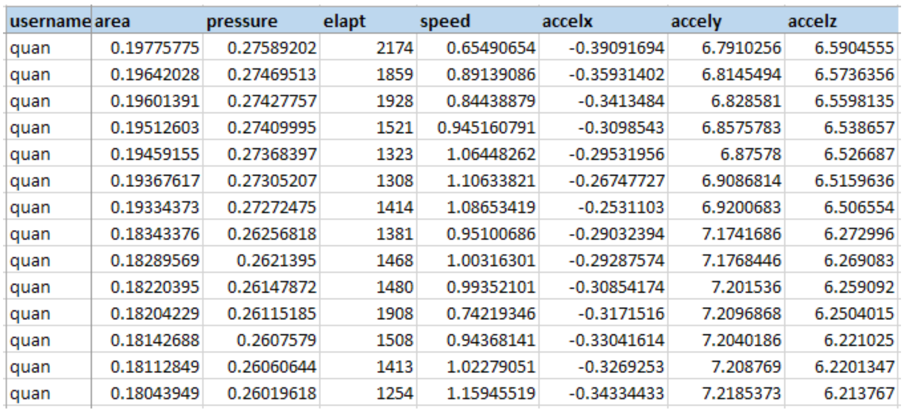
username	area	pressure	elapt	speed	accelx	accely	accelz
iamneil	0.17681286	0.25747746	3551	0.37224887	-0.25670224	7.2760243	6.1728773
iamneil	0.17671072	0.25747466	2681	0.48169331	-0.25056195	7.27452	6.1863356
iamneil	0.17671913	0.2576704	2503	0.49319571	-0.24438109	7.271085	6.192806
iamneil	0.1766208	0.25753647	3508	0.33450181	-0.24418378	7.2596545	6.200556
iamneil	0.17668664	0.25775072	4163	0.32529769	-0.2370442	7.2578316	6.208085
iamneil	0.17661905	0.25759554	3533	0.39153402	-0.22563586	7.2506833	6.2272253
iamneil	0.17651646	0.25762114	2679	0.47985118	-0.20891301	7.2529087	6.22729
iamneil	0.17658995	0.25768408	2581	0.48586849	-0.19470108	7.2564144	6.2280345
iamneil	0.17668033	0.25773412	3264	0.37235678	-0.18740436	7.250082	6.2349257
iamneil	0.17672303	0.25772792	2964	0.38105913	-0.17370863	7.2404222	6.2423563
iamneil	0.17697023	0.25792685	2747	0.469762483	-0.16338985	7.2339625	6.2469287
iamneil	0.17699824	0.25825134	3063	0.3947373	-0.104731835	7.205116	6.264068
iamneil	0.17651646	0.25762114	2764	0.4325862	-0.20891301	7.2529087	6.22729
iamneil	0.17697023	0.25792685	3063	0.3947373	-0.16338985	7.2339625	6.2469287



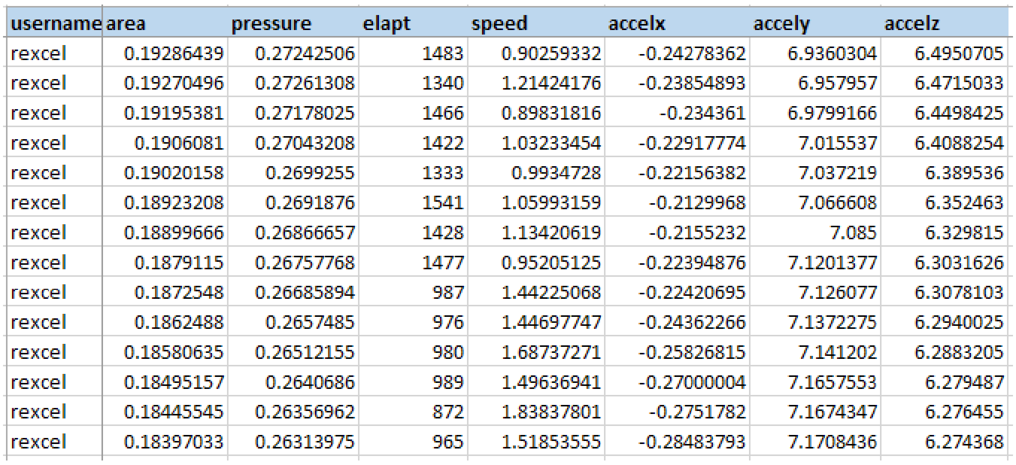
username	area	pressure	elapt	speed	accelx	accely	accelz
inoshi	0.19865854	0.2806533	3021	0.48332622	-0.045898844	7.623891	5.579186
inoshi	0.19799832	0.28009027	3097	0.55422606	-0.043467667	7.6128616	5.5943627
inoshi	0.1977258	0.27978167	3094	0.50005671	-0.041112833	7.6043906	5.602568
inoshi	0.19708371	0.2791261	3061	0.52609437	-0.03904896	7.593788	5.6162186
inoshi	0.19674982	0.27881625	3238	0.47034546	-0.03777597	7.588807	5.6227074
inoshi	0.1960569	0.27821308	3211	0.50063505	-0.0337407	7.5769315	5.637878
inoshi	0.1957838	0.27795398	3212	0.54786	-0.033844933	7.5729833	5.6442714
inoshi	0.19533	0.27751094	3099	0.56615444	-0.028660811	7.569972	5.6517644
inoshi	0.19509774	0.27729234	3459	0.51361692	-0.024060909	7.5677276	5.6563087
inoshi	0.19452232	0.27679622	3000	0.52585129	-0.01786078	7.565525	5.662447
inoshi	0.19424683	0.27659762	3015	0.54186201	-0.013333903	7.5628424	5.6668773
inoshi	0.19363712	0.27610463	2867	0.52377926	-0.007825287	7.560242	5.6727567
inoshi	0.19338568	0.2759098	3071	0.51243565	-0.006099457	7.559602	5.675117
inoshi	0.19829126	0.28031597	2896	0.5596244	-0.044832725	7.6184387	5.5866137



