A Static Authentication Framework Based On Mouse Gesture Dynamics

by

Bassam Sayed

B.Sc., Helwan University, 2003

A Dissertation Submitted in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

A Static Authentication Framework Based On Mouse Gesture Dynamics

by

Bassam Sayed

B.Sc., Helwan University, 2003

Supervisory Committee

Dr. Issa Traore, Supervisor

(Department of Electrical and Computer Engineering)

Dr. Fayez Gebali, Committe Member

(Department of Electrical and Computer Engineering)

Dr. Kui Wu, Outside Member

(Department of Computer Science)

Supervisory Committee

Dr. Issa Traore, Supervisor

(Department of Electrical and Computer Engineering)

Dr. Fayez Gebali, Committe Member

(Department of Electrical and Computer Engineering)

Dr. Kui Wu, Outside Member

(Department of Computer Science)

ABSTRACT

Mouse dynamics biometrics is a behavioural biometrics technology which consists of the movement characteristics of the mouse input device when a computer user is interacting with a graphical user interface. However, existing studies on mouse dynamics analysis have targeted primainely continuous authentication or user re-authentication for which promising results have been achieved. Static authentication using mouse dynamics appear to face some challenges because of the limited amount of data that could reasonably be captured during such process. We present, in this thesis, a new mouse dynamics analysis framework that uses mouse gesture dynamics for static authentication. The captured gestures are analyzed using LVQ neural network classifier. We conducted an experimental evaluation of our framework involving 41 users, achieving FAR = 1.55% and FRR = 2% when four gestures are combined.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

It is a pleasure and honour to thank the many people who made this thesis possible:

It is difficult to express my gratefulness to my supervisor, Dr. Issa Traore. If it were not for his inspiration, and his overwhelming patience that I would be able to finish my thesis. Throughout the period of time I dealt with him, he provided encouragement, good teaching, and even advice for my personal life. I definitely would have been lost without him.

I would like to thank the many people who participated in my experiment which is a main component of the thesis work. In particular all my colleagues at Zeugma Systems Inc. and the Faculty of Engineering at University of Victoria.

I also would like to thank very close friends and colleagues of mine; Akif Nazar, Sherif Sad, Yousry Abdel-hamid, and Soltan Alharbi. As they always encouraged and helped me at times of distress.

I am mostly grateful to my mother, Ayda Fahmy. She raised me, supported me, guided me, and loved me.

Lastly, I wish to thank all my family and friends, especially my wife, Amany Abdelhalim. She supported me, and encouraged me. We passed a lot of hard times together.

I almost forgot, I would like to thank my two sons for making so much noise and causing so much trouble while I was writing this thesis!!

DEDICATION

To my mom Ayda Fahmy and my wife Amany Abdelhalim whom I dedicate this work.

Chapter 1

Introduction

1.1 Context

In the last two decades there have been a steady reliance on the usage of computerized systems in our day-to-day life. Those computerized systems are ever getting more and more networked with relatively high speed networks, in order to make our life easier and even more entertaining. With the emergence of computerized services like online banking and trading and many others, the number of hacking incidents and identity theft have been rising rapidly. The US governments Computer Emergency Response Team reported about 39,000 cases of corporate hacking in 2002, more than 40,000 cases in 2003 and over 62,000 in 2004, and needless to mention that those are just the reported cases [2]. One of the different reasons why the number of hacking incidents is increasing so dramatically is that existing authentication systems are not strong enough to stop intruders from breaking into the system. As a result, new methods are being developed to harden user access as well as to protect the confidentiality and integrity of important data in various computer systems.

The main objective of any authentication system is to protect the resources, which

range from computer systems that contain confidential data to the networks that connect the computer systems themselves. The word authentication comes from the Greek word "authentes"; which means the act of establishing or confirming that someone or something is authentic. Generally, authentication systems achieve their objective through different factors which can be categorized as follows. Something the user has like a security token, or an identity card. Something the user knows like a password; a pass phrase, or a personal identification number (PIN). Something the user is, like his fingerprints or retinal patterns. Biometric recognition systems which fall in the last category are by far one of the strongest authentication approaches available. The word biometrics is defined as "the measurement and recording of the physical or behavioral characteristics of an individual for use in subsequent personal identification" [3]. In the field of information technology, biometrics is defined as "the technologies that measure and analyze human body or behavioral characteristics, such as fingerprints, eye retinas and irises, voice patterns, facial patterns and hand measurements, for authentication purposes" [4]. Despite the wide usage of biometric technology for physical security, the adoption of biometrics in day-to-day use of computer systems has been slow. The main reason of this limited usage of biometrics is the reliance on special hardware devices for biometric data collection. Although some computer vendors have started integrating the needed hardware in their products, the vast majority of machines currently available lack such special hardware devices. This limits the scope of where the biometric technology can be used as it will only be available for the organizations that can buy the required additional hardware. This also applies for the individuals that use their computer systems at home or for their daily activities. It might be hard to convince these individuals to pay extra money for the extra security they will gain by using the special biometric hardware. This is especially valid if the user of the computer system is using it for

regular day-to-day usage like online purchases or for paying bills.

A new category of biometrics that is gaining in popularity is referred to in the literature as behaviometrics for behavioral biometrics, where the analysis focuses on studying user behavior while he is interacting with a computing system for the purpose of identification. One interesting example of behaviometrics is mouse dynamics biometrics. Mouse dynamics biometrics is a new technology which has been proposed initially and extensively studied in our lab for computer user recognition [5, 6]. Prior works on mouse dynamics have focused on improving the design of graphical user interfaces [7, 8]. Mouse dynamics has not been seen as a potential measure for computer security until recently. The work reported in [6] is the first contribution on using mouse dynamics for biometric identification problems. The biometric identification problem is approached by extracting the behavioral features related to the mouse movements and analyzing them to enhance the security of the computer systems. The developed mouse dynamics biometric technology involves a signature, which is unique for every individual. This signature is computed based on selected mouse movement characteristics. Statistical methods are used to compute these characteristics in the feature extraction phase. Later on, the extracted features are fed to a neural network for the recognition phase. The main strength of the mouse dynamics biometric technology lays in its ability to continuously monitor the legitimate as well as illegitimate users throughout their usage of a computer system. This is referred to as continuous authentication. Continuous authentication or identity confirmation based on mouse dynamics is very useful for continuous monitoring applications such as intrusion detection systems and if used properly it can be applied for digital forensics analysis. The initial work was validated through an experiment conducted in 2003 that involved 22 human participants. This experiment achieved an equal error rate of 2.46% for the false rejection and false acceptance rate [5, 6]. A follow up

experiment involving 26 new users conducted in 2007 confirmed the previous results.

1.2 Research Problem

Unlike traditional biometric systems mouse dynamics biometric technology may face some challenges when applied for static authentication, which consists of checking user identity at login time. The key reason for this potential lackness is the data capture process, which requires more time to collect sufficient amount of mouse movements for user identity verification. The main goal of this research is to address these challenges. More specifically, we study the feasibility of mouse dynamics analysis for static authentication by developing a new framework allowing performing the authentication in a short period of time. We use mouse gestures to achieve our goal. In the enrollment phase the user draws a set of gestures several times in-order to record his behavior while drawing these gestures on a computer monitor. We extract the features and analyze them and then we train a neural network which later will be used for identification. In the test phase the user will be asked to replicate a subset of the gestures drawn by him in the enrollment phase to test against his stored signature.

1.3 General Approach

Following a typical biometric analysis process, our proposed approach consists of two main phases, the enrollment phase and the test phase illustrated by Figures 1.1 and 1.2, respectively. In the enrollment phase, we capture raw data and analyze it to extract the features which form the signature of the different users. Then we use these features to train a neural network. The neural network will be used in the test phase to identify the user or verify his identity.

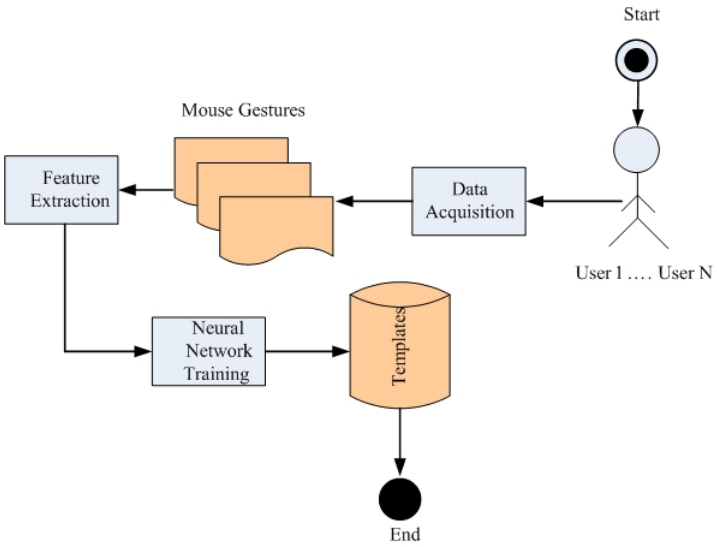


Figure 1.1: Enrollment Phase

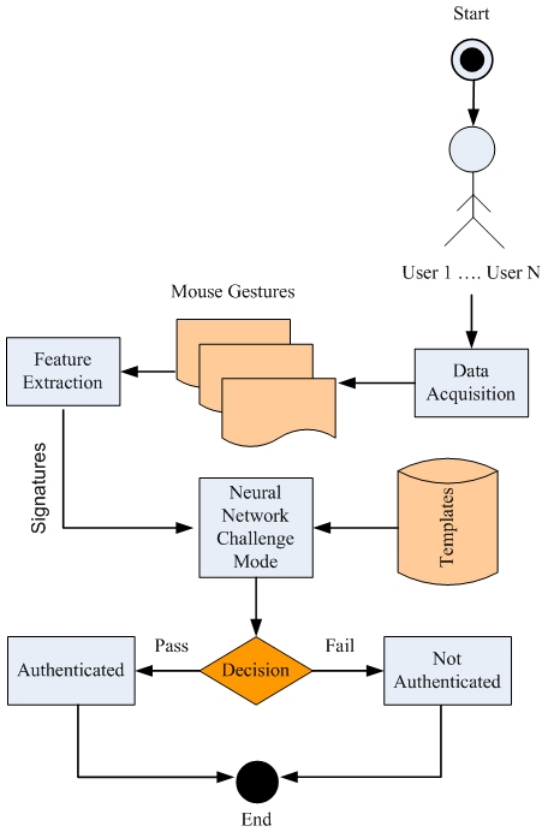


Figure 1.2: Test Phase

Generally the enrollment phase consists of three steps. The first step is the raw mouse dynamics data capture. In this step the user draws the selected gestures and the raw mouse dynamics data get tagged and stored with her/his credentials. In the second stage the features get extracted from the raw mouse data and get combined to form the profile. In the third stage the profile is used to train a neural network. The neural network design is similar for all the users which allows us to store only the state of the neural network for each user as her/his reference signature. The challenge here lays in the signature and the neural network components. On one side, extracting features that form distinct signature for each user is a challenge. On the other side, the challenge would be how to design a neural network that when trained it would be capable of distinguishing between the users.

The test phase would be the actual act of authentication in which a user who is claiming an identity will be asked to replicate a number of the gestures he already sketched in the enrollment phase. The neural network will be used to perform the recognition process by loading the saved neural network state from the profile of the claimed identity and processing the current data to decide whether the authentication pass or fail.

1.4 Contributions

The main contribution of this research is the development of a new biometric analysis technique allowing static authentication based on mouse gestures. The proposed system can be used as replacement or reinforcement for existing legacy textual password based authentication systems and can be used as single or multifactor authentication scheme for e-commerce applications. In addition, the proposed technique tried to overcome some of the limitations of the hand-written signature verification sys-

tems, such as the usage of special hardware and the difficulty of estimating the false acceptance rate (FAR) [9].

1.5 Thesis Outline

The rest of the thesis is structured as follows:

- In Chapter 2, we give a brief overview of biometrics systems and discuss their main characteristics.
- In Chapter 3, we summarize and discuss related work on gesture analysis and mouse dynamics and motivate our research work.
- Chapter 4 discusses the main contribution of this thesis. It illustrates the overall design of the biometric analysis framework developed throughout our research. This includes the data capturing, user enrollment, feature extraction and training of the neural network and the detection model.
- In Chapter 5, we describe the experiment that we have conducted, evaluate the proposed framework and discuss corresponding results.
- In Chapter 6, we conclude by summarizing the results of the research and discussing future work that will be conducted for further enhancements.

Chapter 2

A Brief Introduction to Biometrics

In this chapter we give an overview of biometric systems and discuss their design issues and performance metrics.

2.1 Overview

The word biometric is derived from two Greek words "bio" which means life and metric which means "to measure". The idea of identifying humans based on their distinguishing physiological characteristics, dates back to the ancient times [10, 11]. At the time of the ancient Egypt the workers who were building the great pyramids were not only identified by their names but also with some distinctive features they have, such as height, eye colour, and scars. The Pharaohs themselves were authenticating decrees by adding their thumbprint to the papyrus papers along with their signatures [10, 11]. In the recent years biometric technologies gained a lot of momentum as they started being used pervasively, such as in passports. The biometric passport looks like the regular passport except it holds a tiny computer chip. The computer chip holds biometric information about the owner of the passport like fingerprints and face image along with the regular information like the name and date of

birth [12]. We can define the biometrics in its modern form as the study of methods for uniquely identifying humans based upon one or more intrinsic physiological or behavioral traits.

2.2 Categories of Biometric Techniques

Biometrics techniques were usually grouped into two main categories [4, 13], namely behavioral and physiological biometrics, however recently a third category named Soft Biometrics [14] has emerged as shown in Figure 2.1.

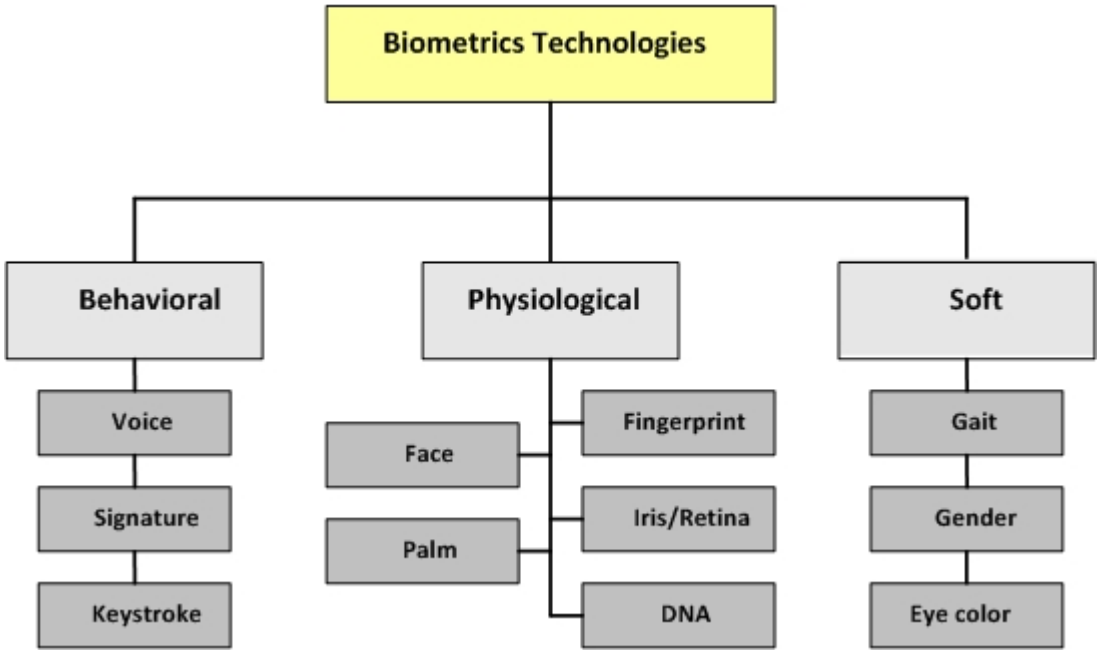


Figure 2.1: Biometric Technologies

- **Physiological Biometrics:** establish a person identity based upon one or more physical characteristics of the human body such as fingerprints, face, iris, retina, palm, vessel structure, and DNA codes.
- **Behavioral Biometrics:** establish a person identity based upon his behaviour or actions. Behavioral biometrics are all about the "how": how a person signs,

how a person talks, or how he/she types on the keyboard. Voice signature, handwritten signature, and keystroke dynamics are all examples of behavioral biometrics. As mentioned earlier, behaviometrics is a new subcategory of behavioral biometrics which capture and analyze human computer interactions. Examples of such biometrics include mouse and keystroke dynamics.

- **Soft Biometrics:** usually cover physiological characteristics that provide some information about a person but lack the distinctiveness and permanence to sufficiently differentiate any two individuals. Typically a soft biometric technique is combined with other biometric methods to improve its performance, but can not be used as a standalone biometric solution. Gait, gender, eye color, and ethnicity are examples of soft biometrics [14].

2.3 Biometric Systems Architecture

Generally any biometric system involves a combination of hardware and software components. The hardware components are responsible for live capturing of biometric data. Usually the hardware components consist of sensors or capturing devices that record the biometric data in a raw format which will be analyzed later by the software components. Typically the cost of any biometric system depends heavily on the price of the hardware components. The software components are responsible for managing the biometric data and performing a pattern matching process to authenticate users.

Typically, biometric systems implement a generic model to check users identity. The generic biometric process involves two main phases. These phases are common in their goals and general procedure, but different from one technology to another in their implementation and technical details. A brief summary for each phase is presented in the remaining of this subsection.

2.3.1 Enrollment and Signature creation Phase

Biometrics systems typically do not compare the recorded biometric traits directly to the current sample. Instead they create signatures or templates for comparison. Before any biometric system starts identifying individuals, trustworthy samples or biometric traits must be collected and processed so that the signatures and templates can be constructed and stored for later usage; this process is known as the enrollment phase. The data collected in the enrollment phase is a key aspect to any biometric analysis process. The data itself must or at least should be both distinctive between individuals and repeatable over time for the same person. The quality of the collected biometric samples greatly affects the overall accuracy and performance of the biometric system.

At the end of the enrollment phase, the collected biometric samples are used to create the biometric signature or template. In this process the biometric system analyzes the enrollment samples to extract biometrical patterns that contain unique, distinctive, and stable features and ignore any noisy and non-useful data. The extracted biometrical patterns form the templates or the signatures of the users that will serve as references to be used in the authentication procedure. Technically these signatures or templates are a result of a pattern learning process which is usually based on artificial intelligence or machine learning techniques.

2.3.2 Matching and Test Phase

The matching and test phase operates in one of two separate modes, the verification mode and the identification mode. In the former mode, the user claims an identity and provides a live sample. The system will process the live sample and compare it to the stored signature or template of the claimed identity, if it is a match the user is accepted, otherwise the user is rejected. Whereas in the later mode, the identity

of the user that provided the live sample is not known in advance. The biometric system will compare the extracted patterns from the live sample to all the templates or signatures that exist in the database of the enrolled users resulting in a match or no match.

2.4 Biometrics Quality Challenges

Researchers face many challenges while developing biometric systems. In [15], these challenges are pointed out in four main categories:

1. **Accuracy:** Biometrics systems like any pattern recognition system are imperfect. Unlike authentication systems based on password or challenge/response questions, biometrics systems do not have a perfect match. There are three reasons underlying the imperfect accuracy of the biometric systems as outlined in [15].

    - ***Information limitation:*** means that the biometric samples do not have enough distinguishing and distinctive information content to discriminate the individuals effectively.
    - ***Representation limitation:*** Practical biometric systems should have a feature extraction technique that extracts a representation scheme which retains all the discriminatory information in the sensed measurements. This is not always as accurate as expected.
    - ***Invariance limitation:*** the ideal biometric matcher given the representation scheme should minimize the discrepancy within the same class (inter-class) and maximize the variation among the different classes (intra-class).

2. **Scale:** How does the number of the identities in the enrollment database affect the speed and accuracy of the biometric systems is considered another challenge. Nowadays, biometric systems have become so involved in our life that it may be possible to have hundreds of thousands or even millions of individuals in one database. This will not affect verification systems since essentially they perform a one-to-one match, however in the identification systems, performing a one-to-one match for the N individuals in the database is time-consuming and the time will increase linearly with the number of the records in the database. Typical methods attempted to solve this problem include, the use of multiple or faster hardware and the use of exogenous data (e.g. gender, age, geographical location) supplied by human operators.

3. **Security:** Another challenge is the integrity of the biometric systems. Making sure that the input biometric sample was offered by its legitimate owner and that the system indeed matched the input pattern with genuinely enrolled pattern sample form the two sides of the biometric systems integrity currency.

4. **Privacy:** There is a fundamental contradiction between privacy and biometrics, at least from the point of view of some individuals that enrol in any biometric system. Usually the users have concerns regarding their biometric data. For example, will the biometric data be used in a different area other than the intended one? E.g. will the fingerprints provided for access control be matched against others in a criminal database? Obviously, some strategies need to be implemented to solve this fundamental privacy problem.