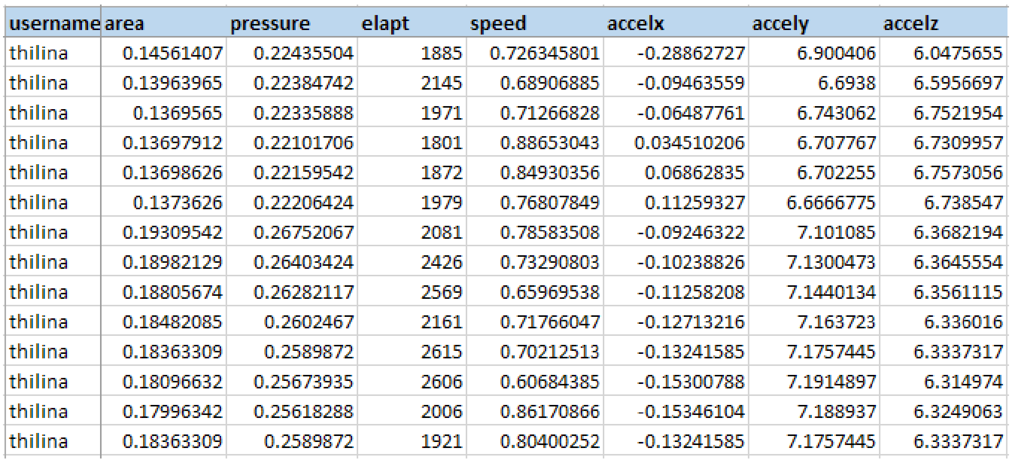
username	area	pressure	elapt	speed	accelx	accely	accelz
jaliya	0.22105251	0.29230163	2300	0.6473089	-0.36483842	7.0889473	6.2694497
jaliya	0.21458331	0.285992	3348	0.55167093	-0.35243118	7.0816674	6.343963
jaliya	0.21090633	0.28347367	1792	0.78802735	-0.28136283	7.073096	6.3573456
jaliya	0.20524831	0.27831477	1616	0.88045826	-0.21653268	7.0763516	6.37384
jaliya	0.20314221	0.27669567	1401	1.10844897	-0.18196432	7.07793	6.3759403
jaliya	0.19834621	0.27257878	1600	0.85245564	-0.12830001	7.086766	6.3736243
jaliya	0.19544911	0.2698355	1468	0.9487746	-0.113384455	7.0954747	6.3728995
jaliya	0.17941979	0.25657725	1527	1.02767625	-0.14128315	7.1916804	6.3591027
jaliya	0.17955351	0.25682142	1533	0.93250508	-0.13609001	7.1783886	6.374541
jaliya	0.1796389	0.25727728	1301	1.14863839	-0.14443931	7.1290565	6.4187226
jaliya	0.17991978	0.25765103	1433	1.01458546	-0.1409175	7.096518	6.44492
jaliya	0.18075444	0.2588591	1299	1.02356415	-0.12831649	7.0833006	6.4551563
jaliya	0.18129711	0.25936413	1964	0.82074825	-0.1347957	7.0803514	6.4741845
jaliya	0.18144855	0.25990114	1306	1.04386487	-0.142808	7.07372	6.4750896



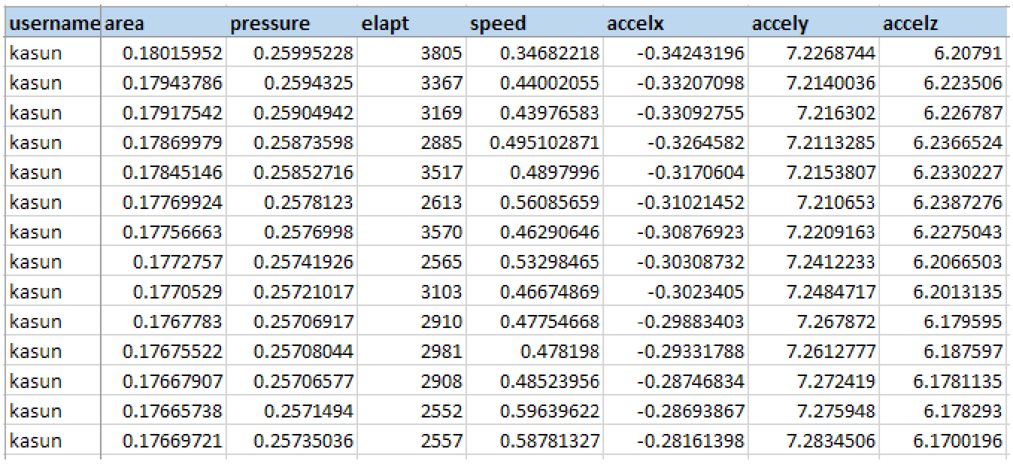
username	area	pressure	elapt	speed	accelx	accely	accelz
quan	0.19775775	0.27589202	2174	0.65490654	-0.39091694	6.7910256	6.5904555
quan	0.19642028	0.27469513	1859	0.89139086	-0.35931402	6.8145494	6.5736356
quan	0.19601391	0.27427757	1928	0.84438879	-0.3413484	6.828581	6.5598135
quan	0.19512603	0.27409995	1521	0.945160791	-0.3098543	6.8575783	6.538657
quan	0.19459155	0.27368397	1323	1.06448262	-0.29531956	6.87578	6.526687
quan	0.19367617	0.27305207	1308	1.10633821	-0.26747727	6.9086814	6.5159636
quan	0.19334373	0.27272475	1414	1.08653419	-0.2531103	6.9200683	6.506554
quan	0.18343376	0.26256818	1381	0.95100686	-0.29032394	7.1741686	6.272996
quan	0.18289569	0.2621395	1468	1.00316301	-0.29287574	7.1768446	6.269083
quan	0.18220395	0.26147872	1480	0.99352101	-0.30854174	7.201536	6.259092
quan	0.18204229	0.26115185	1908	0.74219346	-0.3171516	7.2096868	6.2504015
quan	0.18142688	0.2607579	1508	0.94368141	-0.33041614	7.2040186	6.221025
quan	0.18112849	0.26060644	1413	1.02279051	-0.3269253	7.208769	6.2201347
quan	0.18043949	0.26019618	1254	1.15945519	-0.34334433	7.2185373	6.213767



username	area	pressure	elapt	speed	accelx	accely	accelz
rexcel	0.19286439	0.27242506	1483	0.90259332	-0.24278362	6.9360304	6.4950705
rexcel	0.19270496	0.27261308	1340	1.21424176	-0.23854893	6.957957	6.4715033
rexcel	0.19195381	0.27178025	1466	0.89831816	-0.234361	6.9799166	6.4498425
rexcel	0.1906081	0.27043208	1422	1.03233454	-0.22917774	7.015537	6.4088254
rexcel	0.19020158	0.2699255	1333	0.9934728	-0.22156382	7.037219	6.389536
rexcel	0.18923208	0.2691876	1541	1.05993159	-0.2129968	7.066608	6.352463
rexcel	0.18899666	0.26866657	1428	1.13420619	-0.2155232	7.085	6.329815
rexcel	0.1879115	0.26757768	1477	0.95205125	-0.22394876	7.1201377	6.3031626
rexcel	0.1872548	0.26685894	987	1.44225068	-0.22420695	7.126077	6.3078103
rexcel	0.1862488	0.2657485	976	1.44697747	-0.24362266	7.1372275	6.2940025
rexcel	0.18580635	0.26512155	980	1.68737271	-0.25826815	7.141202	6.2883205
rexcel	0.18495157	0.2640686	989	1.49636941	-0.27000004	7.1657553	6.279487
rexcel	0.18445545	0.26356962	872	1.83837801	-0.2751782	7.1674347	6.276455
rexcel	0.18397033	0.26313975	965	1.51853555	-0.28483793	7.1708436	6.274368



username	area	pressure	elapt	speed	accelx	accely	accelz
thilina	0.14561407	0.22435504	1885	0.726345801	-0.28862727	6.900406	6.0475655
thilina	0.13963965	0.22384742	2145	0.68906885	-0.09463559	6.6938	6.5956697
thilina	0.1369565	0.22335888	1971	0.71266828	-0.06487761	6.743062	6.7521954
thilina	0.13697912	0.22101706	1801	0.88653043	0.034510206	6.707767	6.7309957
thilina	0.13698626	0.22159542	1872	0.84930356	0.06862835	6.702255	6.7573056
thilina	0.1373626	0.22206424	1979	0.76807849	0.11259327	6.6666775	6.738547
thilina	0.19309542	0.26752067	2081	0.78583508	-0.09246322	7.101085	6.3682194
thilina	0.18982129	0.26403424	2426	0.73290803	-0.10238826	7.1300473	6.3645554
thilina	0.18805674	0.26282117	2569	0.65969538	-0.11258208	7.1440134	6.3561115
thilina	0.18482085	0.2602467	2161	0.71766047	-0.12713216	7.163723	6.336016
thilina	0.18363309	0.2589872	2615	0.70212513	-0.13241585	7.1757445	6.3337317
thilina	0.18096632	0.25673935	2606	0.60684385	-0.15300788	7.1914897	6.314974
thilina	0.17996342	0.25618288	2006	0.86170866	-0.15346104	7.188937	6.3249063
thilina	0.18363309	0.2589872	1921	0.80400252	-0.13241585	7.1757445	6.3337317