2.5 Biometric Systems Performance

Biometric systems are not perfect match systems and hence can not recognize an individual with absolute certainty. In fact, the decision process is based on a probabilistic match between the live sample and a stored biometric template in the database. Most of the biometric systems can be evaluated using the following measures [13]:

- **False Acceptance Rate (FAR) or False Match Rate (FMR):** the expected probability of an erroneous conclusion by the biometric system that a biometrical signature stored in the database is from the same person that has just presented a live sample, when in fact, it is not.
- **False Rejection Rate (FRR) or False Non-Match Rate (FNMR):** the expected probability of an erroneous conclusion by the biometric system that a biometrical signature stored in the database is not from the same person that has just presented a live sample, when in fact, it is.
- **Failure to Acquire (FTA):** the expected probability of transactions for which the biometric system is unable to capture the biometrical pattern with sufficient quality for matching purpose.
- **Failure to Enroll (FTE):** the expected probability of the population of users that were unable to enroll their biometrical measurements into the system in order to create a template of sufficient quality.

Performance metrics can be illustrated and analyzed using the following graphs:

- **Receiver Operating Characteristic (ROC) Curve:** In general, the ROC curve is a plot of the false acceptance rate on the x-axis against the corresponding rate of correctly accepting genuine users plotted on the y-axis.

- **Detection of Error Trade-off (DET) Curve:** DET serves as another mean of plotting the results of the biometric systems. It is a modified version of the ROC curve that plots the false acceptance rate (FAR) on the x-axis and false rejection rate (FRR) on the y-axis.

Chapter 3

Related Work

In this chapter, we survey and summerize related work on mouse dynamics and gesture analysis. By the end of the chapter, we discuss how corresponding literature relate to our work.

3.1 Mouse Dynamics in Human-Computer Interaction Studies

Towards the end of the 20th century the Human-Computer Interaction (HCI) field became increasingly essential as the computers became increasingly inexpensive, small, and powerful. The field of HCI focuses on the understanding of the interactions between the people and computers. The interactions happen at the user interface or simply the interface, which usually consists of both software and hardware components. Since the turn of the millennium, the computer mouse is the main input device in the graphical user interface (GUI) environments. Earlier works on mouse dynamics analysis have focused essentially on user interface design improvement issues. Fitts' law is one of the key results obtained from these prior works on mouse dynamics.

Fitts' law by far is one of the most successful and well-studied laws that model the act of pointing, both in the real world as in drawing on papers with a pen or in the computer world when using a mouse or light pen. Fitts' law was first introduced by Paul Fitts in 1954 [16, 17]. Fitts' law models both the point-and-click and drag-and-drop actions for a mouse in the computer world. It basically models the speed and accuracy tradeoffs in rapid, aimed movements. The Fitts' law has been formulated in different forms but the most common one is the Shannon formulation proposed by Scott Mackenzie [18] as follows.

$$T = a + b \log_2(\frac{D}{W} + 1)$$

Where:

- T is the average time taken to complete the movement.
- a and$b$ are empirically determined constants, that are device dependent.
- D is the distance from the starting point to the center of the target.
- W is the width of the target, which corresponds to the "accuracy", which normally falls between $\pm\frac{W}{2}$.

In the above expression $\log_2(\frac{D}{W} + 1)$ is referred to as the index of difficulty (ID). Figure 3.1 demonstrates an example of the Fitts' law on a computer monitor with a mouse cursor.

Fitts' law states that bigger targets can be reached faster than smaller targets when the distance is constant, and close targets can be reached faster than far targets when the width of the target is constant. Fitts' law was even successful in predicting the movement times for assembly line work. Nevertheless, Fitts' law has it own disadvantages. The main disadvantage of the Fitts' law is the inherited one-dimensionality

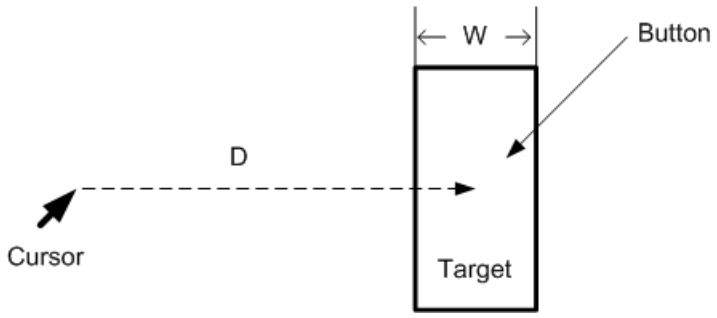


Figure 3.1: Illustration of Fitts' Law.

of the formula. Fitts' original experiments tested human performance in drawing horizontal movements towards a target. Both the direction of the movement and the width of the target area were measured along the same axis as demonstrated in Figure 3.1. For this reason, and based on the fact that computer monitors are 2D displays, Mackenzie extended the Fitts' law to overcome this limitation and modified the formula to deal with 2D tasks. Mainly, he adjusted the index of difficulty part of the formula to consider the height of the target along its width. Also he took the angle of approach into consideration as illustrated by Figure 3.2 [18, 19].

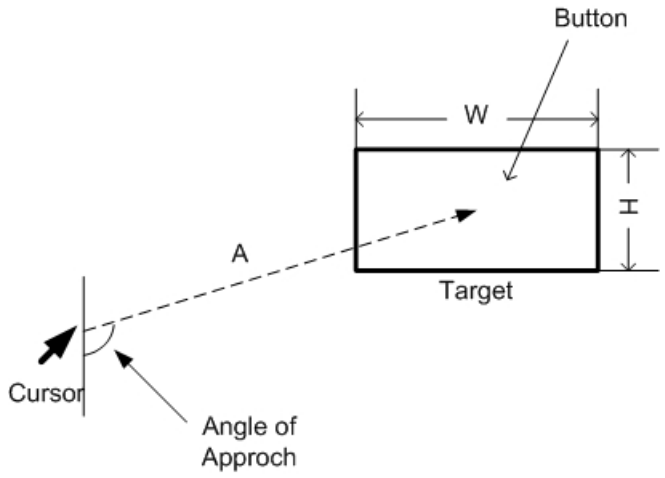


Figure 3.2: Illustration of Mackenzie's modification to the Fitts' Law.

Later, Oel *et al.* [7] proved that the formulas presented by Fitts and some of its derivatives like Mackenzie's formula will not be so accurate if the target areas

are relatively small. They performed an experiment that involved 32 experienced computer users. In the experiment they asked the subjects to move the computer mouse as quickly as possible to click on a target area. They counted not only the successful trials but also the missed ones. Later on, they showed that the predicted production time for the small area targets, e.g. radio buttons, and check boxes, in the GUI were not that accurate. They analyzed experimental data from the past work done by other researchers as well as their own data. Finally, they defined a new formula which is a power model that contains a logarithmic model within its exponent that fits the data curve better than the original Fitts' law and the mentioned derived formulas. They also proved that Fitts' law is an approximation of their formula [7]. The final power law they proposed is given by:

$$MT = (a.w^b).A^{c+d.log_2(W)}$$

Where:

- W is the width of the target area.
- A is the amplitude or the distance from the start point to the center of the target area.
- a, b, c, and d are empirically determined values when fitting the data curve.

Whisenand *et al.* [8] also conducted an experiment that involved 32 experienced computer users. In their experiment they showed that Fitts' law and the derived formulas were not that accurate in predicting the movement time *(MT)* and that the variance of the predicted time ranged from 44% to 97%. They proved that the point-and-click task was more accurately predicted than the drag-and-drop. Also the angle of approach was an important factor in the experiment. They showed that

approaching target areas in a diagonal form were inaccurately predicted compared to approaching them in a vertical or horizontal form. The conclusion of their work, rather than being a new formula to predict the movement time accurately, was a set of recommendations for the design of user interfaces. They claim that these recommendations can improve user experience when interacting with the GUI. For example, they recommended using square targets when possible, and sizing their width between 8mm to 16mm.

3.2 Mouse Dynamics as a Behavioral Biometrics

As reported above, a lot of attention has been given to the use of the computer mouse as an input device in the human computer interaction field. But not until recently, mouse dynamics emerged as a behavioral biometric technology. As a matter of fact, our research group is one of the pioneers in that field and a comprehensive research has been done establishing and validating the biometrics characteristics of mouse dynamics.

More specifically, Ahmed and Traore in [5, 20, 6] established that the actions recorded for a specific user while interacting with a graphical user interface is intrinsic to that user. These actions are recorded passively and validated throughout the session [5]. The outcome of that research can be used in the intrusion detection field as well as access control as proposed by the authors. They defined and studied seven different features that model the biometric characteristics of mouse dynamics. They grouped the seven features into five categories to form the signature of each user as follows [5]:

1. Movement Speed: Movement speed compared to traveled distance factor.

2. Movement Direction: covers average movement speed per movement direction

and movement direction histogram factors.

3. Action Type: covers point and click, double click, and mouse move factors.

4. Travelled Distance: Traveled distance histogram.

5. Elapsed time: Movement elapsed time histogram.

In the user enrollment mode they used a feed-forward multilayer perceptron neural network to learn the user behaviour based on the mouse signature. The status of the trained network then gets stored in a signature database. In the detection mode, the stored status of the trained neural network is loaded and the current session data is applied to the neural network to output what is referred to by the researchers as the confidence ratio (CR). The confidence ratio is a percentage number that represents the degree of likeness of the two behaviours being compared.

In the experiment they conducted to validate their model, they collected data from 22 participants. Then they used a one-hold-out cross validation test to compute the performance of the proposed system. They reached FAR of 2.4649% and FRR of 2.4614% when they adjusted the threshold value of the confidence ratio to the point of 50% which is the crossover point in the ROC curve [5]. These results were later confirmed by increasing the overall number of participants to 48 users [21]. The interesting outcome of this research is that the mouse dynamics can be successfully used as a behavioral biometric. Although, the work accomplished in this research can be used both for static and dynamic authentication systems, the primary focus of the study was initially on continuous authentication application which requires the user to be logged into the system to start the monitoring. Static authentication will require designing a special purpose GUI and asking the user to perform predefined actions to login, and could present some challenges related to the length of the time required to capture enough data for user recognition.

Gamboa *et al.* in [22], performed a similar research and showed that the mouse dynamics collected while interacting with a graphical user interface is intrinsic to each user. They basically conducted an experiment to capture the user interaction via a pointing device while playing a memory game. The pointing device was a computer mouse and the memory game was developed as a java applet running in a web browser. They asked 50 volunteers to participate in their experiment. They collected the interaction data and then analyzed them to extract the features. They grouped the features into two sets (i) Spatial features and (ii) Temporal features; in total they extracted 63 features. Next they used a sequential forward selection technique that is based on the greedy algorithm to select the best single feature and then add one feature at a time to the feature vector. Each time a feature is selected, the feature vector is fed to a classifier that minimizes the equal error rate of the system. The algorithm stopped when the equal error rate was not decreasing. The sequential classifier would accept or reject the claimed identity when the probability distribution of the user was greater than a limit λ that was adjusted to operate at the crossover point, corresponding to the equal error rate (EER).

They showed that the EER progressively tends to zero as more strokes are recorded. Additionally the number of features in the feature vector was different for each user, ranging from one to eleven features. This means that the more interaction data the system records, the more accurate the system should be. Also not all the features are needed in order to classify the users. But as we commented before it might be difficult to use such a method for static authentication at login time since the authors reported that the memory game took from 10 to 15 minutes on average to get completed.

Pusara and Brodley in [23] proposed an approach for user re-authentication based on the data captured from the mouse device. Their hypothesis was based on the

possibility to model the user behaviour through his mouse movements. They implemented a system that continuously monitor the user-invoked mouse events and movements, and raised an alarm when the user behaviour deviated from the learned normal behaviour. They organized all the possible mouse movements and events in a hierarchy. Based on that hierarchy the features were extracted. In their approach they used one profile for the learning process, this profile was considered as the normal profile and any other behaviour that deviates sufficiently is considered abnormal or anomalous. The down side of this approach, is that they assume only one user is using the computer system. They used decision tree classifier in the decision process. They conducted an experiment that included 11 users and they reached false positive rate of 0.43% and false negative rate of 1.75%. The authors clearly mentioned that their method would fail if the user did not utilize the mouse or did not generate enough mouse movements and events.

3.3 Modeling Stroke Gesture Performance

A lot of attention has been paid to improving the performance of gesture recognition. However, little research was focused on modeling the human performance in producing these gestures. Modeling human performance in producing gestures would help in advancing the design and evaluation of the existing and future gesture based user-interfaces. As we mentioned before, Fitts' law and its derivatives were successful in modeling the human performance in visually guided tasks but they are inappropriate to model the freehand open-loop stroke gestures. In [1], Cao and Zhai tried to construct a fairly accurate computational model that can predict the production time of single pen-stroke gestures as a function of its composition. They based their research on the previous work of Isokoski [24] and Viviani *et al.* [25]. Isokoski based

his assumptions on the fact that any gesture can be approximated to a certain number of straight line segments. The best correlation result between the predicted and actual production time was $R^2 = 0.85$ on Uni-stroke gestures and between 0.5 and 0.8 on other types of gestures [37]. The main advantage of Isokoski's work was the simplicity and ease of application. However, the difficulty of defining the number of straight lines needed to approximate the gestures was the main drawback of his work [1]. On the other hand Viviani *et al.*, studied the human drawing behaviour at a lower motor control level. They proposed a formula that models the instantaneous tangential velocity as a function of curvature. The formula, which was named Viviani's power law of curvature, is defined as

$$V = KR^{\beta}$$

Where:

- V is the instantaneous tangential velocity.
- R is the radius of the curvature.
- K and β are constants of the model.

Simply, Viviani's power law of curvature states that the larger the curvature the trajectory has at a given point, the slower the motion will be at that point.

Cao and Zhai [1] based their model on the assumption that any gesture can be broken down into several basic elements or components that can be represented with a lower-level model. This is somewhat similar to Isokoski's model. However, they did not approximate the gestures into straight lines but they took into consideration the curves and the corners. They put together the basic components into three elements as follows:

1. **Smooth Curve (Arc):** The production time T of the curve is defined as follows:

$$T(curve) = \frac{\alpha}{K} r^{1-\beta}$$

Where:

- r is the radius of the arc.
- α is the sweep angle.
- K and β are constants.

2. **Straight Line:** They proposed more than one model to compute the production time of the straight lines and later they proved that the power model $T(line) = mL^n$ is the most valid one according to their experiments. This model suggests that humans tend to move faster with longer lines and hence the power-like relationship between the production time and the length L. In this formula, m and n are empirically determined constants.

3. **Corner:** can be defined as the sudden change of stroke direction with respect to the arms that form it. They stated that it is difficult to define the operational boundaries of the corners. So they formulated a tentative representation of the production time of the corner as the relation between the net contribution of the sudden change in direction to the total production time of the corner. That is defined as $T(corner) = f(\theta)$ where f is a function of the corner angle θ empirically determined by the experiments.

Finally, they break down any gesture into these three basic elements as illustrated by Figure 3.3 and they compute the total production time by the summation of the production time of the basic elements as follows:

$$T(gesture) = \sum T(line) + \sum T(curve) + \sum T(corner)$$

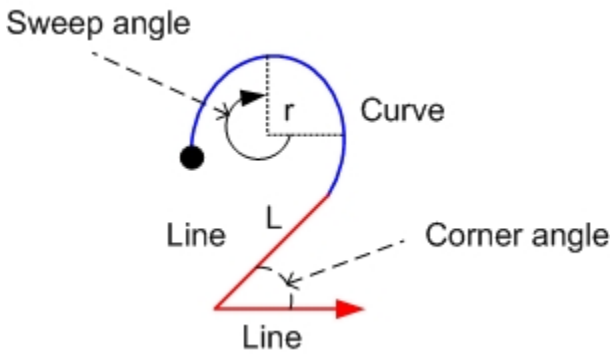


Figure 3.3: Gesture decomposition into basic elements (from [1]).

In their experiment they grouped the gestures into five categories. The first three categories were examples of the basic models described before; however, the last two gesture categories namely Polyline and Arbitrary were used to test their summative model. The Arabic number two in Figure 3.3 is an example of the arbitrary gesture, while polyline is number of lines connected with corners. They reached high correlation level of $R^2 = 0.9$ or higher for all the different gesture sets, which proves that their theoretical formulas were fairly accurate in modeling the human performance in drawing stroke gestures using a light pen. The only down side from our point of view is that they did not mention if these formulas will remain accurate if the input device was different from a pen (stylus). In other words, what would be the results of the experiment if the input device was a computer mouse?

3.4 Authentication Based on Gestures, Shapes and Strokes

Mayer *et al.,* in [26], explored the usage of graphical passwords as an alternative to text-based passwords. They based their research on the fact that humans tend to remember graphical objects better than words. In addition, while there are roughly 2×10^{14} eight characters passwords consisting of upper case, lower case, and digits, it is not that hard to find the password in a crafted dictionary of words. As a matter of fact they referred to a study that involved 14,000 UNIX passwords and in which almost 25% of the passwords were found in a carefully formed dictionary. The authors proposed two schemes of forming a password; the first is a text-based password with graphical assistance and the other is a password which is completely graphical. They refer to the second scheme as Draw-a-Secret or DAS. It is also important to mention that the authors in their research targeted the hand-held personal digital assistance (PDA) devices that have the stylus as their main input method. In the draw-a-secret method they designed an interface consisting of a rectangular grid of size $G \times G$. The user is asked to draw a shape on this grid. For each cell the user crosses while drawing the shape, the corresponding coordinate get stored with a special coordinate indicating the "pen up" event. The password is defined by the coordinates sequence and length of the drawn strokes. The user is required to input the same drawing in the same sequence and length in-order for the password to get accepted.

The authors showed mathematically that the graphical password space is even bigger than the textual password one. Also they explored the memorability of the graphical passwords and showed that their DAS scheme is easy to memorize, especially when drawing simple shapes or objects. On the other hand, the authors assumed that the user would shield the screen from onlookers when drawing the password, which

actually might not be practical all the time, especially when spy cameras are being utilized. In other words, if an intruder was able to see the drawing or the shape of a password, he might be able to replicate it. Hence biometrics comes into handy.

Bromme and Al-Zubi [27], proposed a new authentication method based on a multifactor biometric sketch. In the proposed method, the final decision was based on more than one factor. The key factor was the actual sketch drawn by the user, which is composed of a set of deformable shapes. The secondary factor, which was added for increasing the reliability of the system, is user's knowledge of how to fulfil a specific sketching task. These tasks were negotiated with the users in the enrollment phase by the authentication system and were considered as secrets. They conducted an experiment that included 10 users, who were asked to draw some predefined shapes on a tablet computer with a digital pen. The experiment included two tests. In the first test, they emphasized the statistical part of the recognition system by asking all the participants in the experiment to draw the same PIN number 0123. The recognition error rate ranged from 25.7% for only one digit to 3.9% when the four digits were combined. In the second test, they asked the participants to draw a sketch composed of a specific type/number of shapes combined together. They did not depict the way the shapes should be combined and they left it open to the imagination of the participants. Then, they conducted three different types of imposter tests. In the first test, the imposter has full knowledge of the sketch. In the second test , the imposter has partial knowledge about the sketch and in the third test, the imposter has no knowledge of the sketch he is trying to forge. The equal error rates ranged from 7.25% to 0% when the imposter has no knowledge of the sketch. The interesting result of the conducted research is that it is hard to duplicate the structural information of a signature when no knowledge is available about it. In addition, increasing the number of objects in the behaviometric signature results in higher accuracy.

In [28], Bella and Palmer developed a system that investigate whether pianists can be identified based on the dynamics of their finger movements while in music performance. In their experiment, four skilled pianists memorized and performed identical melodies. They used motion capture system to record the finger movements and the melodies were performed on a digital piano. The movement data of the fingers were relative to the piano keyboard in the vertical plane. In addition, the timing of piano key movements was also recorded. They used functional data analysis technique to analyze the movement velocity and acceleration, and to build two curves, one before the key-press and one after the key-press. They reached 87% correct pianists identification using the "before key-press" curve and 84% with the "after key-press" curve.

Hayashi *et al.* in [29] proposed a user identification scheme using computer mouse. The main goal of their experimental results was to prove that mouse can be used for identification. They conducted two experiments in the first of which, the users draw a circle between concentric circles on a computer monitor. The users were allowed to draw varying shapes of circles as long as they lay between the concentric circles shown on the screen. In the second experiment, the users were allowed to drew any figure within the concentric circles used in the first experiment. In the two experiments they captured and stored the mouse coordinates, time in millisecond, and the distance between the center of the drawn circle or shape to the center of the concentric circles. They stored all the captured data in a database that was used later in the identification phase. They define a formula to calculate the match rate between the sample data in the test phase to the stored data in the database which is compared to a threshold for user identification. They achieved a performance of FRR = 15% and FAR = 7% for the first experiment. In the second experiment they reached FRR = 13% and FAR = 0%.

Syukri *et al.* in [30] commented on the work done in [29] and proposed a new technique that utilizes a more complex figure objects than the ones proposed in [29]. They used signatures drawn by a mouse as the complex figure objects in their new technique. The proposed technique used the same match rate formula proposed in [29] and added extra steps and extracted more features in order to achieve better results. They conducted two experiments in the first of which, they used a static database and in the second they used a dynamically updated database. They proved that the results of the dynamically updated database is better than the static one. They achieved FRR = 9% and FAR = 8% for the static database, and FRR = 7% and FAR = 4% for the dynamic database. It is important to mention that neither reference [29] nor reference [30] provide any indication about the number of participants in their experiment, which makes it hard to judge the significance of the obtained results.