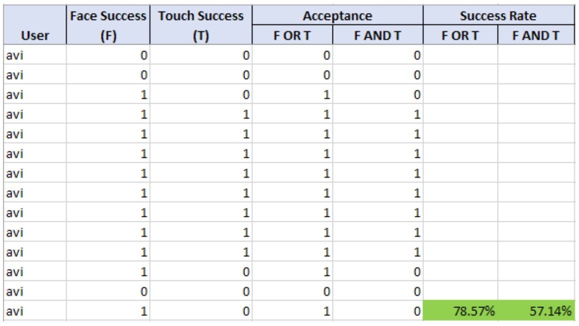
username	area	pressure	elapt	speed	accelx	accely	accelz
kasun	0.18015952	0.25995228	3805	0.34682218	-0.34243196	7.2268744	6.20791
kasun	0.17943786	0.2594325	3367	0.44002055	-0.33207098	7.2140036	6.223506
kasun	0.17917542	0.25904942	3169	0.43976583	-0.33092755	7.216302	6.226787
kasun	0.17869979	0.25873598	2885	0.495102871	-0.3264582	7.2113285	6.2366524
kasun	0.17845146	0.25852716	3517	0.4897996	-0.3170604	7.2153807	6.2330227
kasun	0.17769924	0.2578123	2613	0.56085659	-0.31021452	7.210653	6.2387276
kasun	0.17756663	0.2576998	3570	0.46290646	-0.30876923	7.2209163	6.2275043
kasun	0.1772757	0.25741926	2565	0.53298465	-0.30308732	7.2412233	6.2066503
kasun	0.1770529	0.25721017	3103	0.46674869	-0.3023405	7.2484717	6.2013135
kasun	0.1767783	0.25706917	2910	0.47754668	-0.29883403	7.267872	6.179595
kasun	0.17675522	0.25708044	2981	0.478198	-0.29331788	7.2612777	6.187597
kasun	0.17667907	0.25706577	2908	0.48523956	-0.28746834	7.272419	6.1781135
kasun	0.17665738	0.2571494	2552	0.59639622	-0.28693867	7.275948	6.178293
kasun	0.17669721	0.25735036	2557	0.58781327	-0.28161398	7.2834506	6.1700196

Please refer ”2013MIS015-CD-1” for more details (”train/touch”)

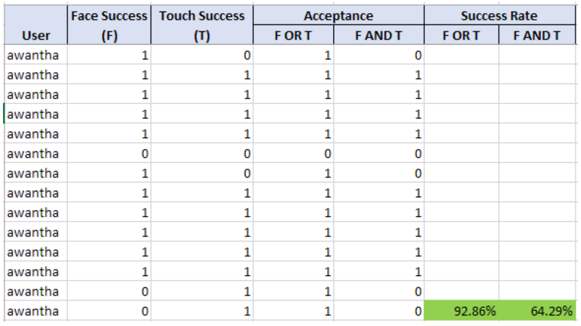
Appendix C

User Verification Result Analysis Using Training Dataset

Success Rates for User1 & User2

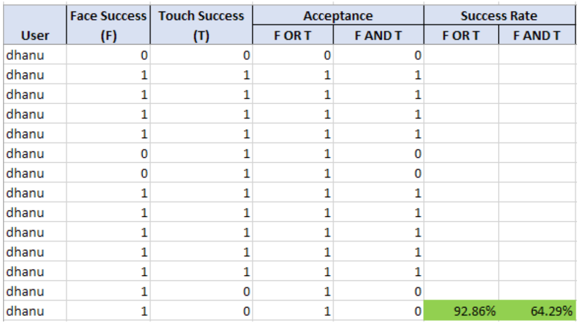


	Face Success	Touch Success	Acceptance		Success Rate	
User	(F)	(T)	F OR T	F AND T	F OR T	F AND T
avi	0	0	0	0		
avi	0	0	0	0		
avi	1	0	1	0		
avi	1	1	1	1		
avi	1	1	1	1		
avi	1	1	1	1		
avi	1	1	1	1		
avi	1	1	1	1		
avi	1	1	1	1		
avi	1	1	1	1		
avi	1	1	1	1		
avi	1	0	1	0		
avi	0	0	0	0		
avi	1	0	1	0	78.57%	57.14%

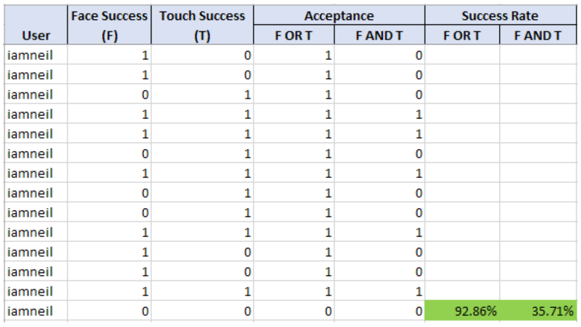


	Face Success	Touch Success	Acceptance		Success Rate	
User	(F)	(T)	F OR T	F AND T	F OR T	F AND T
awantha	1	0	1	0		
awantha	1	1	1	1		
awantha	1	1	1	1		
awantha	1	1	1	1		
awantha	1	1	1	1		
awantha	0	0	0	0		
awantha	1	0	1	0		
awantha	1	1	1	1		
awantha	1	1	1	1		
awantha	1	1	1	1		
awantha	1	1	1	1		
awantha	1	1	1	1		
awantha	0	1	1	0		
awantha	0	1	1	0	92.86%	64.29%

Success Rates for User3 & User4

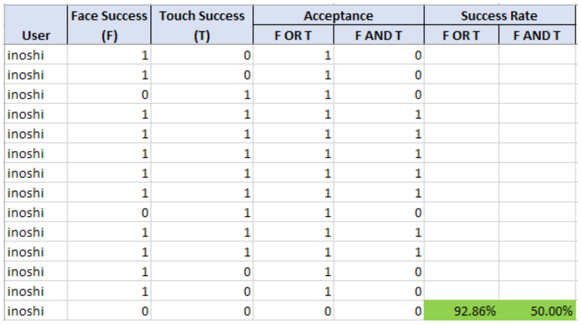


	Face Success	Touch Success	Acceptance		Success Rate	
User	(F)	(T)	F OR T	F AND T	F OR T	F AND T
dhanu	0	0	0	0		
dhanu	1	1	1	1		
dhanu	1	1	1	1		
dhanu	1	1	1	1		
dhanu	1	1	1	1		
dhanu	0	1	1	0		
dhanu	0	1	1	0		
dhanu	1	1	1	1		
dhanu	1	1	1	1		
dhanu	1	1	1	1		
dhanu	1	1	1	1		
dhanu	1	1	1	1		
dhanu	1	0	1	0		
dhanu	1	0	1	0	92.86%	64.29%

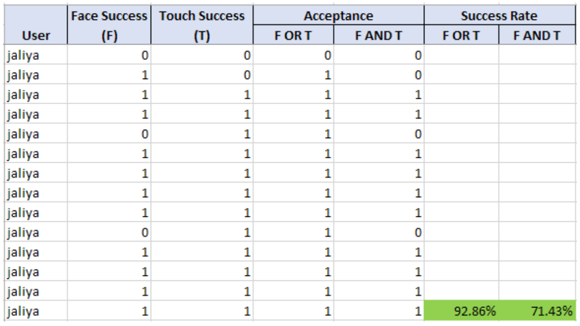


	Face Success	Touch Success	Acceptance		Success Rate	
User	(F)	(T)	F OR T	F AND T	F OR T	F AND T
iamneil	1	0	1	0		
iamneil	1	0	1	0		
iamneil	0	1	1	0		
iamneil	1	1	1	1		
iamneil	1	1	1	1		
iamneil	0	1	1	0		
iamneil	1	1	1	1		
iamneil	0	1	1	0		
iamneil	0	1	1	0		
iamneil	1	1	1	1		
iamneil	1	0	1	0		
iamneil	1	0	1	0		
iamneil	1	1	1	1		
iamneil	0	0	0	0	92.86%	35.71%

Success Rates for User5 & User6

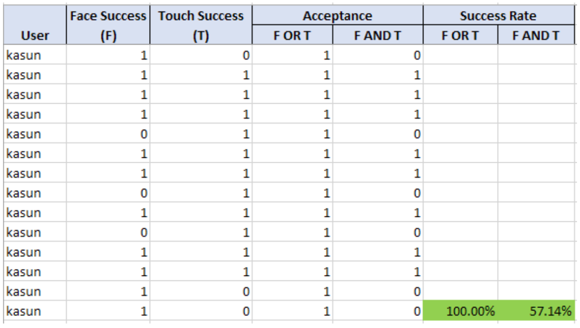


	Face Success	Touch Success	Acceptance		Success Rate	
User	(F)	(T)	F OR T	F AND T	F OR T	F AND T
inoshi	1	0	1	0		
inoshi	1	0	1	0		
inoshi	0	1	1	0		
inoshi	1	1	1	1		
inoshi	1	1	1	1		
inoshi	1	1	1	1		
inoshi	1	1	1	1		
inoshi	1	1	1	1		
inoshi	0	1	1	0		
inoshi	1	1	1	1		
inoshi	1	1	1	1		
inoshi	1	0	1	0		
inoshi	1	0	1	0		
inoshi	0	0	0	0	92.86%	50.00%

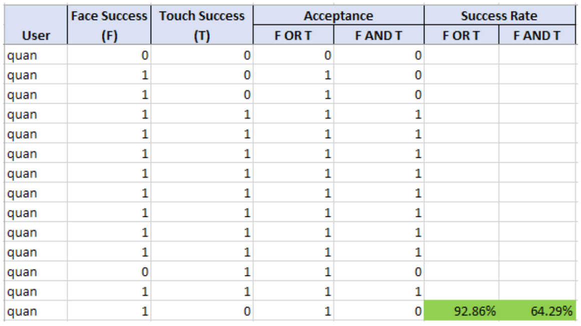


	Face Success	Touch Success	Acceptance		Success Rate	
User	(F)	(T)	F OR T	F AND T	F OR T	F AND T
jaliya	0	0	0	0		
jaliya	1	0	1	0		
jaliya	1	1	1	1		
jaliya	1	1	1	1		
jaliya	0	1	1	0		
jaliya	1	1	1	1		
jaliya	1	1	1	1		
jaliya	1	1	1	1		
jaliya	1	1	1	1		
jaliya	0	1	1	0		
jaliya	1	1	1	1		
jaliya	1	1	1	1		
jaliya	1	1	1	1		
jaliya	1	1	1	1	92.86%	71.43%