3.5 Hand-written Signature Verification Systems

Dynamic hand-written signature verification systems (HSV) typically requires a light pen that is combined with a graphical tablet or a touch enabled device. Plamondon and Lorette in [31], showed that there is a great variability in signatures according to country, age, time, habits, psychological or mental state. Therefore, it is hard to build a database of signatures that represent the real-world. It is clear that some of the mentioned factors would affect our framework in addition to any behavioral biometric system however, some of them should not affect our system since it is based on mouse gestures. For instance, the country factor should not affect our proposed system as gestures can be any drawing from any language or can be drawing that has no meaning and is not tied to a specific language. They also noted that people were not always happy to have their signatures stored in test databases used by other

people to practice forging them which is not the case in our system. In addition, Brault and Plamondon, in [32], noted that some signatures tend to be simple enough and could be easily forged, while others may be quite complex. By carefully choosing the gestures which are complex enough, we can overcome such a limitation. Also the final decision of our system could depend on the result of multiple gestures not just one gesture.

In [9], Gupta and McCabe outlined, based on a review of HSV systems, that it is very difficult to estimate the false acceptance rate (FAR) of the hand-written signature verification systems since actual forgeries are impossible to obtain. Hence, applying skilled or random forgeries is the only method that the performance evaluation of such systems could rely on. Applying random or what is sometimes referred to as zero-effort forgeries usually results in low FAR, because either the forger did not have an information about the signature he was trying to forge or the system randomly selects other signatures to compare against the current signature. On the other hand, applying skilled forgeries by allowing a skilled forger to practice the signature first, might not be practical at all the times. In our experiment, we asked the participants to draw the same gestures, which allowed us to use cross-validation technique to estimate the FAR of our proposed framework.

3.6 Discussion

The main objective of our proposed research is to develop an effective authentication system using mouse gesture dynamics by exploiting the underlying biometrics information. To our knowledge our system is the first of such kind in the literature. As we have shown previously, mouse dynamics has been extensivly studied in the HCI field to improve the user interface design, and also has been studied as a be-

havioral biometric technique. Mouse dynamics was successfully used for continuous authentication as demonstrated in the literature review. However, we do think that it is not straight forward to apply the same mouse dynamics techniques for static authentication. In the meanwhile, many researchers have proposed other techniques for replacing the legacy static authentication methods. For instance, as discussed in the above literature review, researchers have considered graphical or sketch-based techniques as possible alternative or reinforcement for conventional passwords. This was simply based on the fact that humans tend to remember shapes more easily compared to textual passwords and it is very hard to guess the password if it was a graphical one. Other researchers considered the hand-written signatures as another possible replacement however, as shown previously, HSV systems impose their own issues and require a special hardware devices.

In this work, we have decided to use the mouse gesture dynamics to combine the advantages of the graphical passwords (which are gestures in our case) with the behavioral mouse dynamics to propose a framework that can perform the authentication in a short period of time and at the same time avoid some of the issues that has been outlined in the above litreature review.

Chapter 4

Gesture Analysis and Detection Technique

In this chapter, we present our detection and analysis methods for the proposed behavioural biometric system based on mouse gesture dynamics. The system generally requires any typical pointing device. In our experiment, we used the computer mouse which is the traditional pointing device for any general purpose computing system.

4.1 Pilot Experiment and System Design

In the early stages of our research, we conducted a pilot experiment that involved six users. The main purpose of the experiment was to explore the feasibility of our assumption. Which is, whether it is possible to differentiate between individuals based on their behavioural biometrics while drawing mouse gestures. The participants in the pilot experiment were asked to replicate eight different types of gestures by drawing each gesture 20 times. The same eight gestures were all similar for all the participants and the only requirement was to draw them in one stroke. We collected the raw data from the drawing area in the form of the hor-

izontal coordinate (X-axis), vertical coordinate (Y-axis) and the absolute time in millisecond at each pixel. Each gesture replication for a given gesture can be defined as a sequence of data points and each data point can be represented by a triple $< x, y, t >$ consisting of its X-coordinate, Y-coordinate, and absolute time respectively. The j^{th} replication of a gesture G can be represented as a sequence $G_j = \{< x_{1j}, y_{1j}, t_{1j} >, < x_{2j}, y_{2j}, t_{2j} >, \cdots < x_{nj}, y_{nj}, t_{nj} >\}$ where n is referred to as the size of the drawn gesture and each $< x_{ij}, y_{ij}, t_{ij} >$ ($where\ 1 \leqslant i \leqslant n$) is a data point. Figure 4.1 illustrates an example of a drawn gesture.

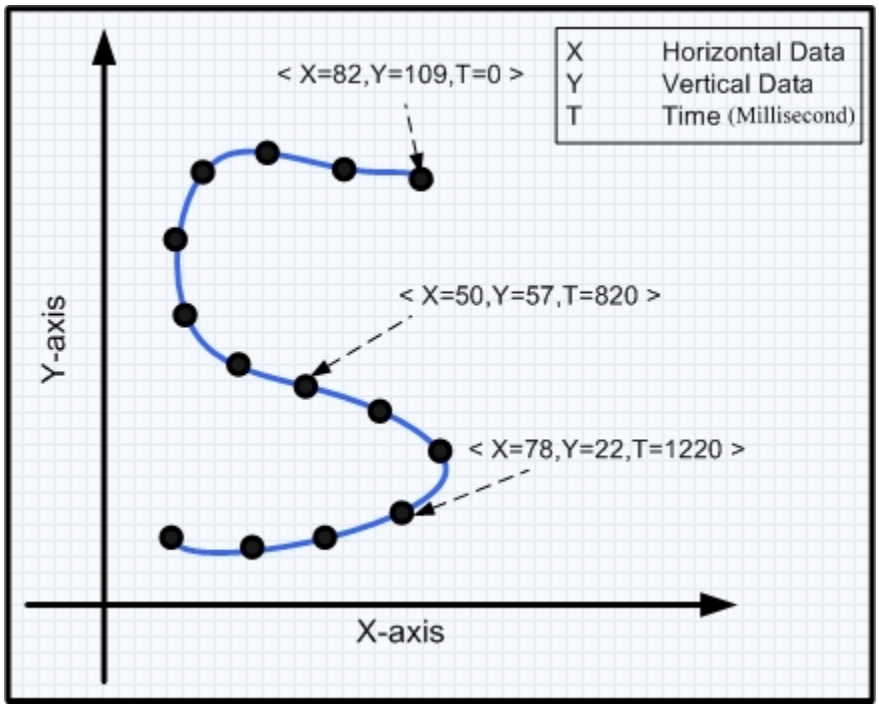


Figure 4.1: Example of a drawn gesture involving n=14 data points.

Based on the pilot experiment we observed the following:

- The average gesture size drawn in one stroke was 64 data points.
- Some participants started to get used to the experiment and started to draw the gesture in a faster way, which is a departure from their normal behaviour.

- The raw data had some noise, like repetitive data points or data points with the same time stamp which must be filtered.
- Although the users were told to be as consistent as they can while drawing the gestures, as expected variability in shape and size were clearly a major observation.

Based on the data collected in the pilot study, we were able to design our gesture data acquisition and analysis framework. Our framework, depicted by Figure 4.2, consists of four modules:

1. Gesture Creation Module.
2. Data Acquisition and Preparation Module.
3. Feature Extraction Module.
4. Classification Module.

We describe in more details each of these modules in subsequent sections.

4.2 Gesture Creation

The gesture creation module as illustrated in Figure 4.3 is a simple drawing application used to ask the participant to freely draw a pre-defined set of gestures. The main purpose of this module is to make the participant draw the gestures in his own way in-order to replicate them later on. It is important to note here that the gestures themselves are not tied to any language and they do not have necessarily a meaning. They can be any drawing that can be produced in a uni-stroke. Also, the gesture creation module serves as a practice step for the participants to get familiar with the idea of drawing mouse gestures.

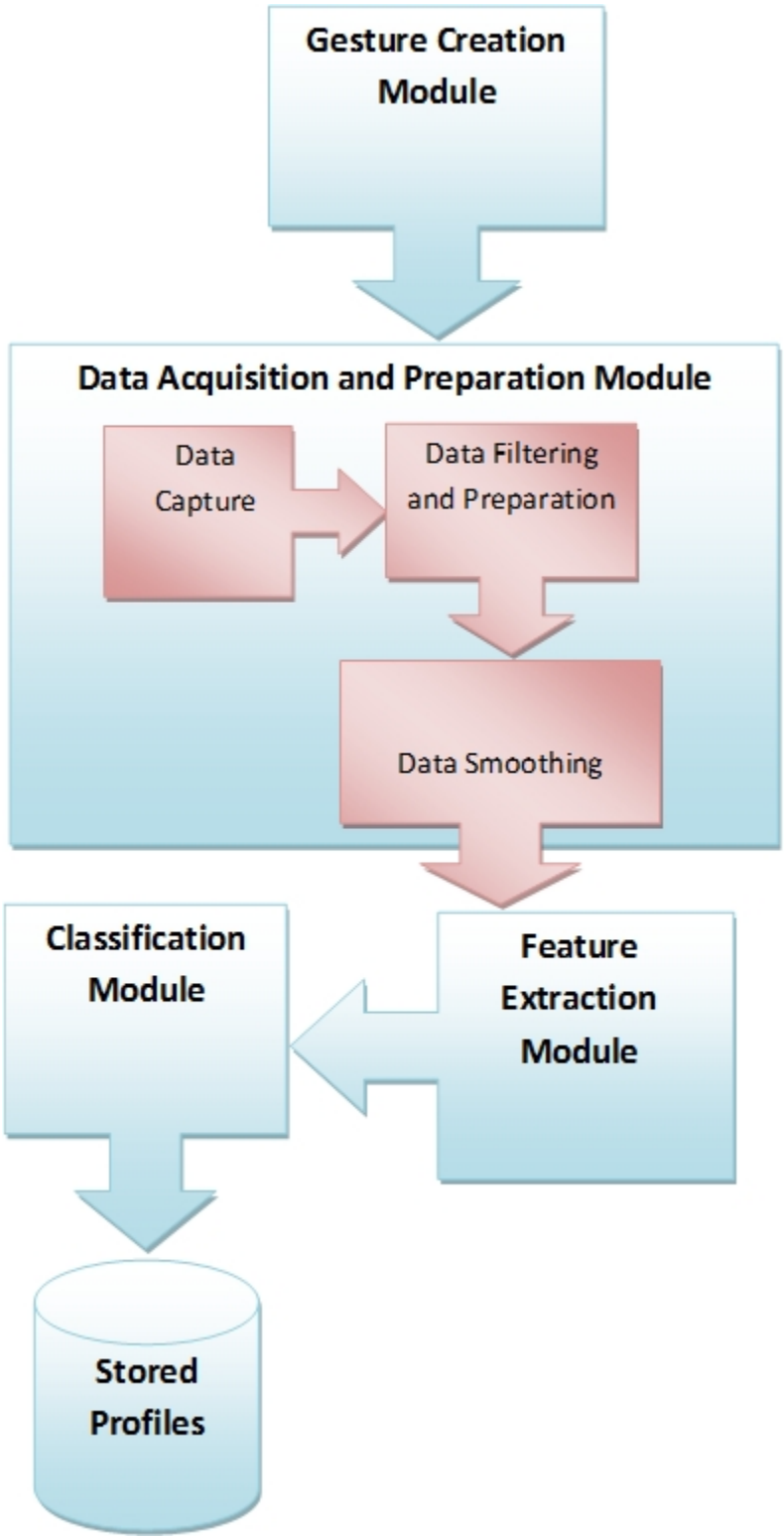


Figure 4.2: Gesture detection and analysis framework architecture.

The gesture creation module assists the user in two different ways. Firstly, it normalizes the input to the center of the drawing area. This is achieved by computing the centroid of the X-coordinate Cen_x and Y-coordinate Cen_y of the gesture in both the X-axis and Y-axis of the gesture data and then subtracting the centroid Cen_x and Cen_y from each data point to position the drawing about the center of the drawing area. This is computed by the following formulas:

$$Cen_x = \frac{\sum_{i=1}^{n} x_i}{n}, \forall\ x_i, i = 1 \ldots n \Rightarrow x_i' = x_i - Cen_x \tag{4.1}$$

$$Cen_y = \frac{\sum_{i=1}^{n} y_i}{n}, \forall\ y_i, i = 1 \ldots n \Rightarrow y_i' = y_i - Cen_y \tag{4.2}$$

where n is the number of data points in the gesture

Although this shifting of the drawn gesture is done, the data get stored without saving these changes. The main usage of the center normalization is for the comparison step that is explained later.

Secondly, the module normalizes the gesture spacing to achieve size of 64 data points. The 64 data points were based on the pilot experiment that we did in the early stages of our research. As mentioned earlier, we were able to determine the average size of drawing the pre-defined set of gestures in one stroke. Figure 4.3 illustrates the outcome of the gesture normalization by the gesture creation tool.

4.3 Data Acquisition and Preparation

The data acquisition and preparation module involves three main components, namely data acquisition, data preparation, and data smoothing.

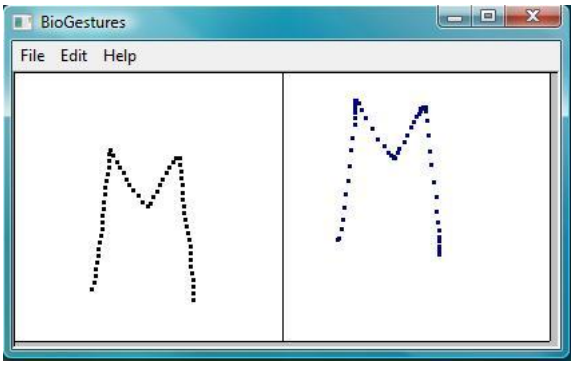
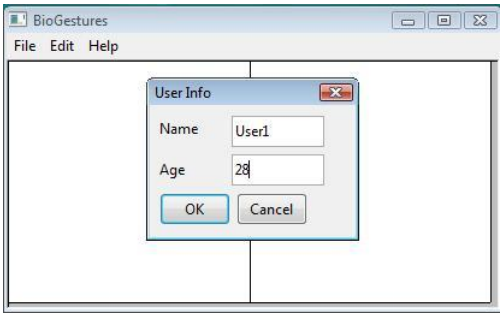


Figure 4.3: Example gesture normalization achieved by the gesture creation tool: before normalization (right) and after normalization (left).

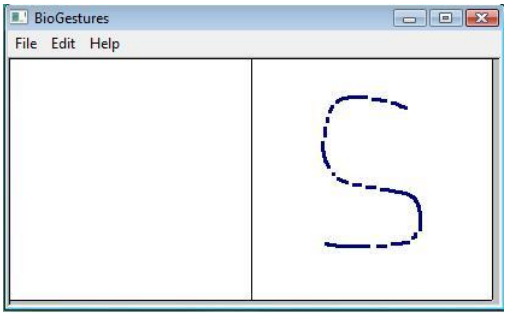
4.3.1 Data Acquisition

The data acquisition component loads the gestures created initially by the user using the gesture creation module and present them to the user to replicate. The data acquisition module records the user interaction while drawing the gesture. The module basically records the horizontal coordinates denoted by x_{ij}, vertical coordinates denoted by y_{ij} and the absolute time in milliseconds starting from the origin of the gesture t_{ij}, where n is the gesture size, $1 \leq i \leq n$, and j is the gesture replication number. Figure 4.4 illustrates data acquisition during the user enrollment process. As shown in Figure 4.4(a), before enrolling, the user must input his personal information. For each user the program creates a directory which will contain the user replications for the different gestures. Figure 4.4(b) depicts a sample gesture which looks very similar to Latin letter S that the user must replicate for specific number of times (e.g. 20 times). During the enrollment, each replica of a gesture provided by the user is compared to the original gesture template and rejected in case where there is a substantial difference between them in shape. For instance, in Figure 4.4(c) an example of rejected user input for current gesture is shown. A visual feedback is given to the user by mirroring his input in red color. Figure 4.4(d) shows an example of accepted input and visual feedback is given by mirroring the user input in green

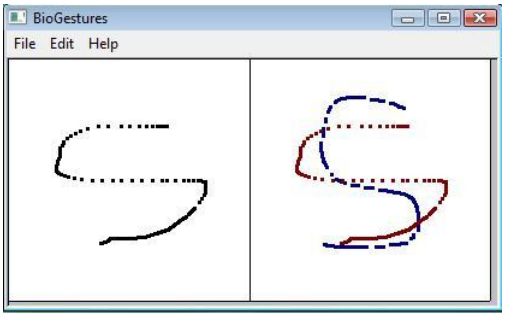
colour. The user has to wait three seconds between each successful replication. The idea behind this waiting time is to prevent the user from drawing the gesture too fast. Actually the module asks the user to release the mouse between each successful replication during the wait time. The main reason we implemented such wait time and mouse release, is based on one of the observations made in the pilot experiment. We assume that the wait time and mouse release will force the users to maintain their normal behaviour each time they replicate the gesture.



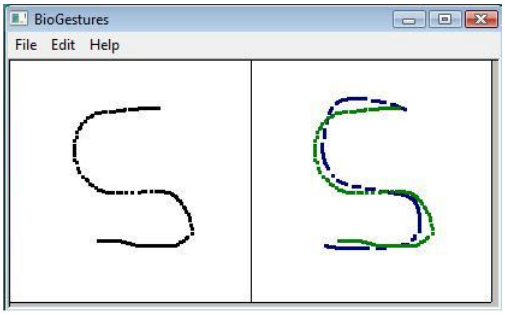
(a) The user inputs his name and age.



(b) The Module loads the S letter gesture template. The user is expected to replicate the gesture in the left area.



(c) Example of rejected replication from the user.



(d) Example of accepted replication from the user.

Figure 4.4: User Enrollment Process and Tool

The data acquisition module compares the input from the user to the example gesture using relatively simple comparison formula to determine if the input from the user is close to the example or not. The main purpose of such comparison is to circumvent miss-drawn gestures and to provide visual feedback to the user about the input as illustrated in Figures 4.4(c) and 4.4(d). The comparison formula measures

the angle between the vector representation of the input gesture and the example one, with respect to the X-axis and Y-axis coordinate data . Given $G_1 = \{< x_{11}, y_{11} >, < x_{21}, y_{21} >, \cdots < x_{n1}, y_{n1} >\}$, the drawn gesture, and $G_2 = \{< x_{12}, y_{12} >, < x_{22}, y_{22} > , \cdots < x_{n2}, y_{n2} >\}$, the example (template) gesture, where n is the gesture size.We use the following formula to compute the angle between the two vectors:

$$\cos\theta = \frac{u.v}{\sqrt{u^2} * \sqrt{v^2}} = \frac{u.v}{\sqrt{u^2 * v^2}} \tag{4.3}$$

Where $u = G_1, v = G_2$ *and* $u.v$ *is the dot product of the two vectors.*

We found that we can use 0.8 as a threshold for the minimum accepted input, which is good enough to verify that the drawn gesture is close to the example one. The 0.8 threshold value corresponds to 36 degrees which allows the two vector representations of the gestures to be 36 degrees apart or less. Based on our pilot study having this threshold value gives some degree of freedom as humans can not replicate a drawing in the same exact way it was drawn originally.

4.3.2 Data Preprocessing

The data acquisition module pre-processes the collected raw data from the computer mouse in such a way that some noise patterns are ignored or dropped. This has to happen since the data resulted from the pointing devices is usually jagged and the input devices produce irregular data. This kind of pre-processing will filter data resulting from two common problems of the pointing devices: data points generated with the same timestamp where $t_i = t_{i+1}$ and redundant data points where $(x_i, y_i) = (x_{i+1}, y_{i+1})$.

After preprocessing the raw data, the data acquisition module normalizes the input data in two different ways. The first is center normalization and the second

is space normalization. Both types of normalization implement the same formula applied by the gesture creation module. However, the space normalization of the data acquisition module applies the formula to a portion of the gesture data. We have to normalize the spacing so that the final size of the gesture is equal to the size of the template gesture in order to compare the two gestures as explained later. The gesture drawn by the user is divided into four segments and we only apply the spacing normalization on the last segment of the gesture. The main reason for performing the spacing normalization on the last segment, is that we want to avoid changing the user data as much as possible. Figure 4.5 illustrates the spacing normalization of the last segment of the gesture. Note how the data points in the last segment have changed.

Figure 4.5: Gesture normalization can happen by either adding or removing data points to the last segment of the gesture.

4.3.3 Raw Data Smoothing

Data Smoothing is generally used to eliminate noise and extract real patterns from data. In our framework, we use smoothing to smooth the data among the different replications obtained for each user. Generally humans can not draw the same gesture

with the same exact detail twice under the same conditions. This will result in some variability in the replicas produced by the same individual for the same gesture. Data smoothing allows us to smooth such variability and minimize its effect on the learning process. We use the robust version of the standard weighted least squares regression (WLSR) method to smooth the data. The robust version of such a method gives a zero weight to the data points which are far away from the overall mean of the data. In other words, it eliminates the negative effect of the outliers on the smoothing process. The MATLAB program implementation assigns a zero weight to the outlier data points which are far from the mean of the data with the distance of absolute six means. For clarity, Figure 4.6 illustrates an example for applying weighted least square regression method on sample data belonging to one of the users in our pilot study. Figure 4.6 illustrates how the data of the origin of 20 replica of a given gesture are being smoothed.

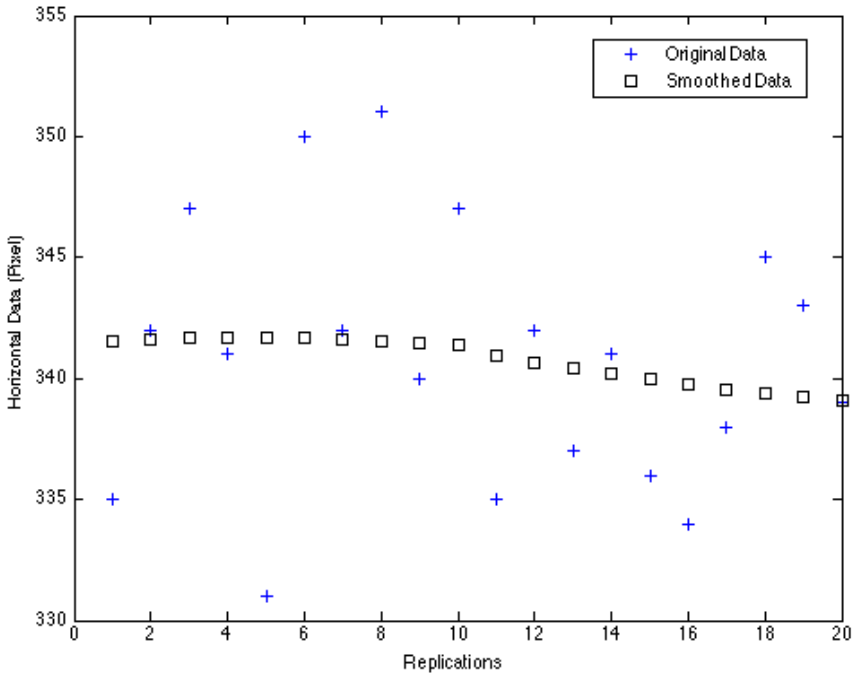


Figure 4.6: Example of data smoothing using weighted least square regression method.

We apply the smoothing step only to the horizontal (X-axis) and vertical (Y-axis) coordinates data excluding time. We construct a vector that aggregates the same occurrence of the first data point from each of the different replications. Then we apply WLSR method to fit the data in the vector to produce smoothed data. We repeat the process for each of the remaining data points of the gesture. Figure 4.7 illustrates the result of applying smoothing to 20 replica of Arabic numeric "five" gesture belonging to the same individual.

Algorithm 1 summarizes our smoothing process and assumes the following:

1. Let m be the number of replications.
2. Let n be the size of the gesture.
3. Let $p_{ij} = (x_{ij}, y_{ij})$ be a data point, where $1 \leq j \leq m$, $1 \leq i \leq n$.
4. Given a gesture G, we denote by G_j the j^{th} replica: $G_j = (p_{1j}, p_{2j}, \ldots, p_{nj})$
5. Let P_i denote a vector containing the i^{th} data point from each of the different replications, where $i = 1, 2 \ldots n : P_i = (p_{i1}, p_{i2}, \ldots, p_{im})$.

Algorithm 1 SMOOTH($VG \leftarrow \{G_1, G_2 \ldots G_m\}, n, m$)

Require: Integers ($n > 1$) and ($m \geqslant 1$).
Ensure: The value of $V'G \leftarrow \{G'_1, G'_2 \ldots G'_m\}$ smoothed data.

1: $TV \leftarrow \varnothing$ {Temporary vector}
2: **for** $i \leftarrow 1$ to n **do**
3: $\quad P'_i \leftarrow WLSR(P_i)$
4: $\quad TV \leftarrow TV \bigcup \{P'_i\}$
5: **end for**
6: $V'G \leftarrow \{TV\}^T$ {Transpose the TV}
7: **return** $V'G$

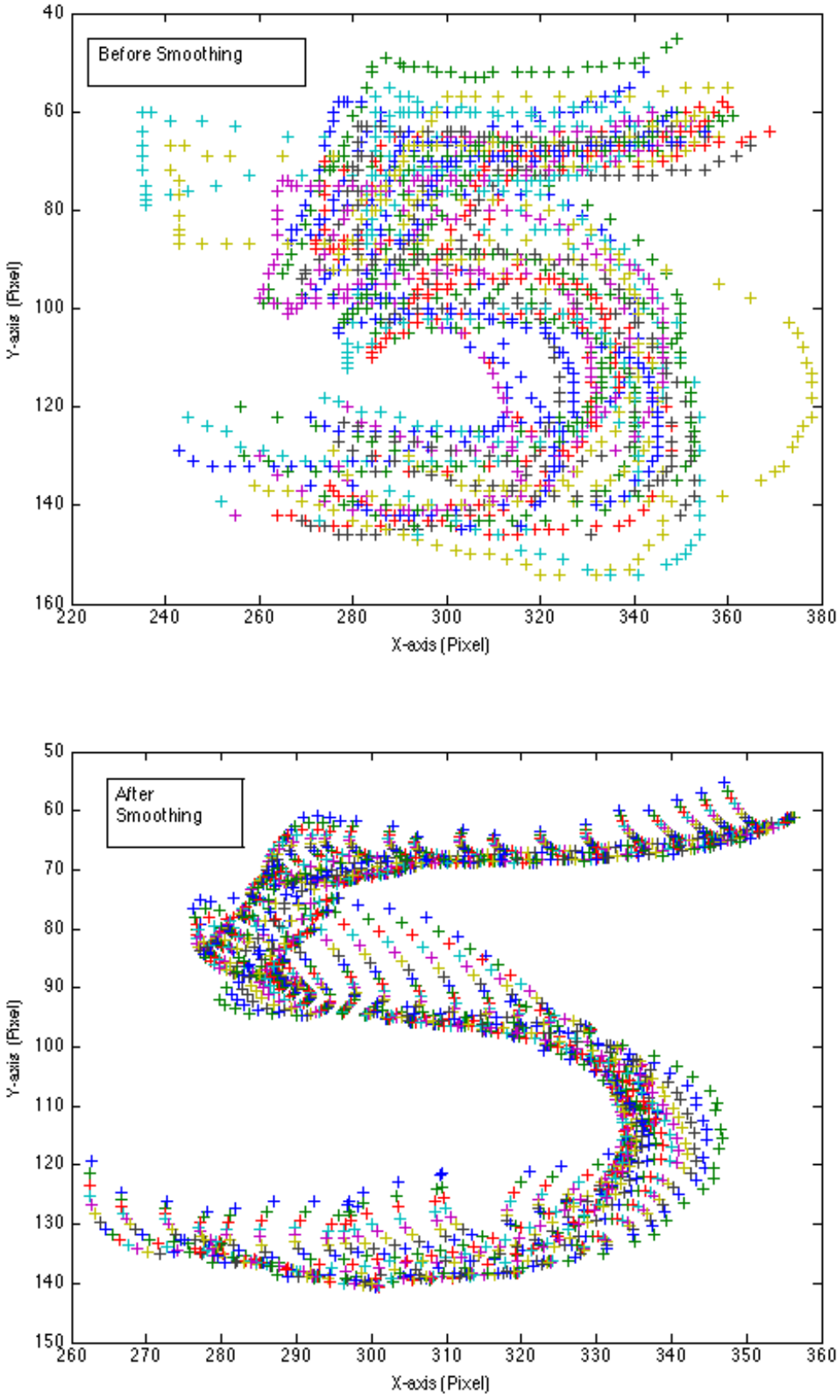


Figure 4.7: Smoothing 20 Replications for Arabic Numerical Five Gesture.

4.4 Feature Extraction

Based on the pilot study, the output of the data acquisition module by itself is not enough to form a unique signature for each user. The feature extraction module extracts 12 features from the raw data. The complete list of the extracted features is provided in Table 4.4.

Feature Description	Notation	Definition
Horizontal Coordinate	x	$X - axis\ data$
Vertical Coordinate	y	$Y - axis\ data$
Absolute Time	t	-
Horizontal Velocity	hv	$v_{hor} = \frac{\Delta x}{\Delta t}$
Vertical Velocity	vv	$v_{ver} = \frac{\Delta y}{\Delta t}$
Tangential Velocity	tv	$v = \sqrt{v_{hor}^2 + v_{ver}^2}$
Tangential Acceleration	ta	$v' = \frac{\Delta v}{\Delta t}$
Tangential Jerk	tj	$v'' = \frac{\Delta v'}{\Delta t}$
Path from the origin in pixels	l	-
Slope angle of the tangent	θ_i	$\theta_i = \arctan(\frac{y_i}{x_i})$
Curvature	c	$c = \frac{\Delta \theta}{\Delta l}$
Curvature rate of change	δc	$\delta c = \frac{\Delta c}{\Delta l}$

Table 4.1: Extracted features from raw data.

Figure 4.8 illustrates the angle of the tangent with the X-axis and the length of the path from the origin to the data point. Both measures are used to compute the curvature and its rate of change with respect to the length from the origin. Figure 4.9 and 4.10 compares four features extracted from two replications from User 1 to one replication from user 2 for the same gesture (M gesture), User 1 and User 2 are two arbitrary users from our pilot study. Note how the extracted features from the two replications of User 1 are close to each other compared to the features extracted from User 2.

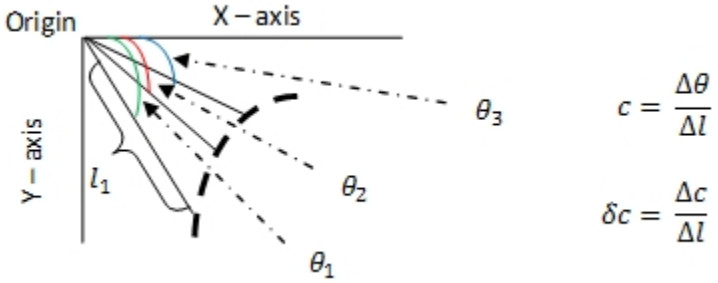


Figure 4.8: Angle of curvature and its rate of change for a portion of a drawn gesture.

4.5 Classification Techniques

Throughout our research we applied different classification techniques in order to determine an effective classifier. In our experiment we used one statistical classifier namely the principal component analysis and two classifiers based on neural networks. In the process of designing the neural network classifiers we followed two different approaches, one based on monolithic design and the other based on modular design. As we will explain later the modular neural network design proves to be the most suitable for our problem in terms of classification performance.

4.5.1 Principal Component Analysis Technique

Our choice of principal component analysis (PCA) technique [33] was based on its successful usage in areas like face-recognition and pattern classification in high dimensional data. In our case, we have $F = 12$ features as outlined in Table 4.4, the dimensionality of each feature is $GS = 64$, where GS is the gesture size. We applied PCA on our data assuming the total dimensionality equals to $F \times GS$. We applied the standard PCA steps outlined in [34]. Although the PCA was successful in producing the eigenvectors from our data, using the eigenvectors to classify the users had a poor performance. The performance was as low as 63% success rate with only three users in the test population.

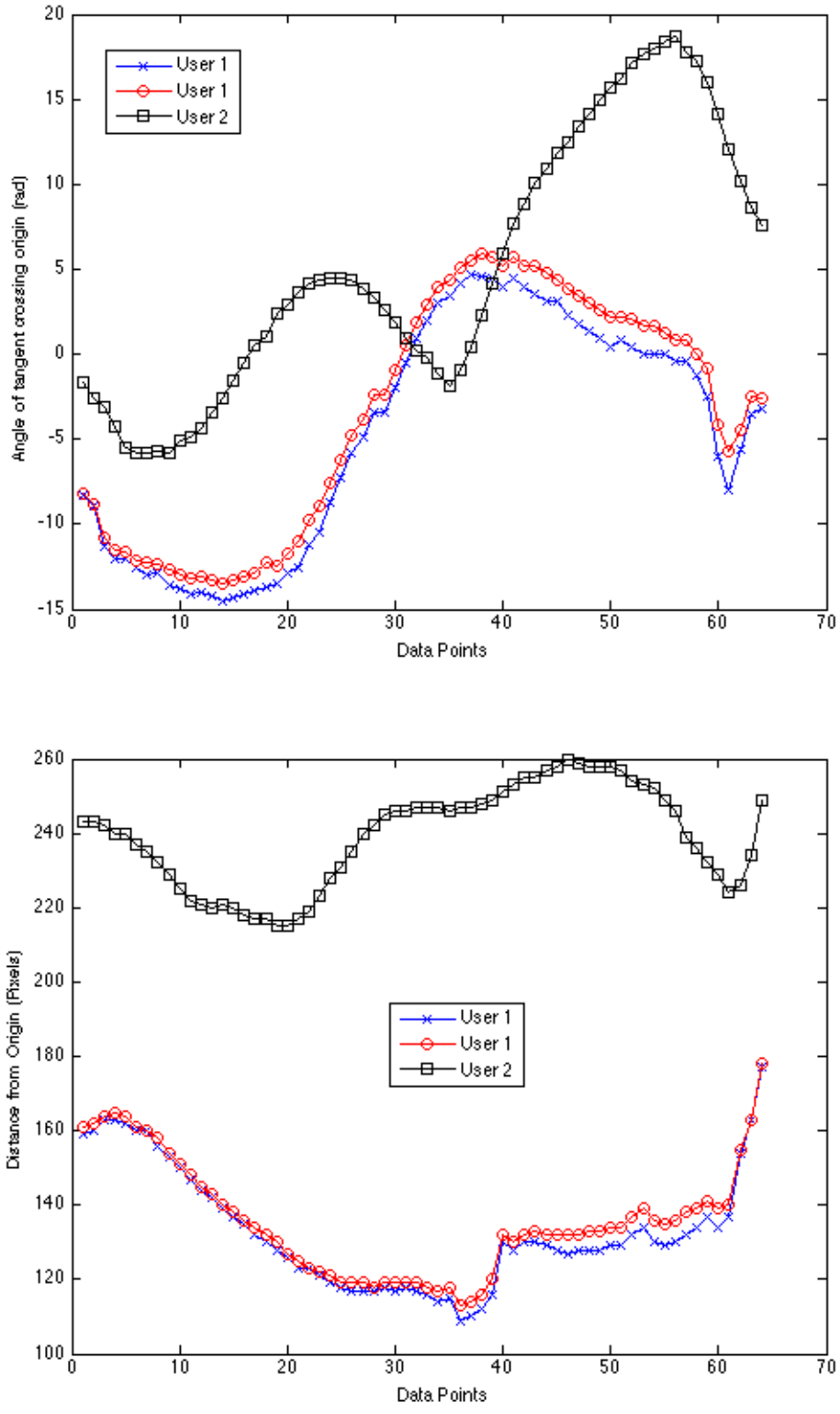


Figure 4.9: Comparing Angle of Curvature and Distance from Origin features of two replica belonging to user 1 and one replica belonging to user 2 for the same gesture.

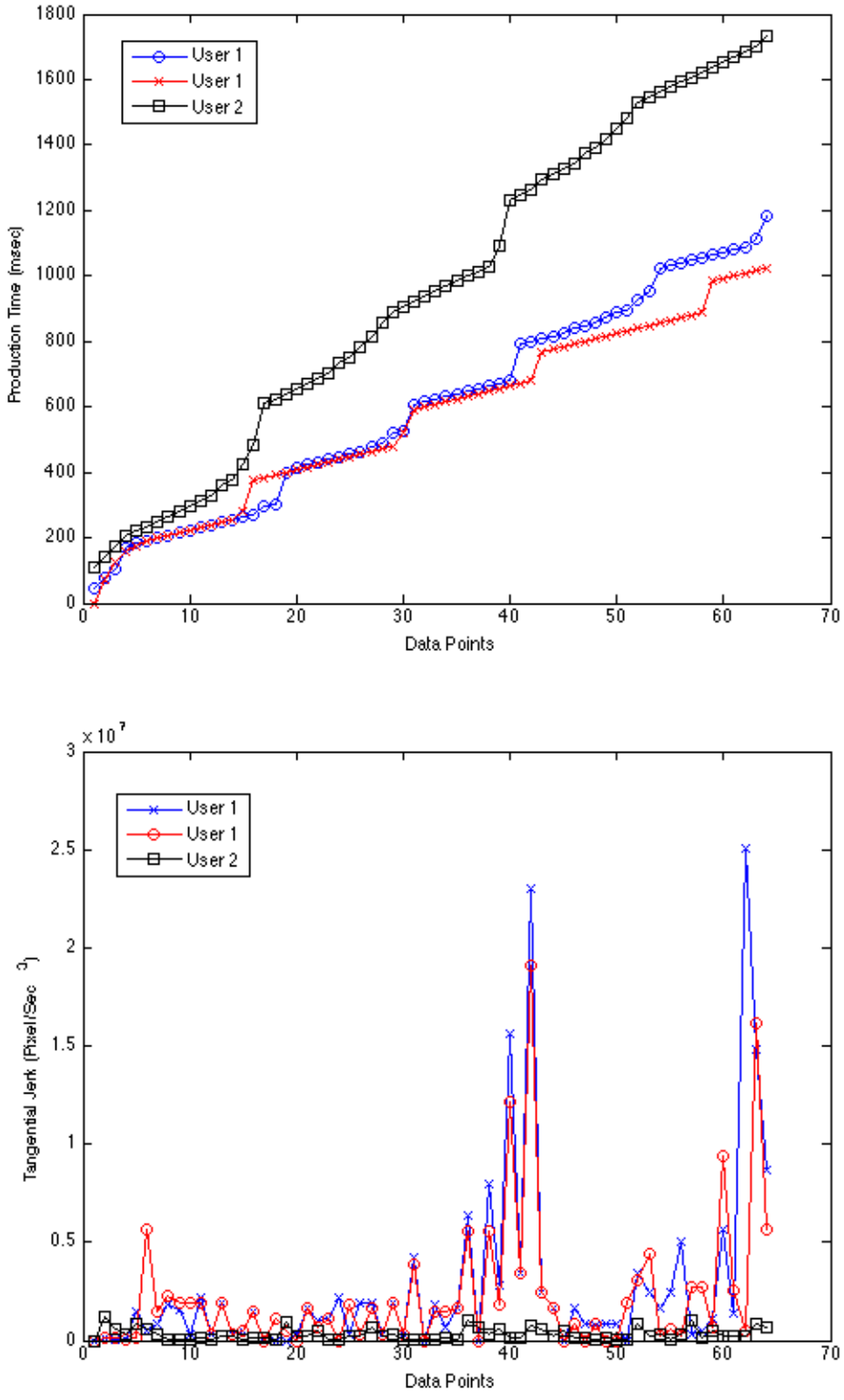


Figure 4.10: Comparing Production Time and Tangential Jerk features of two replica belonging to user 1 and one replica belonging to user 2 for the same gesture.

We do think that the main cause behind PCA poor performance which is also outlined as a known limitation in [34] is the fact that PCA does not account for class separability since it does not make any use of the class labels provided with the data. Which means that using the eigenvectors for classification does not guarantee the maximum discrimination between the different classes. In addition, although our data is a multivariate data we applied the PCA method in a way that does not take into account such information, and all the features were combined as one feature space.

4.5.2 Neural Network Techniques

As mentioned before, we tried two different approaches based on neural networks. However, both approaches use the same neural network type. The Learning Vector Quantization (LVQ) neural network is a supervised variant of the Self Organizing Maps (SOM) developed by Kohonen [35]. Both are examples of competitive learning networks. Generally the goal of competitive networks is clustering the data into a number of clusters such that within the same cluster the data is in some sense alike. In other words the competitive layers modify their weights to recognize frequently presented vectors by measuring the euclidean distance between the weight vectors and the exemplary data. In addition, since the LVQ is a supervised learning version of the SOM, its main purpose is discriminant analysis rather than unsupervised clustering which is the main characteristic of the SOM. Fundamentally, it maps the n-dimensional input space into m-dimensional output space by drawing decision boundaries between the clusters. Another characteristic of LVQ is that it preserves the inherent topology of the training set. The rationale behind that preservation is a nearest neighbor relationship between the input vectors in the training set. Hence the network will be able to classify unseen patterns based on the nearest neighbor

[35]. Generally the LVQ algorithm involves the following steps:

1. With each output neuron o, class label y_o is associated

2. The input vectors (exemplary data) x^p with the correct labels y_o^p will form the learning sample.

3. Computing euclidean distance measure between weight vector w_o and input vector x^p, not only the winner neuron k_1 is determined, but so is the second best neuron k_2.

$$|x^p - w_{k_1}| < |x^p - w_{k_2}| < |x^p - w_i| \ \forall \ i \neq k_1, k_2$$

4. The labels $y_{k_1}^p, y_{k_2}^p$ are compared with the actual label d^p of that exemplary pattern and the weight update rule will be used selectively based on this comparison.

The above steps means that the neuron k_1 with the correct label is moved towards the example pattern and the other neuron k_2 is moved away from it.

In step 4 we used the LVQ2 training algorithm developed by Kohonen [35], which is summarized in Algorithm 2 as follows:

Algorithm 2 LVQ2()

Require: Real number $(0 < \gamma \leqslant 1)$ {where γ is the learning rate}.

if $(y_{k_1}^p = d^p, y_{k_2}^p \neq d^p)$ **then**
  $w_{k_1}(t+1) = w_{k_1}(t) + \gamma \ (x^p - w_{k_1}(t))$ {where t denotes unit time.}
  $w_{k_2}(t+1) = w_{k_2}(t) + \gamma \ (x^p + w_{k_2}(t))$
end if

We explored the usage of the feed-forward back propagation multilayer perceptrons network; however the training step using this type of neural network was exhaustive and time consuming. The justification behind that is the back propagation nature of

the training process. We actually stopped the training process when it exceeded five hours on a modern computer system (2 GHz Core 2 Duo CPU and 2 GB RAM) for only a population of two users.