Success Rates for User7 & User8

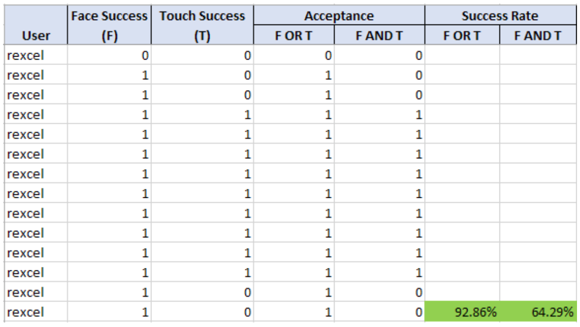


	Face Success	Touch Success	Acceptance		Success Rate	
User	(F)	(T)	F OR T	F AND T	F OR T	F AND T
kasun	1	0	1	0		
kasun	1	1	1	1		
kasun	1	1	1	1		
kasun	1	1	1	1		
kasun	0	1	1	0		
kasun	1	1	1	1		
kasun	1	1	1	1		
kasun	0	1	1	0		
kasun	1	1	1	1		
kasun	0	1	1	0		
kasun	1	1	1	1		
kasun	1	1	1	1		
kasun	1	0	1	0		
kasun	1	0	1	0	100.00%	57.14%

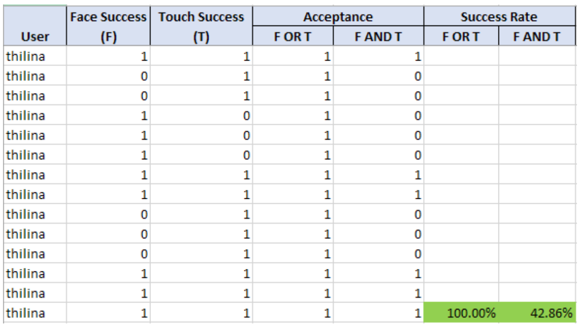


	Face Success	Touch Success	Acceptance		Success Rate	
User	(F)	(T)	F OR T	F AND T	F OR T	F AND T
quan	0	0	0	0		
quan	1	0	1	0		
quan	1	0	1	0		
quan	1	1	1	1		
quan	1	1	1	1		
quan	1	1	1	1		
quan	1	1	1	1		
quan	1	1	1	1		
quan	1	1	1	1		
quan	1	1	1	1		
quan	1	1	1	1		
quan	0	1	1	0		
quan	1	1	1	1		
quan	1	0	1	0	92.86%	64.29%

Success Rates for User9 & User10



	Face Success	Touch Success	Acceptance		Success Rate	
User	(F)	(T)	F OR T	F AND T	F OR T	F AND T
rexcel	0	0	0	0		
rexcel	1	0	1	0		
rexcel	1	0	1	0		
rexcel	1	1	1	1		
rexcel	1	1	1	1		
rexcel	1	1	1	1		
rexcel	1	1	1	1		
rexcel	1	1	1	1		
rexcel	1	1	1	1		
rexcel	1	1	1	1		
rexcel	1	1	1	1		
rexcel	1	1	1	1		
rexcel	1	0	1	0		
rexcel	1	0	1	0	92.86%	64.29%



	Face Success	Touch Success	Acceptance		Success Rate	
User	(F)	(T)	F OR T	F AND T	F OR T	F AND T
thilina	1	1	1	1		
thilina	0	1	1	0		
thilina	0	1	1	0		
thilina	1	0	1	0		
thilina	1	0	1	0		
thilina	1	0	1	0		
thilina	1	1	1	1		
thilina	1	1	1	1		
thilina	0	1	1	0		
thilina	0	1	1	0		
thilina	0	1	1	0		
thilina	1	1	1	1		
thilina	1	1	1	1		
thilina	1	1	1	1	100.00%	42.86%

Please refer ”2013MIS015-CD-1” for more details (”verification”)

Bibliography

[1] Bosomworth , Mobile Marketing Statistics (2015),[Online].Available:
http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile

[2] Rainie and J. Anderson, ”The Future of the Internet III”,[Online].Available:
http://www.pewinternet.org/Reports/2008/The-Future-of-the-Internet-III.aspx

[3] S. Furnell,N. Clarke and S. KaratzouniBhaumik, ”Beyond the pin: Enhancing user authentication for mobile devices”, Computer Fraud and Security. (2008)

[4] A. De Luca, A. Hang, F. Brudy, C. Lindner, H. Hussmann,”Touch me once and I know it’s you! Implicit Authentication based on Touch Screen Patterns”, (2012)

[5] D. L. Nelson, V. S. Reed, J. R. Walling, ”Pictorial superiority effect,” *Journal of Experimental Psychology: Human Learning and Memory*, ch. 2, sec. 5, pp. 523-528, 1976.

[6] S. Sonkamble, R. Thool, B. Sonkamble, ”Survey of biometric recognition systems and their applications,” *Journal of Theoretical and Applied Information Technology*, ch.11, sec. 1, pp.45-51,2010.

[7] M. Jakobsson, E. Shi, P. Golle, R. Chow, ”Implicit authentication for mobile devices,” *In Proceedings HotSec*, 2009, USENIX Association, 9-9.

[8] E. Shi, Y. Niu, M. Jakobsson, and R. Chow. Implicit Authentication through Learning User Behaviour, [Online] Available:
https://www.cs.umd.edu/ elaine/docs/isc.pdf

[9] A. De Luca, M. Denzel, H. Hussmann, ”Look into my eyes! Can you guess my password,” *Proc. SOUPS ACM*, 2009, vol. 7, pp. 1-12.

[10] W. Meng, D. S. Wong, L. Kwok, ”The Effect of Adaptive Mechanism on Behavioural Biometric Based Mobile Phone Authentication,” *Information Management and Computer Security*, 2014,ch. 22, sec.2 , pp.155-166.