The Monolithic Neural Network Approach

Typically, the LVQ neural network consists of three layers: an input layer, a hidden layer which is sometimes referred to as Kohonen layer, and an output layer. The network is depicted by Figure 4.11. The number of neurons in the output layer of the standard LVQ architecture is always equal to the number of the different classes characterizing the input data. This means that the output of the neural network is an index of one of the classes in the training population. In our trial to find the best architecture to solve our problem, the output layer consisted of $N_{output} = U$, where U is the number of classes (users) the neural network is classifying. Normally, the number of neurons in the input layer has a relation with the input data. In our case, $N_{input} = GS \times F$ neurons, where GS is the gesture size and F is the number of features. We used LVQ2 as the learning algorithm and started experimenting with different setups to find out what is the best number of neurons in the hidden layer. Based on our pilot study, we observed the following:

- The more we increase the number of users in the training population, the lower is the network performance, given that the numbers of features and neurons in the hidden layer are fixed.

- The more we increase the number of features, the lower is the network performance, given that the numbers of users and neurons in the hidden layer are fixed.

Based on the previous observations, we concluded that finding the optimal number of neurons in the hidden layer will not be achieved unless we fix both the number of features and users in the training population. This means that each time we add a new user to our experiment we need to change the architecture of the neural network, which is not an acceptable solution. On the other hand, using all the extracted features will limit the number of users the network can classify with acceptable performance. This is especially valid, when one neural network is used to classify all the users in the training population. We do think that the root cause of these limitations goes back to the monolithic nature of the network architecture.

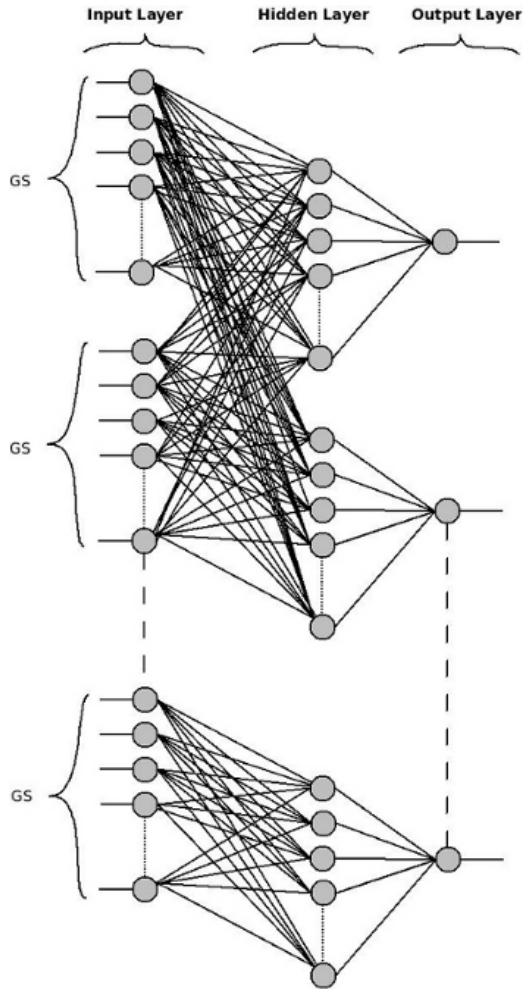


Figure 4.11: The Monolithic LVQ Neural Network

The Modular Neural Network Approach

Like any neural network design, the modular neural network design emulates the functioning of the human brain. The brain is a highly structured entity with localized regions of neurons specialized in performing specific tasks. Hence the main rational behind the modular neural network design. On the other hand, a monolithic neural network is considered as an unstructured black box that does not have the flexibility and modularity to solve a subset of certain problems. In [**?**], the monolithic nature of a neural network is proven to be one of its major performance limiting characteristic.

We used the modular neural network design since we were not satisfied with the monolithic neural network performance. We do see that having a modular neural network design best suite our problem domain. As we outlined previously, the monolithic network did not scale well when we added either more features or users in the system. Since we believe that all the extracted features are needed to correctly identify the individuals. We designed a modular neural network that make use of all the extracted features. We grouped the input features into four feature sets FS_i, where $i = 1, 2 \ldots 4$ and we designed the neural network so that it has four modules NNM_i, $Where\ i = 1, 2 \ldots 4$. Each of the four feature groups will be an input for a module of the neural network $FS_i \rightarrow NNM_i$. The grouping of the features into the sets was based on the logical relations between the features. Note that feature l is used in two different groups. We grouped the features into two spatial sets, one spatiotemporal set, and one temporal set as follows:

1. Spatiotemporal Set $(FS_1) : x, y$, and t
2. Spatial Set $(FS_2) : l, c$, and δc
3. Temporal Set $(FS_3) : hv, vv, tv, ta$, and tj
4. Spatial Set $(FS_4) : l$ and θ

The architecture of the neural network modules is the same for all the users in the training population. Each module of the neural network is an independent system by itself and follows the typical LVQ design. As mentioned before, the number of neurons of the output layer of the LVQ is equal to the number of class the network is characterizing. In our modular neural network design the output layer has two neurons, as the network is classifying two classes. The first class is the current user class (index) or self and the second class represent the remaining users in the training population or non-self. We will explain in a later section how the training of the neural network works. Based on the data collected in the pilot study and different setups of the neural network architecture, we came to the conclusion that 72 neurons in the hidden layer was acceptable. The number of neurons in the hidden layer is similar for all the modules. The hidden layer and the output layer design is similar for all the modules in the network. The input layer is the exception; as the number of neurons in the input layer for each module equals to the number of features in the corresponding feature set multiplied by the gesture size, which is expressed mathematically as follows $N_{input}(NNM_i) = F_i \times GS$, where F_i is the number of features in feature set $FS_i, (1 \leq i \leq 4)$. Figure 4.12 illustrates the general module architecture of the LVQ neural network.

Every module in the neural network is considered as a separate classifier. The decision of each module does not affect the other modules. That being said, the final decision is based on the aggregated output of the four modules. The output of each module is aggregated through a decision fusion scheme; the output of this fusion scheme is the actual final decision. Although many different fusion schemes could be used, we used in our framework a scheme based on majority voting technique to derive the final decision. The majority voting technique requires λ modules to agree on the same decision in order to consider that decision final where $\{2 \leq \lambda \leq 4\}$.

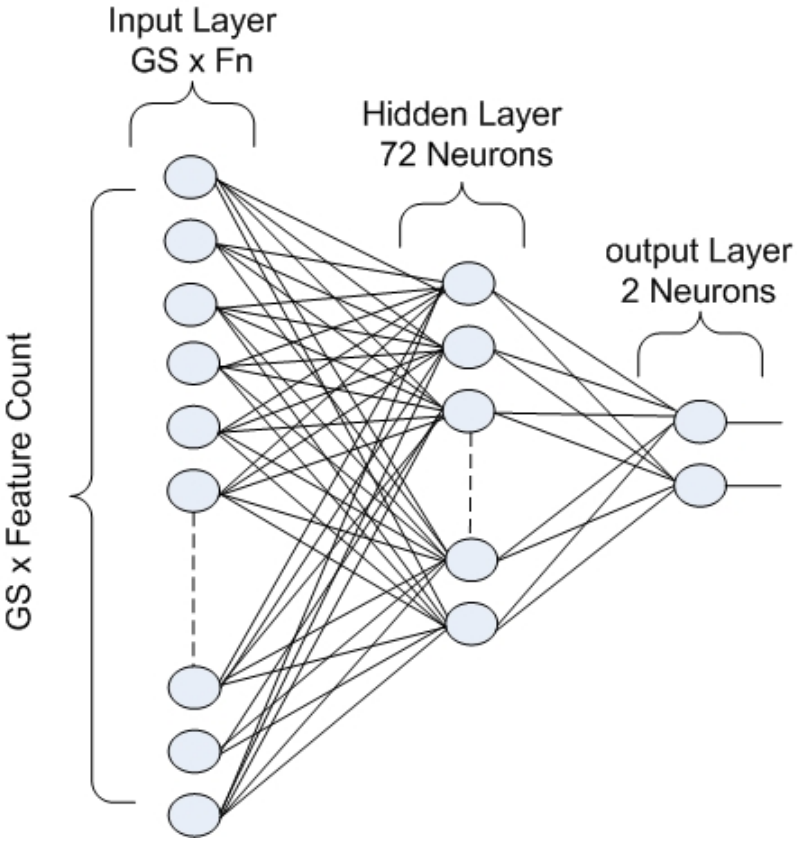


Figure 4.12: General module architecture of the LVQ neural network .

If none of the modules agree on the same decision (i.e. $\lambda \leq 1$) the majority voting fusion scheme will reject the input sample, which means that the network is not able to successfully classify the input. In other words the neural network was not able to find a close neighbor to map the input to an output in the topology of the training set. Figure 4.14 illustrates the majority voting fusion scheme.

Neural Network Training

Figure 4.13 illustrates how the different feature sets are used to train the modular neural network. We follow the regular procedure of training the LVQ neural network by presenting the sample data along with its labels. In our framework, there is a separate profile for each gesture for each user. It is important to note that while

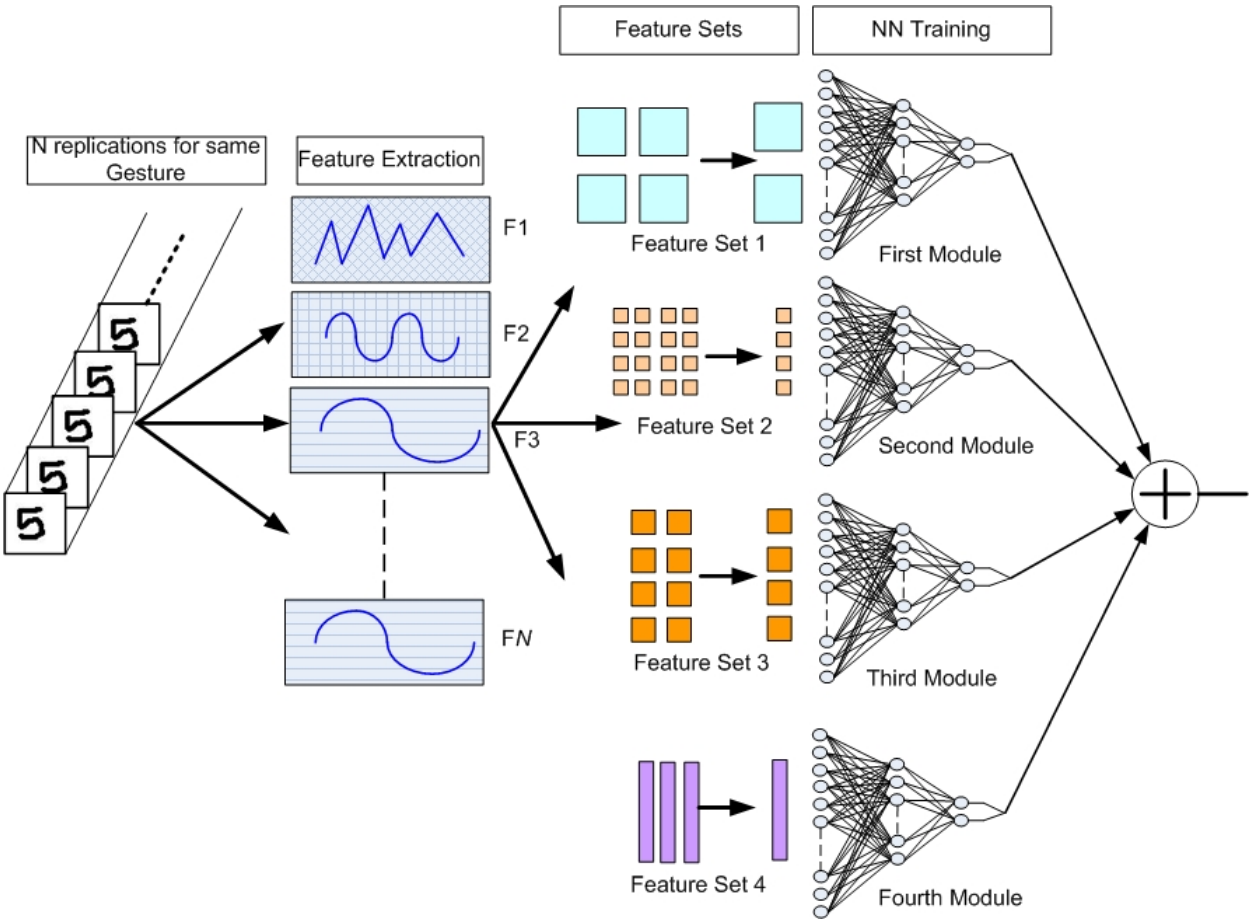


Figure 4.13: Training the modular LVQ network with the different feature sets.

all the gestures share the same modular neural network architecture explained in the previous section, each gesture for each user is associated with a separate set of weights derived through training. For each gesture for each user, the associated weights will represent the corresponding profile.

In our pilot study all the gesture replications drawn by each user were divided into two separate sets, a training set and a test set. We apply a hierarchical training procedure to train the neural network. The main objective of the hierarchical training is to improve the performance of the network. An example of the procedure is illustrated in Figure 4.15. We form a three level tree-like hierarchy from the training set by averaging the different replications drawn by the user. The averaging of the repli-

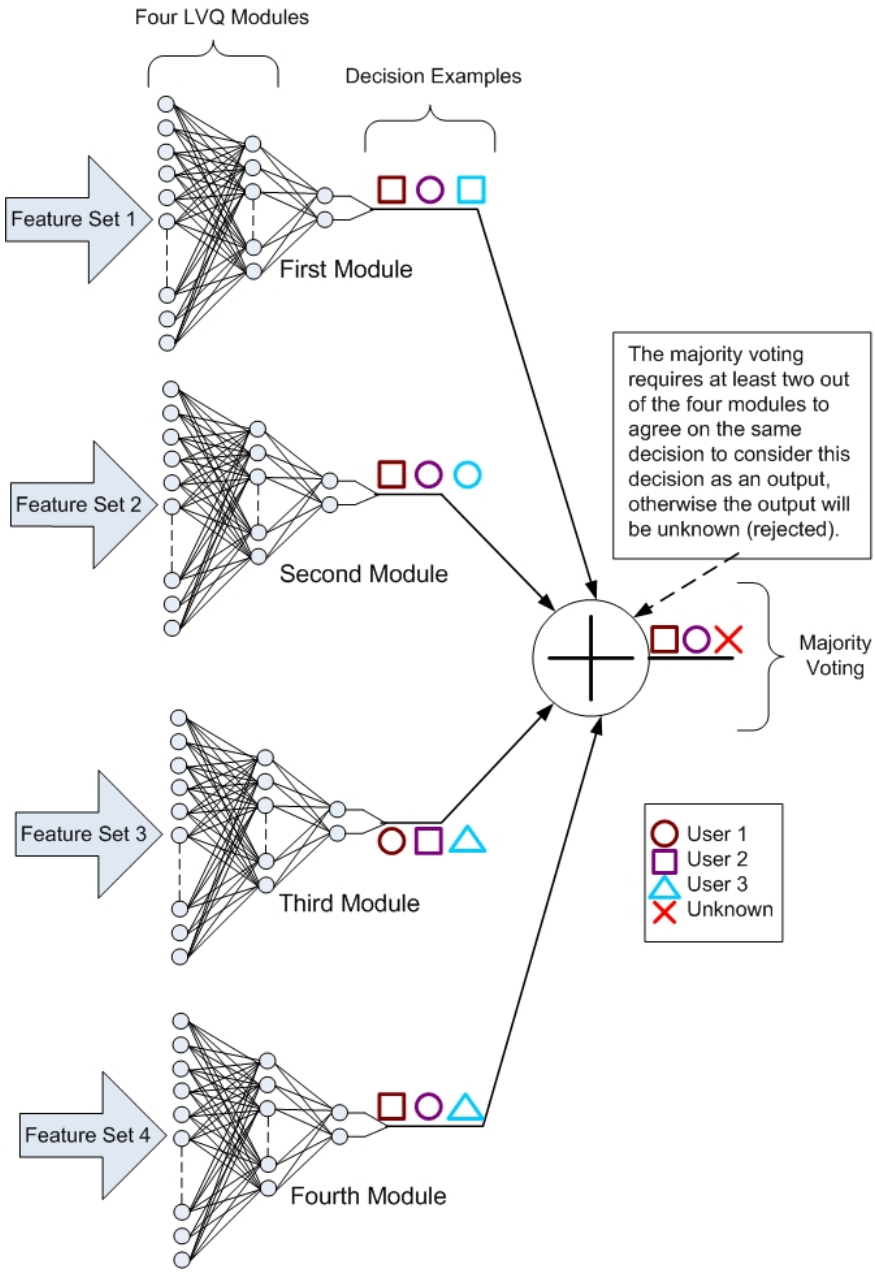


Figure 4.14: The modular LVQ majority voting fusion scheme.

cations follows a specific procedure. The procedure starts by dividing the replications in the training set into groups of equal size which form the bottom most level. The size of each group τ must be less than the size of the training set ω. The average of each group is calculated to form the second level of the hierarchy. We call the second

level, the level of the sub-means. The root of the hierarchy which is the overall mean of the replications, get formed by calculating the average of the sub-means in the second level which forms the top most level. After constructing such hierarchy the training of the modular neural network starts from the root to the bottom level of the hierarchy. In other words, we construct the hierarchy in a bottom-up approach and then we train the network in a top-down way.

The rational behind such technique is that starting the training of the neural network with the overall mean of the replication allows the weights of the neural network to get quickly adjusted to the center of the current user cluster of replications. Then the weights get adjusted to the sub-means in the second level until reaching the actual replications of the user in the bottom level. In the pilot experiment we noticed that by applying the hierarchical training technique, the training error in some cases converges to zero, which is optimal.

We train the neural network through both positive and negative training. Obviously, positive training is based on the set of replications drawn by user X. However, the selection of the negative training set is not straightforward. One way of conducting the negative training is by considering all or a subset of the replications of the remaining users in the training population as the negative training set. For instance, if we have n users in the training population the negative training set for each user may be the replications of the remaining $n-1$ users. Clearly this is not the optimal solution from a scalability point of view. Ideally the negative training set should contain the replications of the nearest subject to the current user. This should be good enough for the neural network to draw the decision boundary between the current user and the closest subject, which in turn allows the network to discriminate between the current user and all the subjects in the training population. In our implementation, we used formula 4.4 shown below, to pick the nearest and the second nearest users

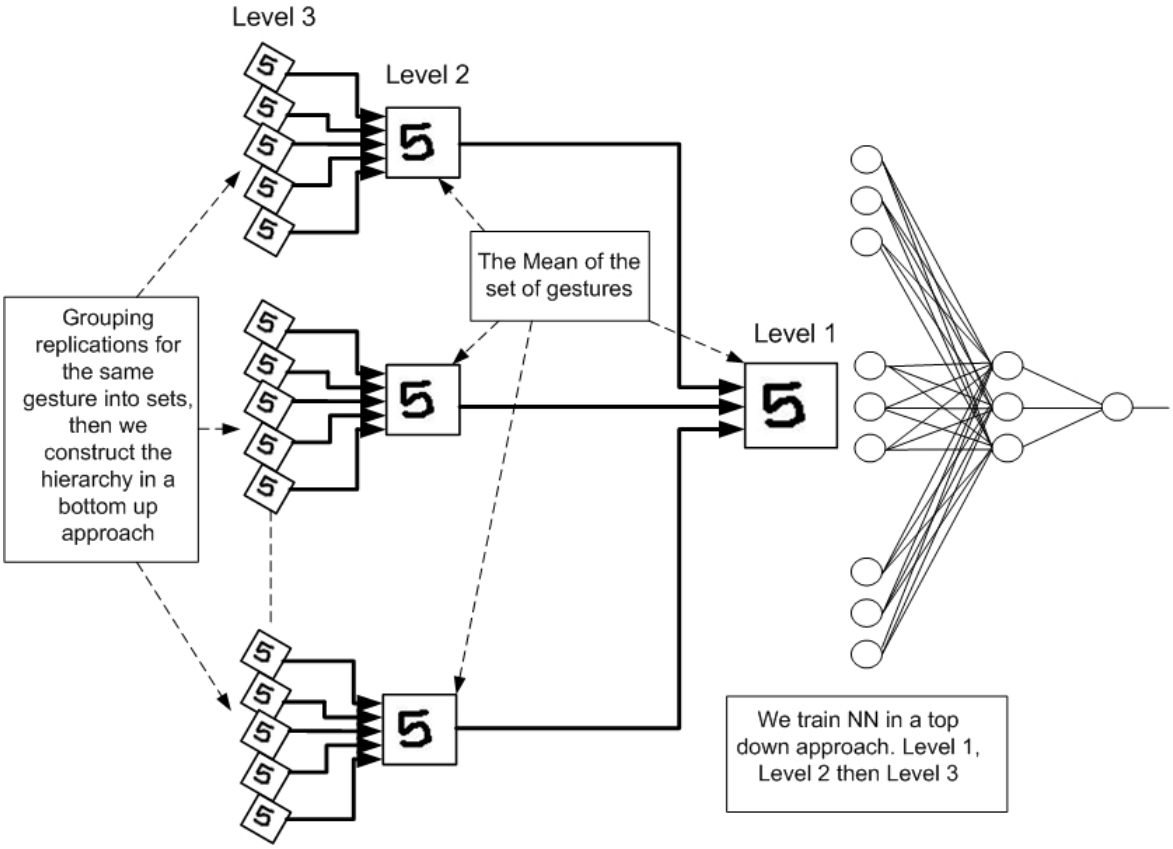


Figure 4.15: The Hierarchical LVQ Neural Network Training

to conduct the negative training. The same formula has been used by Lee *et al.* in [36] to measure the distance between the feature vectors in their proposed system for on-line human signature verification. Let $m(a,i)$ and $\sigma^2(a,i)$ be the sample mean and the sample variance of feature i computed from the replications of a specific gesture for user a, respectively. Then the distance measure for feature i between user a and user b for a specific gesture is defined as:

$$d_i(a,b) = \frac{|m(a,i) - m(b,i)|}{\sqrt{\sigma^2(a,i) + \sigma^2(b,i)}} \tag{4.4}$$

Based on the above formula, the distance between user a and user b is defined as:

$$d(a,b) = \sum_{i=1}^{n} d_i(a,b) \tag{4.5}$$

We say that user b is the nearest subject to user a in population P if:

$$d(a,b) = min_{x \in P \ and \ x \neq a} d(a,x)$$

We take the gesture replications of the nearest and second nearest users in population P as the negative training samples. Then we apply the hierarchical training technique discussed previously. We repeat the same procedure for each user in the training population. Finally all the derived profiles are stored to be used later in the test phase.

4.6 Test Session and Parameters

At the beginning of the testing session, the user will be asked to draw different types of gestures selected from the set of gestures provided during the enrollment phase; let ζ denote such number. The user will be asked to provide a number of replications for each of the selected gestures; let α denote such number. For a reproduction session of a gesture to be considered valid, a number of the provided replications should match successfully the profile; let β denote such number; note that $1 \leq \beta \leq \alpha$. Table 4.6 summarizes the above mentioned variables. As mentioned before, during the enrollment, the data acquisition module compares the provided sample against the example gesture and provide visual feedback to the user. Note that such early comparison targets only the shape not the dynamics and is for error checking. In some cases the user makes a mistake while drawing the gesture, this early comparison is designed to handle such mistakes. Furthermore this comparison is rather superficial and not

as deep as the one used during the classification where more powerful techniques are used.

During a test session, data acquisition may be carried either by including or removing the early comparison step which provides visual feedback to the user. In this work, we have decided primarily to include the early comparison in the test phase for our main experiment. Furthermore, by collecting an additional data sample we will study the impact of not carrying the early comparison. Nevertheless, carrying the early comparison might distract the impostor as the same type of gesture is drawn differently by different individuals. The early comparison will require the attacker to draw the gesture in the same way the legitimate user does (from a shape perspective), which could influence the behavior of the impostor and his/her mouse gesture dynamics profile might deviate from the legitimate one.

Variable	Description
α	corresponds to the number of gesture replications needed to be performed by the user for a specific gesture.
β	corresponds to how many of the α replications need to be accepted in order for that given gesture to be considered successful. Note that $\beta \leq \alpha$.
ζ	corresponds to how many different types of gestures the user must draw in the test phase.

Table 4.2: System variables used by the data acquisition module for the test phase.

4.7 Summary

In this chapter, we presented the analysis and detection technique underlying our framework. We started with the gesture creation module which is used to create the gestures for enrollment. We then presented the data acquisition module that actually captures, preprocesses and smoothes the raw data. The feature extraction module

then extracts more features from the raw data which form the signature of the users. The signature of the users are used as input for the classification module. In the course of finding an effective classifier we applied different classification techniques. The PCA and the monolithic neural network techniques were not suitable for our problem in terms of classification performance. The modular neural network was presented as the proposed solution. The architecture of the network was explained and the decision fusion scheme was discussed. The hierarchical training process which involves positive and negative training was discussed and the selection of the positive and negative training data was clarified. Finally, after the training is done the weights of the neural network with the users credentials are stored as profiles in the user database; these profiles are used later in the test phase.

Chapter 5

Experiment, Evaluation, and Analysis

In order to evaluate our work; we conducted an experiment involving several participants. In this chapter, we present, analyze, and discuss the achieved results.

5.1 Method

The main objective of our experiment was to be able to recognize individuals based on their mouse gestures. Ideally, the system should be able to recognize with high degree of accuracy the behavior of each user while replicating a specific gesture. To achieve such goal, 41 participants were involved in our experiment. The participants ages ranged from 17 to 48 years old and from different backgrounds but generally all the participants in the experiment were computer professionals, university faculty members and students.

Before starting the experiment the participants were shown a sample gesture set illustrated in Figure 5.1 as an example of uni-stroke gestures. The idea was to get the participants familiar with drawing gestures. All the required software was pre-

installed on a laptop machine. All the participants used the same laptop to draw the same set of gestures which was pre-chosen. The participants drew the gesture templates first and then replicated each gesture template 30 times. The gestures replications along with the participants credentials were stored in a user database. There was only one requirement which was to draw such gestures in one stroke, as in practice, programs that make use of mouse gestures typically implement them in a uni-stroke form.

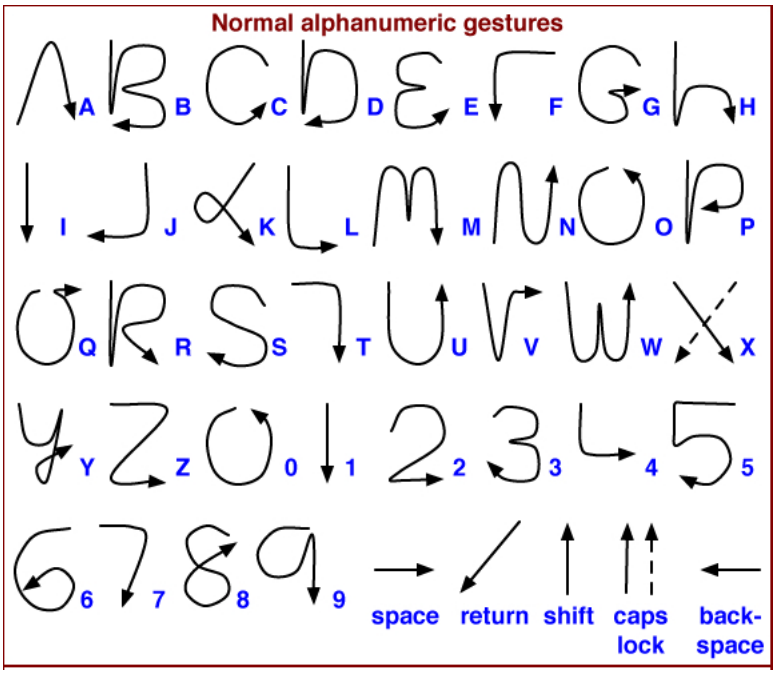


Figure 5.1: Graffiti gesture set used as examples gestures drawn in uni-stroke.

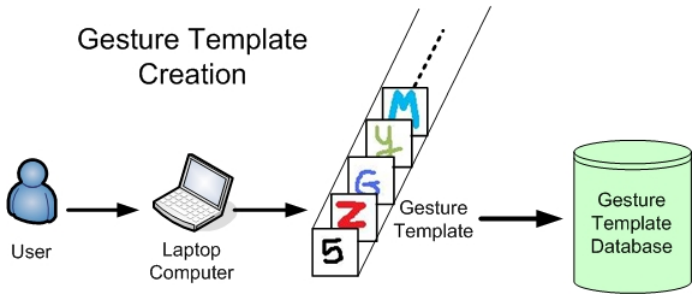
5.2 Apparatus

As we mentioned before, all the participants used the same laptop to enroll in our experiment. The software involved in our experiment was already deployed on the laptop. The software consisted of a gesture creation tool and an enrollment tool. The gesture creation tool is used to create the gesture templates and store them with

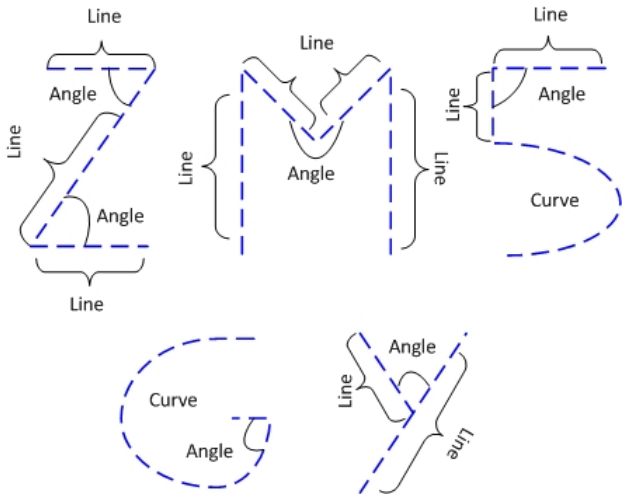
the user credentials in a database. The enrollment tool loads the templates from the database and allows the participant to enroll against them. The replications resulted from the enrollment are stored in the replications database. The hardware configuration of the laptop was an Intel Core 2 Duo processor clocked at 2 GHz with 2 GB of physical memory, running Microsoft Windows Vista. All the participants used Microsoft Explorer optical mouse to replicate the different gestures, even the same mouse pad was used during the experiment.

5.3 Data Collected

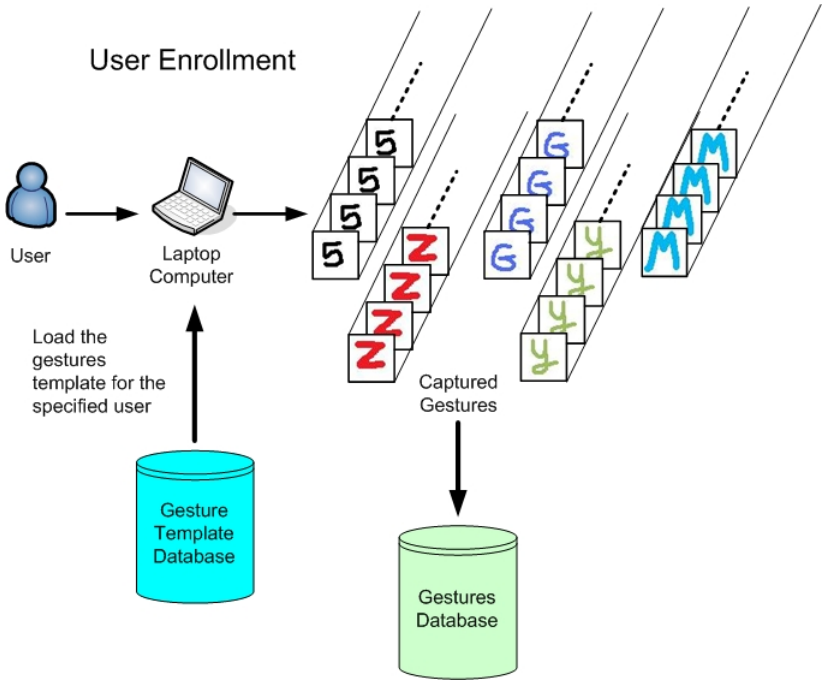
The participants in the experiment were asked to draw five gesture templates as demonstrated in Figrue 5.2(a). Four gestures were chosen from the English language letters and one gesture was the Arabic numerical five. The gestures were chosen to include combinations of lines, angles and curves; Figure 5.2(b) illustrates the lines, angles and curves in each gesture involved in the experiment. The main assumption behind that is, the more angles and curves the gesture has, the more it will require muscle tension and concentration from the users. Which in turn will impose the intrinsic behavior of the human motor control while drawing such gestures. Figure 5.2(c) illustrates the enrollment process of the five gestures. We collected 30 replications for each gesture which resulted in 150 replications per participant, giving in total 6,150 replications. For each user, we selected 25 replica randomly out of the 30 for training purpose and the remaining five were used in the test phase. This means that $\alpha = 5$ and β can range between $(1 \leq \beta \leq 5)$. Both α and β were explained previously in Table 4.6. It is important to note that the same preprocessing, matching and smoothing steps of the data acquisition module were applied on the test data.



(a) Gesture Template Creation.



(b) The lines, angles, and curves of the gestures involved in the experiment.



(c) The enrollment process for the five gestures in our experiment.

Figure 5.2: The gesture decomposition and its creation and enrollment steps in the main experiment.

5.4 Evaluation Process

In order to evaluate the performance of our framework we conduct a one-hold-out cross-validation test involving n rounds, n being the size of our user population. Having $n = 41$ users in our population we repeated the cross-validation test 41 times. In each round $i, (1 \leq i \leq n)$, one of the n users is considered as an outsider (or impostor) who attempts to impersonate each of the $n-1$ users considered as insiders (or legal) users. Let user U_i be the outsider in round i. We start by building a reference profile for each of the $n-1$ remaining users using 25 replicas from each of them as the enrollment samples. Note that none of the replicas from user U_i will be used in this process.

During a test session, the participant is asked to replicate each gesture a certain number of times; for this evaluation, we used $\alpha = 5$ as such number. The 30 replicas collected for user U_i will be divided into six test sets, each consisting of five replicas selected randomly. To calculate the false acceptance rate (FAR), the six test sets will be compared against the profiles of the insiders $U_j, (1 \leq j \leq n, j \neq i)$. Similarly all the remaining replicas not used in the enrollment for the other insiders $U_k, (1 \leq k \leq n, k \neq j, j \neq i, i \neq k)$ will be tested against the profile of U_j. Considering that each insider has five remaining test replicas, this means one test set (of five replicas) per insider. So the total number of trials in round i will be $N_i = (n-1)(6+n-2) = (n-1)(n+4)$, which gives in overall for all the rounds, a total number $N = n(n-1)(n+4)$. A false acceptance FA will occur if the number of matching replicas is above the threshold β:

$$FA = \begin{cases} 1, & if\ number\ of\ matching\ replicas \geq \beta \\ 0, & otherwise \end{cases}$$

The global FAR is computed as the ratio between the total number of false acceptance

over all the test trials and the total number of test trials N:

$$FAR = \frac{\sum_{i=1}^{n} \sum_{j=1}^{(n-1)(n+4)} count(\{FA\}_{ij})}{n(n-1)(n+4)} \times 100$$

$$where\ i\ (1 \leq i \leq n)\ is\ the\ round\ index,$$

$$and\ j\ (1 \leq j \leq (n-1)(n+4))\ is\ the\ test\ trail\ index.$$

To calculate the false rejection rate (FRR), during round i the remaining five replicas (not involved in the enrollment) of each of the insider $U_j, (1 \leq j \leq n, j \neq i)$, is tested against their own reference profile. This corresponds to a total number $n-1$ of trials over round i. So the total number of trials for all the rounds is $M = n(n-1)$. A false rejection FR will occur if the number of matching replicas is below the threshold β:

$$FR = \begin{cases} 1, & if\ number\ of\ matching\ replicas < \beta \\ 0, & otherwise \end{cases}$$

The global FRR is computed as the ratio between the total number of false rejection over all the test trials and the total number of test trials M.

$$FRR = \frac{\sum_{i=1}^{n} \sum_{j=1}^{(n-1)} count(\{FR\}_{ij})}{n(n-1)} \times 100$$

$$where\ i\ (1 \leq i \leq n)\ is\ the\ round\ index,$$

$$and\ j\ (1 \leq j \leq (n-1))\ is\ the\ test\ trail\ index.$$

As mentioned before, system variable α was selected and fixed to equal five replica however, we varied the values of the other two system variables to measure their impact on the framework performance. Namely, the majority voting system variable

λ which ranged from $(2 \leq \lambda \leq 4)$ and β that ranged from $(1 \leq \beta \leq \alpha)$. The global $FAR_{\beta,\lambda}$ and $FRR_{\beta,\lambda}$ are calculated accordingly for each system variable value.

5.5 Evaluation Results

We applied the mentioned one-hold-out cross-validation method separately for each of the five gestures ($\zeta = 1$) involved in our experiment, and computed global FRR and FAR while varying α and β. The obtained results are shown in Tables 5.1 - 5.5 and depicted using DET curves in Figures 5.3 - 5.7.

$\boldsymbol{\lambda = 2}$	$\boldsymbol{\beta = 1}$	$\boldsymbol{\beta = 2}$	$\boldsymbol{\beta = 3}$	$\boldsymbol{\beta = 4}$	$\boldsymbol{\beta = 5}$
FAR	64.33	49.47	37.24	26.38	15.79
FRR	0	0	0.15	3.14	21.28
$\boldsymbol{\lambda = 3}$	$\boldsymbol{\beta = 1}$	$\boldsymbol{\beta = 2}$	$\boldsymbol{\beta = 3}$	$\boldsymbol{\beta = 4}$	$\boldsymbol{\beta = 5}$
FAR	31.09	18.58	11.26	6.43	2.86
FRR	0.30	1.46	5.02	23.75	56.70
$\boldsymbol{\lambda = 4}$	$\boldsymbol{\beta = 1}$	$\boldsymbol{\beta = 2}$	$\boldsymbol{\beta = 3}$	$\boldsymbol{\beta = 4}$	$\boldsymbol{\beta = 5}$
FAR	8.65	3.53	1.47	0.47	0.11
FRR	14.63	33.93	60.39	82.95	95.76

Table 5.1: The recognition performance for “G” gesture.

$\boldsymbol{\lambda = 2}$	$\boldsymbol{\beta = 1}$	$\boldsymbol{\beta = 2}$	$\boldsymbol{\beta = 3}$	$\boldsymbol{\beta = 4}$	$\boldsymbol{\beta = 5}$
FAR	65.24	50.24	37.61	26.4109	15.54
FRR	0	0	0.27	3.50	21.18
$\boldsymbol{\lambda = 3}$	$\boldsymbol{\beta = 1}$	$\boldsymbol{\beta = 2}$	$\boldsymbol{\beta = 3}$	$\boldsymbol{\beta = 4}$	$\boldsymbol{\beta = 5}$
FAR	31.76	18.79	11.28	6.26	2.69
FRR	0.36	2.01	7.68	25.36	57.50
$\boldsymbol{\lambda = 4}$	$\boldsymbol{\beta = 1}$	$\boldsymbol{\beta = 2}$	$\boldsymbol{\beta = 3}$	$\boldsymbol{\beta = 4}$	$\boldsymbol{\beta = 5}$
FAR	8.78	3.63	1.57	0.52	0.14
FRR	15.85	35.15	60.06	81.52	94.48

Table 5.2: The recognition performance for “Y” gesture.

As we can see from the results all the gestures excluding the ”Z“ gesture are near to each other. Some of the best operating points obtained in those cases vary for the

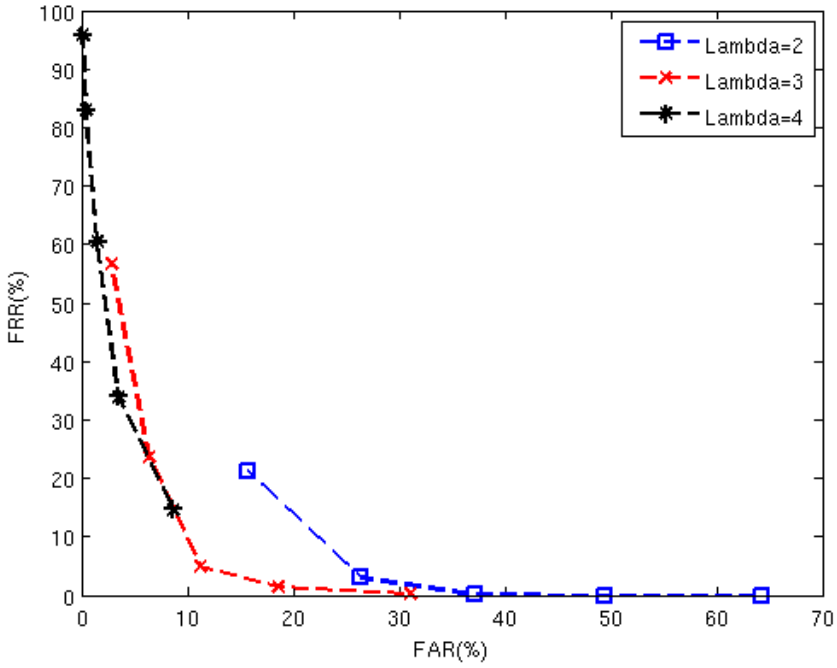


Figure 5.3: The DET curve for the "G" gesture.

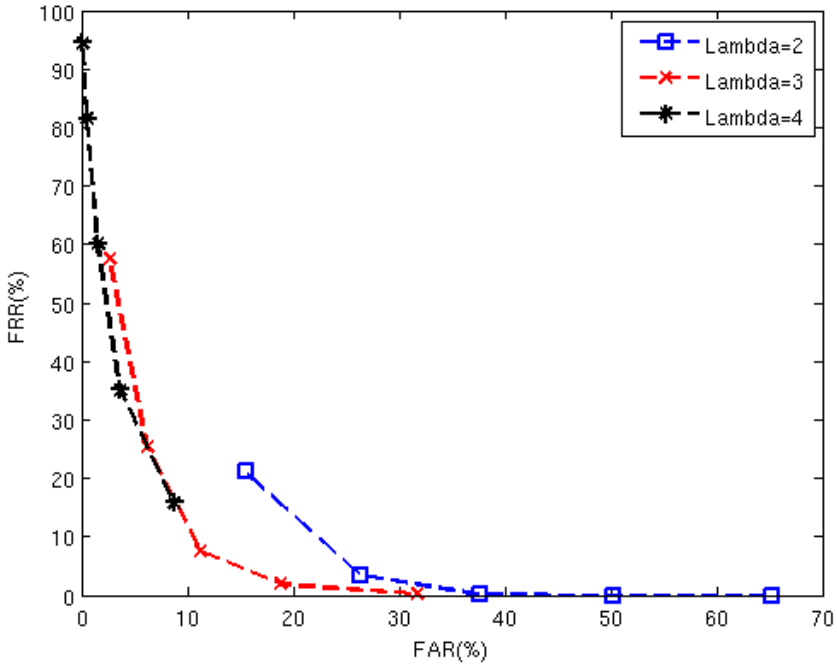


Figure 5.4: The DET curve for the "Y" gesture.