[11] A. Pabbaraju, S. Puchakayala, ”Face Recognition Application on Android (2010),” [Online] Available:
http://web.eecs.umich.edu/ silvio/teaching/EECS598_2010/presentation/Aditya_Sru

[12] G. Duncan (2013), ”Why havent biometrics replaced passwords yet,” [Online] Available: http://www.digitaltrends.com/computing/can-biometrics-secure-our-digital-lives/

[13] R.V. Yampolskiy, V. Govindaraju, ”Behavioural biometrics: a survey and classification,” *Int. J. Biometrics*, Vol. 1, No. 1, pp.81-113, 2008

[14] *Two Factor Authentication Goes Mobile*.1st ed., Goode Intelligence, London, 2012

[15] S. Uellebeck, M. Dumuth, C. Wolf , T. Holz, ”Quantifying the Security of Graphical Passwords: The Case of Android Unlock Patterns,” CCS13 , Berlin, Germany, 2013, pp. 4-8

[16] C. Bhagavatula, Blase Ur, K. Iacovino, S. M. Kywey, L. F. Cranor, M. Savvides, ”Biometric Authentication on iPhone and Android: Usability, Perceptions, and Influences on Adoption,” *USEC* 2015, February 8, 2015

[17] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, ”The Quest to Replace Passwords: A framework for comparative evaluation of Web authentication schemes,” *in Proc. IEEE S&P*, 2012.

[18] Google, ”Introducing android 4.0,” [Online] Available: https://developer.android.com/about/versions/android-4.0-highlights.html.

[19] Apple, ”Using Touch ID on the iPhone,” [Online] Available: http://support.apple.com/kb/ht5883.

[20] R. D. Findling, R. Mayrhofer, ”Towards Face Unlock: On the Difficulty of Reliably Detecting Faces on Mobile Phones,” Bali, Indunesia, pp. 3-5. 2012.

[21] M. Turk and A. Pentland. ”Eigenfaces for recognition. Cognitive Neuroscience,” ch.3, sec. 1, pp.7186, Jan. 1991

[22] W. Kurutach, R. Fooprateepsiri, and S. Phoomvuthisarn. ”A highly robust approach face recognition using hausdorff-trace transformation,” *In ICONIP (2)*, pp. 549556, 2010.

[23] Sony, ”Get started with OpenCV on Android,” [Online] Available: http://developer.sonymobile.com/knowledge-base/tutorials/android_tutorial/get-s

[24] J. Keller, M. Gray, J. Givens. ”A Fuzzy K-Nearest Neighbor Algorithm,” *IEEE Man and Transactions on Systems*, Cybernectics, vol smc-15, no 4, July August 1985.

[25] H. Seiger, N. Kirschnick,S. Moller, Poster: User preferences for biometric authentication methods and graded security on mobile phones , Symposium on Usability, Privacy & Seurity, (SOUPS), 2010.

[26] P. S. Teh, N. Zhang, A. B. J. Teoh, K. Chen, A survey on touch dynamics authentication in mobile devices , Computers & Security 59, 2016, pp 210-235.

[27] ”Android and iOS Squeeze the Competition, Swelling to 96.3% of the Smartphone Operating System Market for Both 4Q14 and CY14, According to IDC”, www.idc.com, 2016. [Online]. Available: `http://www.idc.com/getdoc.jsp?containerId=prUS25450615`. [Accessed: 23-May- 2016].

[28] C. Shen, Y. Zhang, X. Guan and R. Maxion, ”Performance Analysis of Touch-Interaction Behaviour for Active Smartphone Authentication”, IEEE Trans.Inform.Forensic Secur., vol. 11, no. 3, pp. 498-513, 2016.

[29] R. Meier, ”Face Detection in Google Play services — Android Developers Blog”, Android-developers.blogspot.sg, 2015. [Online]. Available:`http://android-developers.blogspot.sg/2015/08/face-detection-in-google-play` [Accessed: 23- May- 2016].

[30] N. Kwak, ”Feature extraction for classification problems and its application to face recognition”, Pattern Recognition, vol. 41, no. 5, pp. 1701-1717, 2008.

[31] V.Vaidehi, S. Vasuhi, Person Authentication using Face Recognition , Proceedings of the world congress on engg and computer science, 2008.

[32] ”2 Billion Consumers Worldwide to Get Smart(phones) by 2016”, eMarketer, 2014[Online].Available: `http://www.emarketer.com/Article/2-Billion-Consumers-Worldwide-Smar` 05-June-2016]

[33] H. Khan,A.Atwater, U. Henqartner, ”Itus: An Implicit Authentication Framework for Android”, Proceedings of the 20th Annual International Conference on Mobile Computing and Networking, MobiCom 14. ACM, New York, NY, USA, pp. 50718, 2014.

[34] D. Buschek,A. De Luca, F. Alt,”Improving Accuracy,Applicability and Usability of Keystroke Biometrics on Mobile Touch screen Devices,” *Proc. of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI15,2015.

[35] R. Gaines, W. Lisowski, N. Shapiro and J. Press, ”Authentication by keystroke timing: some preliminary results (No. R-2526-NSF)”, 1980.

[36] N. Zirjawi, Z. Kurtanovic and W. Walid, ”A Survey about User Requirements for Biometric Authentication on Smartphones”, in Evolving Security and Privacy Requirements Engineering (ESPRE), 2015 IEEE 2nd Workshop on, 2015.