$\lambda = 2$	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
FAR	54.39	41.20	30.45	21.80	13.03
FRR	0	0	0.36	3.47	19.89
$\lambda = 3$	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
FAR	26.10	14.82	9.19	5.30	2.77
FRR	0.21	2.07	7.74	23.23	55.44
$\lambda = 4$	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
FAR	7.27	3.10	1.34	0.55	0.20
FRR	12.05	35.83	59.60	84.31	95.08

Table 5.3: The recognition performance for number “Five” gesture.

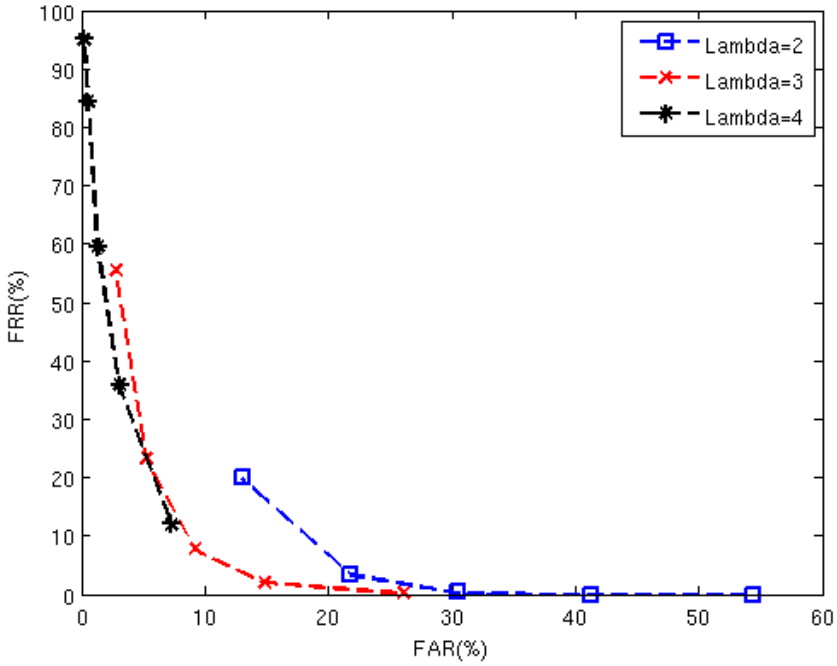


Figure 5.5: The DET curve for the number “Five“ gesture.

FAR between 9.19% to 11.28% while the FRR stand between 5.02% and 9.41%, when $\beta = 3$ and $\lambda = 3$. We do think that the main reason that the performance of the ”Z“ gesture is worse than the other gestures is, the ”Z“ gesture does not have any curves in its shape. For example, at $\lambda = 3$ and $\beta = 3$ the difference between the ”Z“ gesture and the ”M“ gestures is 3.88% for the FAR and 7.78% for the FRR. In any case, it appears from the above results that using a single gesture may not be very effective

$\lambda = 2$	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
FAR	57.12	43.54	32.62	23.66	14.45
FRR	0	0.06	0.33	2.95	19.04
$\lambda = 3$	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
FAR	26.72	15.48	10.35	5.88	3.12
FRR	0.15	1.98	9.41	22.32	54.46
$\lambda = 4$	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
FAR	7.28	3.26	1.43	0.60	0.24
FRR	9.74	33.97	57.71	83.67	95.08

Table 5.4: The recognition performance for the "M" gesture.

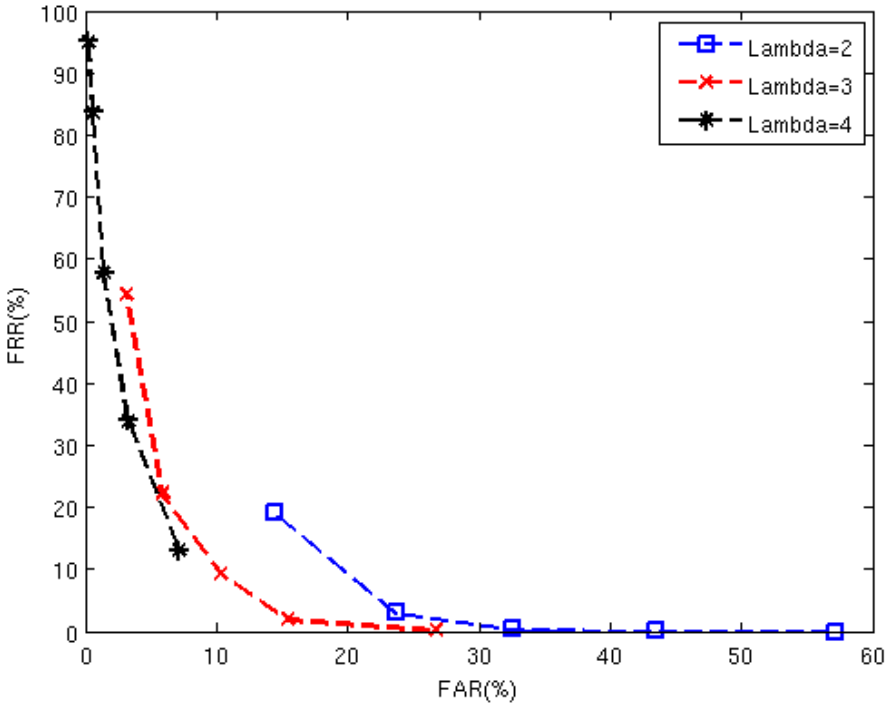


Figure 5.6: The DET curve for the "M" gesture.

for static authentication.

Under such consideration, we modified our evaluation method so that it includes two gestures in the test session ($\zeta = 2$) as opposed to only one gesture ($\zeta = 1$) in the previous evaluation. The number of replicas drawn by the individuals remained fixed at $\alpha = 5$ for each gesture however, we varied β system variable. This means that

$\lambda = 2$	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
FAR	68.13	53.71	42.00	30.08	17.61
FRR	0	0	0	5.00	47.50
$\lambda = 3$	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
FAR	37.72	21.89	13.65	7.53	3.52
FRR	2.50	10.00	15.00	42.50	80.00
$\lambda = 4$	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
FAR	11.38	4.17	2.11	0.75	0.16
FRR	20.00	47.50	67.50	82.50	97.50

Table 5.5: The recognition performance for "Z" gesture.

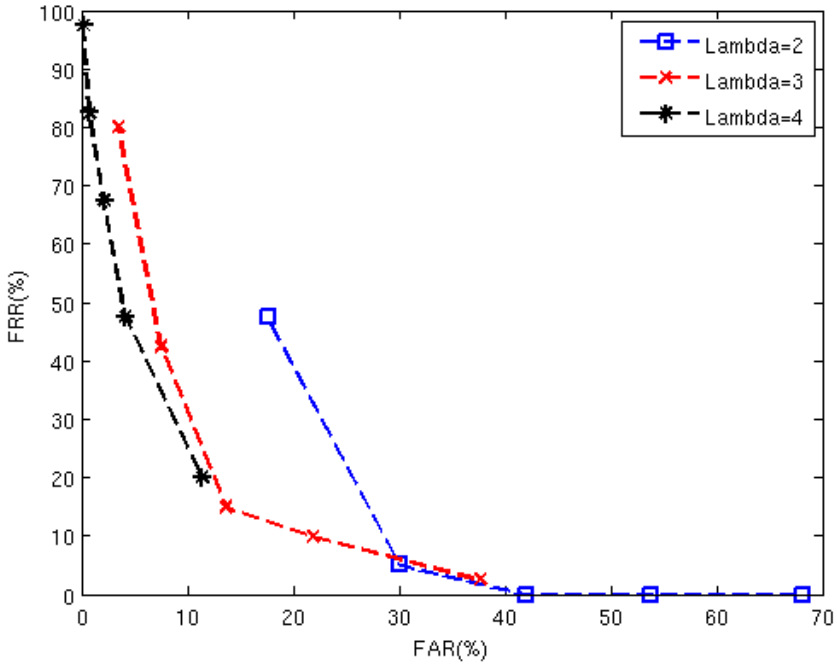


Figure 5.7: The DET curve for the "Z" gesture.

the user will be required to draw in total ten replicas in the test session. Let β_1 be the number of accepted replicas from first gesture and β_2 be the number of accepted replicas from the second gesture. A false acceptance will occur if any combination of β_1 and β_2 satisfies $\big(\beta_1 + \beta_2 \geq \beta,\ where\ \beta_1, \beta_2 \leq \alpha\big)$, and false rejection will occur if any combination of β_1 and β_2 satisfies $\big(\beta_1 + \beta_2 < \beta,\ where\ \beta_1, \beta_2 \leq \alpha\big)$. Tables 5.6 - 5.7 and Figures 5.8 - 5.9 illustrate the performance results and DET curves

obtained for the combination of the "Five" gesture with "G" gesture at one time and with the "M" gesture at a second time. We observe a significant improvement in the performance with (FAR = 3.59%, FRR = 3.35%) for ("Five", "G") combination, and (FAR = 3.13% , FRR = 6.28%) for ("Five", 'M"') combination, when $\beta = 6$ and $\lambda = 3$

$\lambda = 2$	$\beta = 4$	$\beta = 5$	$\beta = 6$	$\beta = 7$	$\beta = 8$
FAR	51.60	40.46	25.92	17.83	11.22
FRR	0	0	0	0	0.1
$\lambda = 3$	$\beta = 4$	$\beta = 5$	$\beta = 6$	$\beta = 7$	$\beta = 8$
FAR	41.65	8.67	3.59	1.79	0.8
FRR	0.18	0.6	3.35	9.87	24.81
$\lambda = 4$	$\beta = 4$	$\beta = 5$	$\beta = 6$	$\beta = 7$	$\beta = 8$
FAR	1.15	0.33	0.05	0.01	0
FRR	40.06	59.08	78.53	91.76	97.31

Table 5.6: The recognition performance for "Five" gesture and "G" gesture combined.

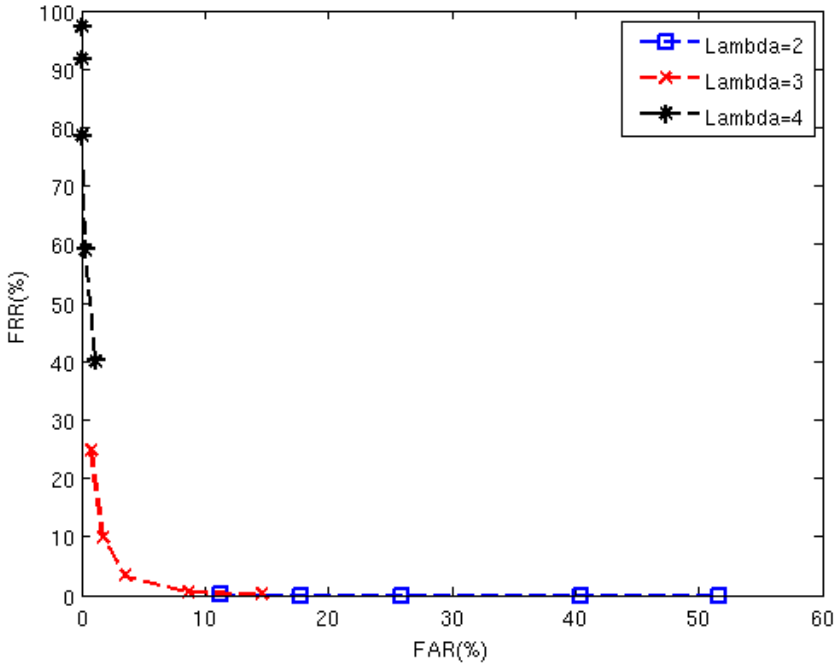


Figure 5.8: The DET curve for the "Five" gesture and "G" gesture combined.

$\lambda = 2$	$\beta = 4$	$\beta = 5$	$\beta = 6$	$\beta = 7$	$\beta = 8$
FAR	50.43	39.51	26.5	17.80	11
FRR	0	0	0	0.73	2
$\lambda = 3$	$\beta = 4$	$\beta = 5$	$\beta = 6$	$\beta = 7$	$\beta = 8$
FAR	12.97	7.17	3.13	1.54	0.71
FRR	2.25	3.35	6.28	15.85	32.92
$\lambda = 4$	$\beta = 4$	$\beta = 5$	$\beta = 6$	$\beta = 7$	$\beta = 8$
FAR	0.93	0.27	0.02	0	0
FRR	39.51	59.81	78.71	91.09	98.41

Table 5.7: The recognition performance for “Five” gesture and “M” gesture combined.

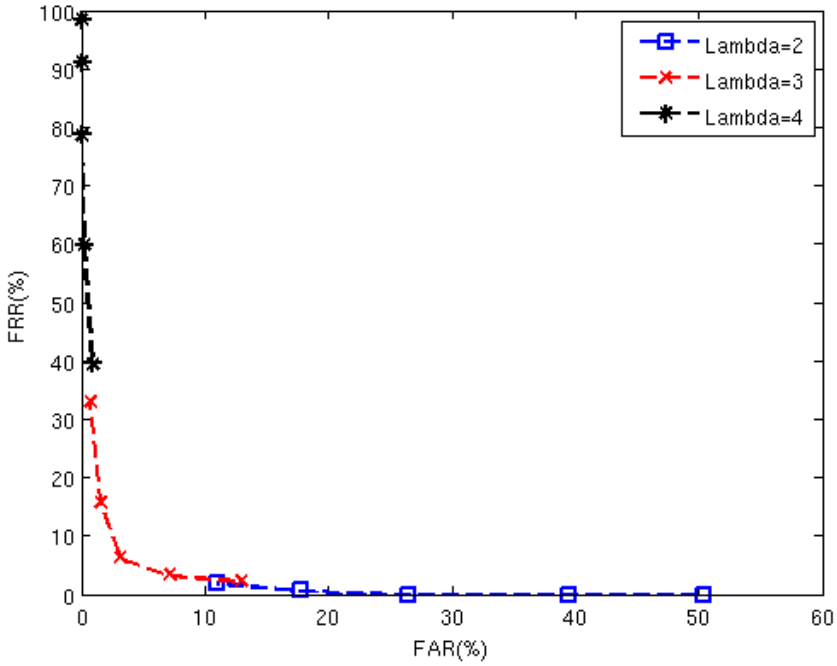


Figure 5.9: The DET curve for the “Five“ gesture and ”M“ gesture combined.

Furthermore, we studied the results of combining three and four gestures ($\zeta = 3, 4$). The previously mentioned method for combining the results of two gestures is applied with the addition of a third and a fourth gesture. Let β_3 corresponds to the number of accepted replicas of the third gesture and β_4 corresponds to the number of accepted replicas of the fourth gesture. Tables 5.8 - 5.9 and Figures 5.10 - 5.11 illustrate the performance results and DET curves obtained for combining the ”five“,

"G", and "M" gestures on one hand and "five", "G", "M", and "Y" gestures on the other hand. When $\beta = 9$ and $\lambda = 3$, we obtain for the ("Five", "G", "M") combination FAR = 1.98% and FRR = 3.41%, and we obtain for the ("Five", "G", "M", "Y") combination FAR = 1.55% and FRR = 2% when $\beta = 11$ and $\lambda = 3$, which is a net improvement in the system performance as the number of gestures increase. However increasing the number of gestures come at the expense of usability. As it should be expected, trade-off must be made between security and usability.

$\boldsymbol{\lambda = 2}$	$\boldsymbol{\beta = 7}$	$\boldsymbol{\beta = 8}$	$\boldsymbol{\beta = 9}$	$\boldsymbol{\beta = 10}$	$\boldsymbol{\beta = 11}$
FAR	39.74	30.83	23.34	16.69	10.71
FRR	0	0	0	0	0.06
$\boldsymbol{\lambda = 3}$	$\boldsymbol{\beta = 7}$	$\boldsymbol{\beta = 8}$	$\boldsymbol{\beta = 9}$	$\boldsymbol{\beta = 10}$	$\boldsymbol{\beta = 11}$
FAR	5.70	3.51	1.98	0.93	0.37
FRR	1.15	2	3.41	7.5	16.76
$\boldsymbol{\lambda = 4}$	$\boldsymbol{\beta = 7}$	$\boldsymbol{\beta = 8}$	$\boldsymbol{\beta = 9}$	$\boldsymbol{\beta = 10}$	$\boldsymbol{\beta = 11}$
FAR	0.05	0.01	0	0	0
FRR	56.28	73.84	86.52	93.65	97.80

Table 5.8: The recognition performance for "Five", "G", and "M" gestures combined.

$\boldsymbol{\lambda = 2}$	$\boldsymbol{\beta = 9}$	$\boldsymbol{\beta = 10}$	$\boldsymbol{\beta = 11}$	$\boldsymbol{\beta = 12}$	$\boldsymbol{\beta = 13}$
FAR	42.34	34.68	27.26	20.92	15.20
FRR	0	0	0	0	0
$\boldsymbol{\lambda = 3}$	$\boldsymbol{\beta = 9}$	$\boldsymbol{\beta = 10}$	$\boldsymbol{\beta = 11}$	$\boldsymbol{\beta = 12}$	$\boldsymbol{\beta = 13}$
FAR	5.0	2.96	1.55	0.84	0.44
FRR	0.6	1.15	2	3.29	6.52
$\boldsymbol{\lambda = 4}$	$\boldsymbol{\beta = 9}$	$\boldsymbol{\beta = 10}$	$\boldsymbol{\beta = 11}$	$\boldsymbol{\beta = 12}$	$\boldsymbol{\beta = 13}$
FAR	0	0	0	0	0
FRR	57.43	69.45	80.97	89.26	93.23

Table 5.9: The recognition performance for "Five", "G", "M", and "Y" gestures combined.

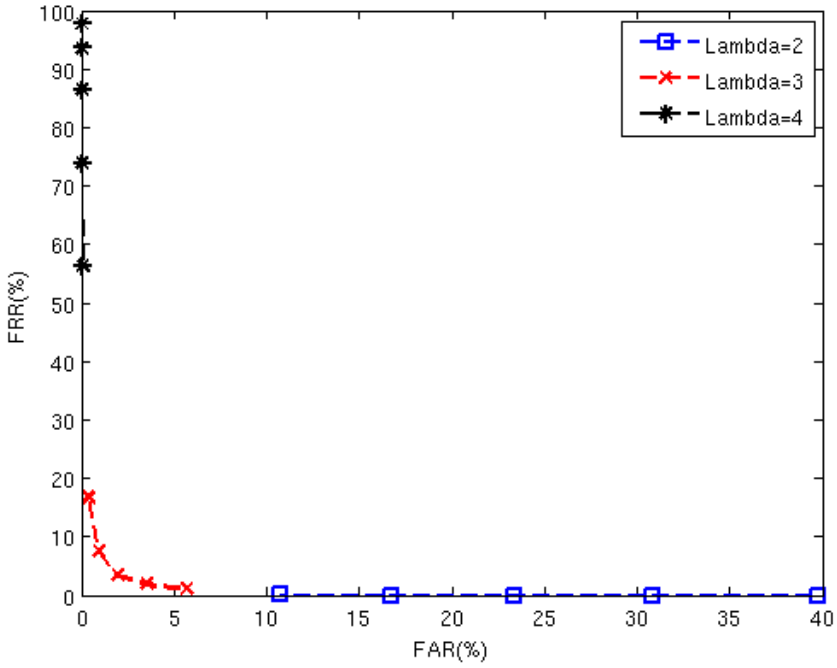


Figure 5.10: The DET curve for the “Five“, ”G“, and ”M“ gesture combined.

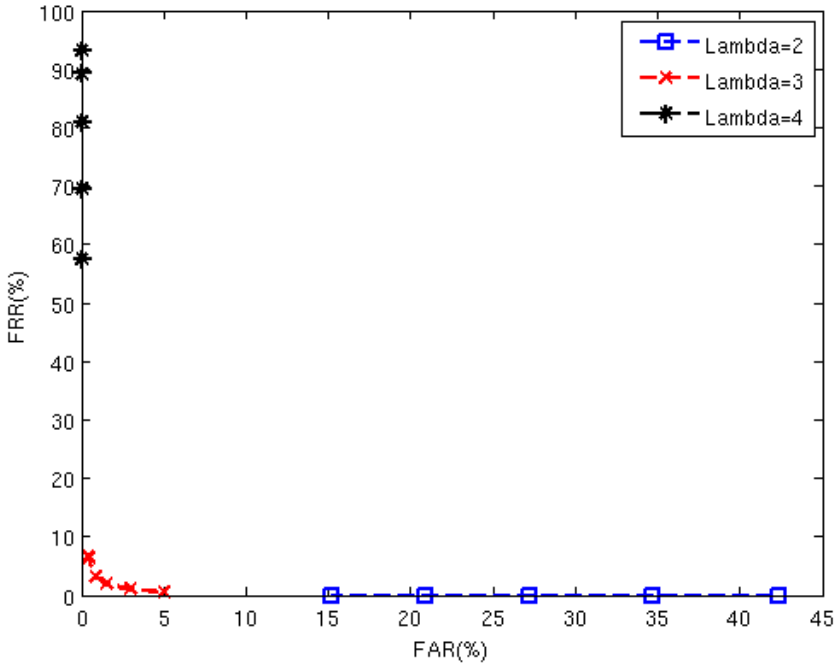


Figure 5.11: The DET curve for the “Five“, ”G“, ”M“, and ”Y“ gesture combined.