[37] H. Crawford, K. Renaud and T. Storer, ”A framework for continuous, transparent mobile device authentication”, Computers & Security, vol. 39, pp. 127-136, 2013.

[38] ”Biometrics”, Wikipedia, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Biometrics. [Accessed: 05- Sep- 2016].

[39] A. K. Jain, A. Ross, and S. Prabhakar, ”An Introduction to Biometric Recognition,” in IEEE Transactions on Circuits and Systems for Video Technology, vol. 14(1), January 2004, pp.4 20.

[40] U. Bakshi and R. Singhal, ”A Survey on Face Detection Methods and Feature Extraction Techniques of Face Recognition”, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), vol. 3, no. 3, 2014.

[41] ”The 2 Big Problems with Fingerprint Security”, Yahoo.com, 2015. [Online]. Available: https://www.yahoo.com/tech/the-2-big-problems-with-fingerprint-security-109371608679.html. [Accessed: 21- Sep- 2016].

[42] Y. Meng, D. Wong, R. Schlegel and L. Kwok, ”Touch gestures based biometric authentication scheme for touchscreen mobile phones”, M. Kutyowski, M. Yung (Eds). Information security and cryptology. Berlin: Springer, pp. 331-350, 2013.

[43] N. Alotaibi, E. Bruno, M. Coakley, A. Gazarov, V. Monaco, S. Winard, F. Witkowsk, A. Copeland, P. Nebauer, C. Keene and J. Williams, ”Text Input Biometric System Design for Handheld Devices”, in Proceedings of Student-Faculty Research Day, Pace University, 2016, pp. B7.18.

[44] H. Saevanee, P. Bhatarakosol, H. Saevanee, and P. Bhatarakosol, ”User Authentication using Combination of Behavioral Biometrics over the Touchpad acting like Touch screen of Mobile Device,” in Proc. ICCEE, 2008, pp.82-86.

[45] N. Zheng, K. Bai, H. Huang and H. Wang, ”You Are How You Touch: User Verification on Smartphones via Tapping Behaviors,” 2014 IEEE 22nd International Conference on Network Protocols, Raleigh, NC, 2014, pp. 221-232.

[46] M. Wajeeh and S. M. Hameed, ”User Authentication Based on Touch Dynamics of Pattern Unlock”, in IJCSMC, 2015, pp. 622 634.

[47] M. Antal and L. Szab, ”Biometric Authentication Based on Touchscreen Swipe Patterns”, Procedia Technology, vol. 22, pp. 862-869, 2016.

[48] ”Local Representation of Facial Features (Face Image Modeling and Representation) (Face Recognition) Part 1”, What-when-how.com, 2016. [Online]. Available: http://what-when-how.com/face-recognition/local-representation-of-facial-features-face-image-modeling-and-representation-face-recognition-part-1/. [Accessed: 25- Sep- 2016].

[49] ”Artifical neural network”, Wikipedia, 2016, [Online]. Available: https://em.wikipedia.org/wiki//Artificial_neural_network. [Accessed: 25- Sep- 2016].

[50] ”Sensors Overview Android Developers”, Developer.android.com, 2016. [Online]. Available: https://developer.android.com/guide/topics/sensors/sensors_overview.html. [Accessed: 29- Sep- 2016].

[51] J. Angulo and E. Wästlund, ”Exploring Touch-screen Biometrics for User Identification on Smart Phones,” IFIP Advances in Information and Communication Technology, Vol. 375, pp. 130-143, 2012.

[52] Study Guide, ”Z Scores (Z Value) & Z Table & Z Transformations”, Sixsigmastudyguide.com, 2014. [Online]. Available: http://sixsigmastudyguide.com/z-scores-z-table-z-transformations/. [Accessed: 29- Sep- 2016].

[53] [Online]. Available: http://www.sjsu.edu/faculty/gerstman/StatPrimer/t-table.pdf. [Accessed: 29- Sep- 2016].

[54] ”Degrees of Freedom”, Onlinestatbook.com, 2016. [Online]. Available: http://onlinestatbook.com/2/estimation/df.html. [Accessed: 18- Oct- 2016].

[55] M. D. Petty, ”Advanced Topics in Calculating and Using Confidence Intervals for Model Validation”, Proceedings of the Spring 2013 Simulation Interoperability Workshop, San Diego CA, April 8-12 2013, pp. 194-204.

[56] M. D. Petty, Calculating and Using Confidence Intervals for Model Validation, Proceedings of the Fall 2012 Simulation Interoperability Workshop, Orlando FL, September 10-14 2012, pp. 37-45.

[57] J. Brownlee, ”An Introduction to Feature Selection - Machine Learning Mastery”, Machine Learning Mastery, 2014. [Online]. Available: http://machinelearningmastery.com/an-introduction-to-feature-selection/. [Accessed: 21- Oct- 2016].

[58] A. Karegowda, A. Manjunath and M. Jayaram, ”Comparative Study of Attribute Selection Using Gain Ratio and Correlation Based Feature Selection”, International Journal of Information Technology and Knowledge Management, vol. 2, no. 2, pp. 271-277, 2010.

[59] ”T-Score vs. Z-Score: Whats the Difference?”, Statistics How To, 2017. [Online]. Available: http://www.statisticshowto.com/when-to-use-a-t-score-vs-z-score/. [Accessed: 09-Feb- 2017].