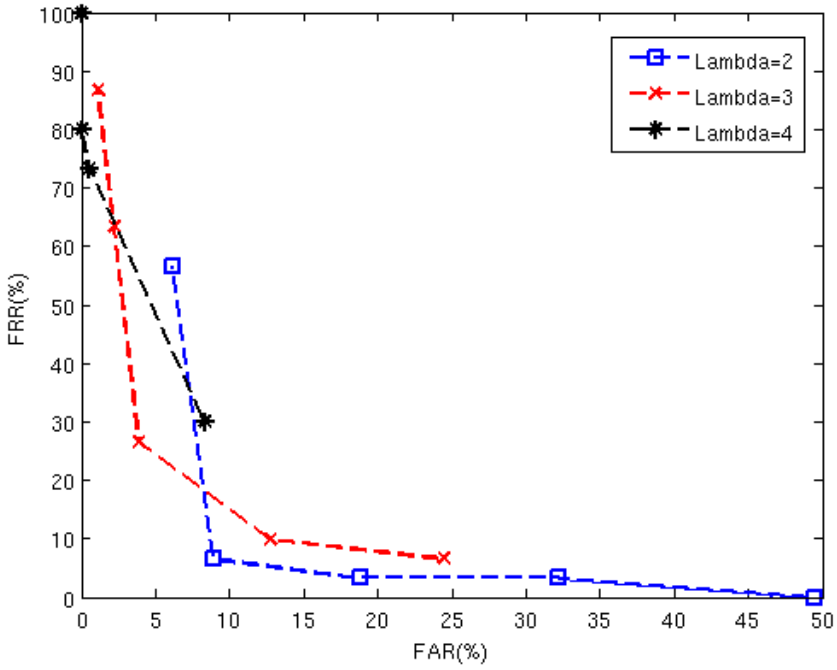
5.6 Follow-up Experiment

Furthermore, we conducted a follow-up experiment to study the effect of the early comparison step which provides visual feedback to the user. The follow-up experiment involved six users who were asked to provide five additional replicas per gesture when the early comparison (visual feedback) is turned off (on top of the ones provided in the main experiment). The main goal from such an experiment is to study the effect of the visual feedback. As we mentioned earlier the visual feedback is based on a trivial comparison which only compares the gestures from the shape perspective. We analyzed the collected data using the same method as in the main experiment. Tables 5.8 - 5.9 and Figures 5.12 - 5.13 illustrate the performance results and DET curves when visual feedback is not provided.

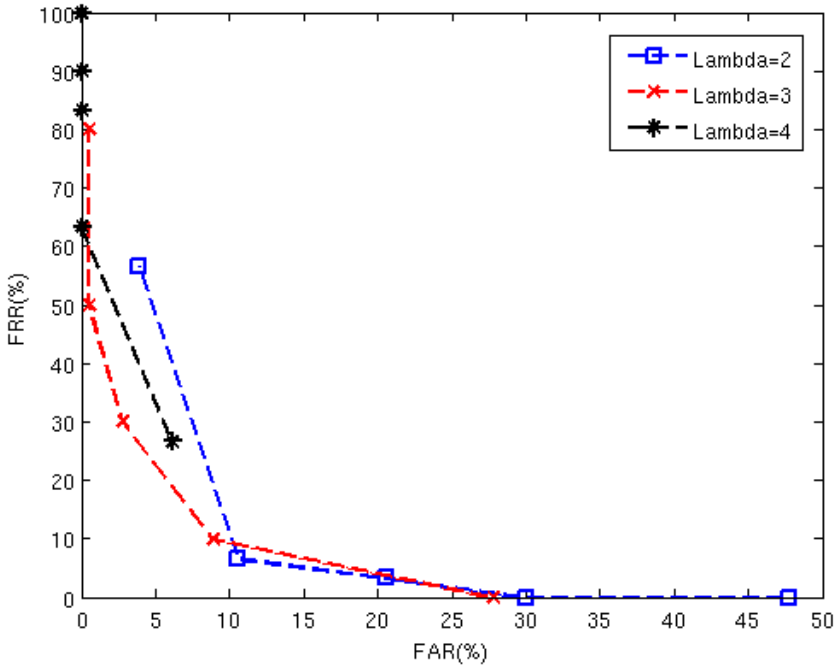
$\lambda = 2$	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
FAR	49.44	32.22	18.88	8.88	6.11
FRR	0	3.33	3.33	6.66	56.66
FAR(Visual FB OFF)	47.77	30	20.55	10.55	3.88
FRR(Visual FB OFF)	0	0	3.33	6.66	56.66
$\lambda = 3$	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
FAR	24.44	12.77	3.88	2.22	1.11
FRR	6.66	10	26.66	63.33	86.66
FAR(Visual FB OFF)	27.77	8.88	2.77	0.55	0.55
FRR(Visual FB OFF)	0	10	30	50	80
$\lambda = 4$	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
FAR	8.33	0.55	0	0	0
FRR	30	73.33	80	100	100
FAR(Visual FB OFF)	6.11	0	0	0	0
FRR(Visual FB OFF)	26.66	63.33	83.33	90	100

Table 5.10: The recognition performance for the "Z" gesture.

As we can see from the results the visual feedback effect is negligible. Intuitively, this makes sense as the comparison is very permissive and entirely based on the shape of the gesture as opposed to their dynamics.

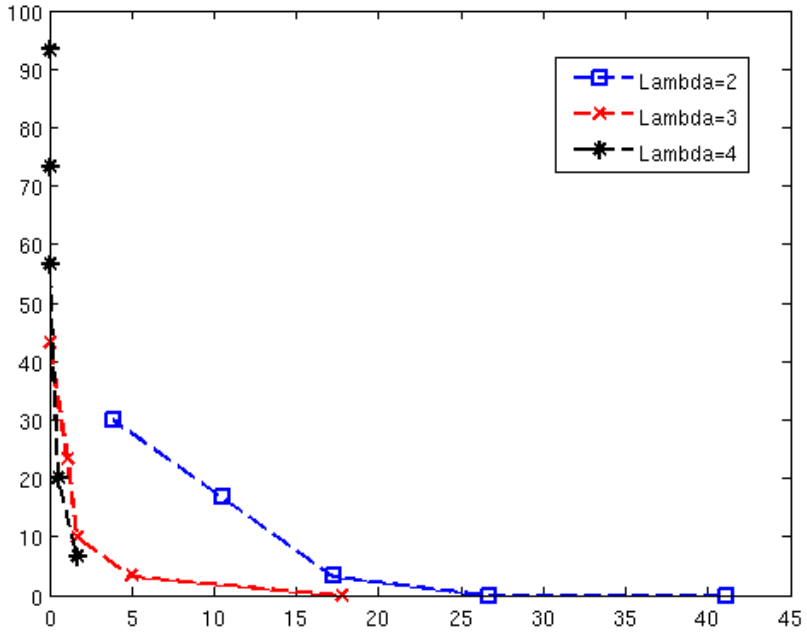


(a) DET curve of the "Z" gesture when visual feedback is provided.

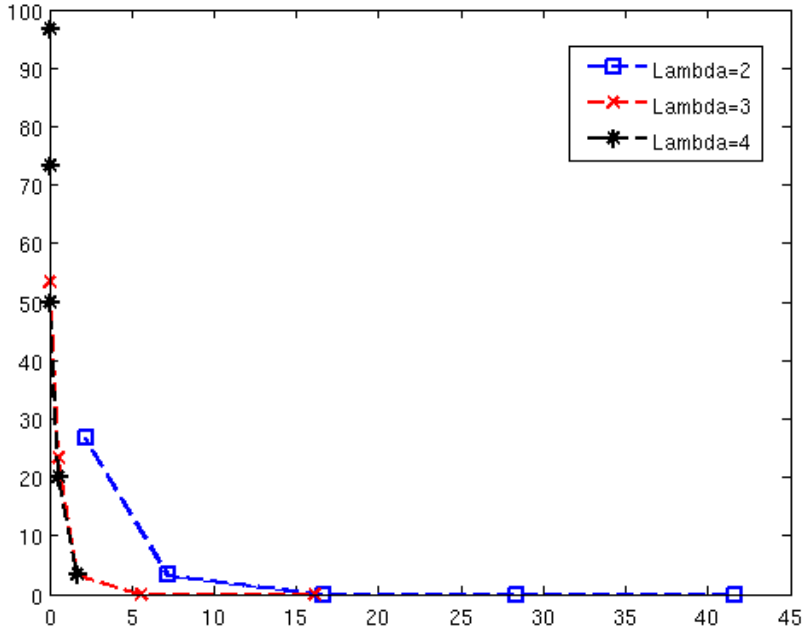


(b) DET curve of the "Z" gesture when visual feedback is not provided.

Figure 5.12: Visual feedback effect on the "Z" gesture.



(a) DET curve of the "M" gesture when visual feedback is provided.



(b) DET curve of the "M" gesture when visual feedback is not provided.

Figure 5.13: Visual feedback effect on the "M" gesture.

$\lambda = 2$	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
FAR	41.11	26.66	17.22	10.55	3.88
FRR	0	0	3.33	16.66	30
FAR(Visual FB OFF)	41.66	28.33	16.66	7.22	2.22
FRR(Visual FB OFF)	0	0	0	3.33	26.66
$\lambda = 3$	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
FAR	17.77	5	1.66	1.11	0
FRR	0	3.33	10	23.33	43.33
FAR(Visual FB OFF)	16.11	5.55	1.66	0.55	0
FRR(Visual FB OFF)	0	0	3.33	23.33	53.33
$\lambda = 4$	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$	$\beta = 5$
FAR	1.66	0.55	0	0	0
FRR	6.66	20	56.66	73.33	93.33
FAR(Visual FB OFF)	1.66	0.55	0	0	0
FRR(Visual FB OFF)	3.33	20	50	73.33	96.66

Table 5.11: The recognition performance for the "M" gesture.

5.7 Observations

From the obtained results, the following can be observed:

- The more curves and angles the gesture has, the more human motor-control is required, which results in more intrinsic behavior to be imposed. This means that the gestures should be carefully chosen so that they include more curves and angles as opposed to straight lines. Actually to some extent, our findings confirms the results of the HCI researchers covered in the literature review. The less curves and angles the gesture have, the more likely humans tend to draw it in the same amount of time. That being said, still there is a fundamental difference, during the HCI experiments the participants were visually guided by the target they want to click on as opposed to our case, which is considered free sketching. In addition, our work does not only look at the production time but also the dynamics and shape aspects of the gestures.

- Although combining several gestures increases the challenge of the test session

on both the legitimate and non-legitimate users, the results show that both the FAR and FRR decrease. On the other hand, combining gestures increase the length of the test session, as the average production time of a gesture is 1.5 second. Table 5.12 provides a breakdown of the duration of the different components of a test session according to the number of gestures combined. The time between gestures during verification which is set to 300 msec can be lowered if necessary; note that the time between each gesture replication during enrollment was set to 3 sec for the reasons explained earlier. The session length for four gestures is 40.7 sec, which is in the order of magnitude of the time involved in using existing second factor authentication schemes such as graphical passwords, challenge questions and hardware authentication tokens.

Number of Gestures Combined	**Gesture Production Time (sec.)**	**Time between Gestures (sec.)**[1]	**Gesture Verification Time (sec.)**[2]	**Total Time (sec.)**
1	7.5	1.2	1.25	9.95
2	15	2.7	2.5	20.2
3	22.5	4.2	3.75	30.45
4	30	5.7	5	40.7

Table 5.12: Length of test session for the different gesture combinations.

- The effect of the early comparison step which provides visual feedback to the users is negligible.

- By combining four gestures, we achieve a FAR = 1.55% and FRR = 2%, which is encouraging and suggest that mouse dynamics be used as a second factor for static authentication at login time.

[1]Pause time performed by the data capture module between each gesture replication drawn by the user.

[2]Time taken by the modular neural network to recognize the gesture and output decision.

5.8 Summary

In this chapter, we have presented the experimental evaluation of our framework involving 41 users. The exact method and the apparatus for the experiment were explained. The results of the five gestures involved in the main experiment were shown and analyzed. The effect of changing the values of the system variables was studied and clearly the performance of the framework degrades or improves accordingly. In the next chapter we present the overall conclusion, ideas, enhancement of the framework and discuss future work.

Chapter 6

Conclusion and Future Work

6.1 Summary

In this thesis we have highlighted the challenges faced by mouse dynamics biometric technology when it is applied for static authentication and we proposed a new framework based on mouse gesture dynamics which address those challenges. The proposed framework uses a modular design of the LVQ neural network for classification. we conducted an experimental evaluation of the framework involving 41 users, yielding FAR = 9.19% , FRR = 7.74% when only one gesture was used, and FAR = 3.59% , FRR = 3.35% when two gestures were combined, and FAR = 1.98% , FRR = 3.41% when three gestures were combined, and FAR = 1.55% , FRR = 2% when four gestures were combined. Our study clearly appeal for the combination of several gestures (at least three) in order to achieve acceptable performance in static user authentication. However, as expected, trade-off must be made between usability and security, which necessarily mean some limitation on the maximum of gestures that could be used. In any case the results obtained in our study are encouraging and highlight the promise of mouse dynamics for static authentication.

6.2 Future Work

In the future, there are three directions we plan to further our research into. Firstly, we intend to investigate a different decision fusion scheme other than the majority voting. For example, if we know that one of the neural network modules is more accurate than the others, we might investigate a weighted majority voting scheme. The interesting part would be how to scale the weights per user as opposed to a system wide weight.

Secondly, we plan to investigate other classification techniques that might improve the overall performance of our framework. One obvious classification technique would be a technique that utilize the fuzzy systems, for instance, fuzzy-neural classifier or fuzzy decision tree. The main reason a fuzzy system would be more adequate than others is the nature of the behavioral biometrics itself. Generally, humans have commonalities but when it comes to a particular individual humans are not the same. In other words, instead of having an evaluation system that is binary and the thresholds are discrete as in the current work, the system should use confidence ratios. For example, instead of the system output being user X or not (binary), it could be user X with confidence ratio 90% and user Y with confidence ratio 10%.

Furthermore, as the next wave of enhancement in the world of user interface (UI) is the touch technology, nowadays there are many consumer electronics (e.g. smart phones and mobile internet devices) which are entrely based on touch or multi-touch UI. A fundamental aspect of the touch UI consist of using gestures for user interaction, as the users interact with these devices by drawing gestures, we do think that with a minimal modification, our framework can be extended with the possibility of providing not only static authentication but also continuous authentication on such platforms.

Bibliography

[1] X. Cao and S. Zhai, "Modeling human performance of pen stroke gestures," in *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, (New York, NY, USA), pp. 1495–1504, ACM, 2007.

[2] J. R. Vacca, *Computer Forensics: Computer Crime Scene Investigation*. Charles River Media, 2005.

[3] Wiktionay.com, "Biometrics," (http://en.wiktionary.org/wiki/biometrics), October 2008.

[4] Whatis.com, "Biometrics," (http://searchsecurity.techtarget.com), October 2008.

[5] A. A. E. Ahmed and I. Traore, "A new biometric technology based on mouse dynamics," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, pp. 165–179, July/Sept. 2007.

[6] A. A. E. Ahmed and I. Traoré, "System and method for determining a computer user profile from a motion-based input device," No. 10/555408, (Patent Canada), November 2003.

[7] P. Oel, P. Schmidt, and A. Shmitt, "Time prediction of mouse-based cursor movements," in *Proceedings of Joint AFIHM-BCS Conference on Human- Computer Interaction IHM-HCI'2001*, vol. 2, pp. 37–40, September 2001.

[8] T. Whisenand and H. Emurian, "Analysis of cursor movements with a mouse," in *Computers in Human Behavior*, vol. 15, pp. 85–103.

[9] G. Gupta and A. McCabe, "A review of dynamic handwritten signature verification," technical report, James Cook University, Australia, 1997.

[10] S. L. Baird, "Biometrics: Security technology," (http://goliath.ecnext.com/coms2/summary_0199-1533309_ITM), February 2002.

[11] J. Close, "Motorola white paper: An introduction to biometrics.," (www.motorola.com/biometrics), August 2006.

[12] Wikipedia, "Biometric passport," (http://en.wikipedia.org/wiki/Biometric_passport), October 2008.

[13] A. K. Jain and S. Pankanti, "Biometrics systems: Anatomy of performance," in *EICE Transactions Fundamentals*, vol. E84-D, pp. 788–799, 2001.

[14] A. K. Jain, S. C. Dass, and K. Nandakumar, "Soft biometric traits for personal recognition systems," in *Proceedings of International Conference on Biometric Authentication*, (Hong Kong), pp. 731–738, July 2004.

[15] A. K. Jain, S. Pankanti, S. Prabhakar, L. Hong, and A. Ross, "Biometrics: a grand challenge," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2, pp. 935–942, Aug. 23–26, 2004.

[16] P. M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement," *Journal of Experimental Psychology*, vol. 47, no. 6, pp. 381–391, 1954.

[17] P. M. Fitts and J. R. Peterson, "Information capacity of discrete motor responses," *Journal of Experimental Psychology*, vol. 62, pp. 103–112, February 1964.

[18] I. S. MacKenzie, *Movement time prediction in human-computer interfaces*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995.

[19] I. S. MacKenzie and W. Buxton, "Extending fitts' law to two-dimensional tasks," in *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, (New York, NY, USA), pp. 219–226, ACM, 1992.

[20] A. A. E. Ahmed and I. Traoré, "Detecting computer intrusions using behavioral biometrics," in *Proceedings of the Third Annual Conference on Privacy, Security and Trust*, (New Brunswick, Canada), pp. 91–98, October 2005.

[21] A. Nazar, I. Traore, and A. Ahmed, "Inverse biometrics for mouse dynamics," *International Journal of Artificial Intelligence and Pattern Recognition*, vol. 22, pp. 461–495, May 2008.

[22] H. Gamboa and A. Fred, "A behavioral biometric system based on human-computer interaction," in *Conference on Biometric Technology for Human Identification*, vol. 5404, (Orlando, FL, USA), pp. 381–392, April 2004.

[23] M. Pusara and C. E. Brodley, "User re-authentication via mouse movements," in *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, (New York, NY, USA), pp. 1–8, ACM, 2004.

[24] P. Isokoski, "Model for unistroke writing time," in *Proc. ACM CHI Conference on Human Factors in Computing Systems*, pp. 357–364, 2001.

[25] P. Viviani and Flash, "T. minimum-jerk, two-thirds power law, and isochrony: converging approaches to movement planning," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 21, no. 1, pp. 32–53, 1995.

[26] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A. D. Rubin, "The design and analysis of graphical passwords," in *SSYM'99: Proceedings of the 8th conference on USENIX Security Symposium*, (Berkeley, CA, USA), pp. 1–1, USENIX Association, 1999.

[27] A. Broemme, S. Al-Zubi, A. B. Omme, and S. Al-zubi, "Multifactor biometric sketch authentication," in *Proceedings of the First Conference on Biometrics and Electronic Signatures of the GI Working Group BIOSIG*, pp. 81–90, 2003.

[28] S. Bella and C. Palmer, "Personal identifiers in musicains' finger movement dynamics," *Journal of Cognitive Neuroscience*, vol. 18, 2006.

[29] K. Hayashi, E. Okamoto, and M. Mambo, "Proposal of user identification scheme using mouse," in *ICICS '97: Proceedings of the First International Conference on Information and Communication Security*, (London, UK), pp. 144–148, Springer-Verlag, 1997.

[30] A. F. Syukri, E. Okamoto, and M. Mambo, "A user identification system using signature written with mouse," in *ACISP '98: Proceedings of the Third Australasian Conference on Information Security and Privacy*, (London, UK), pp. 403–414, Springer-Verlag, 1998.

[31] R. Plamondon and G. Lorette, "Automatic signature verification system: From theory to practice," *In International Journal of Pattern Recognition and Artificial Intelligence*, vol. 8, no. 3, pp. 795–811, 1994.

[32] J. Brault and R. Plamondon, "A complexity measure of handwritten curves: Modeling of dynamic signature forgery," in *IEEE Tans on Systems, Man, and Cybernetics*, vol. 23, pp. 400–413, March/April 1993.

[33] Pearson, "On lines and planes of closest fit to systems of points in space." Philosophical Magazine 2 (6):559-572, 1901.

[34] J. Shlens, "A tutorial on principal component analysis," tech. rep., Institute for Nonlinear Science, University of California, San Diego, December 2005.

[35] T. Kohonen, *Self-Organizing Maps*, vol. 30 of *Springer Series in Information Sciences*. Springer, third extended ed., 2001.

[36] F. Azam, *Biologically inspired modular neural networks*. PhD thesis, Virginia Tech, 2000.

[37] L. L. Lee, T. Berger, and E. Aviczer, "Reliable online human signature verification systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 643–647, June 1996.