A GAZE-BASED AUTHENTICATION SYSTEM: FROM AUTHENTICATION TO INTRUSION DETECTION

A Thesis

by

ADIL HAMID MALLA

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Tracy Hammond
Committee Members,	James Caverlee
	Joanna Lahey
Head of Department,	Dilma Da Silva

May 2018

Major Subject: Computer Science

ABSTRACT

The use of authentication systems has increased significantly due to the advancement of technology, greater affordability of devices, increased ease of use, and enhanced functionality. These authentication systems help safeguard users' private personal information. There are a plethora of authentication systems based on a variety of inputs such as pins, biometrics, and smart cards. All of these authentication systems experience different threats and attacks. Shoulder surfing is an attack when an intruder tries to look at what a user is inputting on the authentication system either by looking over the shoulder or using the video technology. Pin-based authentication systems are prone to shoulder surfing; e.g. at ATM's or other public places an intruder can shoulder surf what a user is entering as their pin/password. Biometric-based authentication systems are prone to spoofing attacks. Smart Cards can be easily stolen, replicated, or even spoofed. Thus, the goal of this research is to explore, develop, and quantify an alternate authentication system that addresses issues/attacks faced by the most commonly used authentication systems. We do this through the development of a gaze-based authentication system which addresses the problem of shoulder surfing, video analysis attacks, and spoofing attacks by an intruder. Results show an accuracy of 97.5 % and F-measure of 0.97 is achievable while authenticating a user and an accuracy of 89.5 % and F-measure of 0.89 is achievable when attempting to detect an intruder trying to log in using someone else's password.

DEDICATION

This work is dedicated to my brother Arif Hamid Malla, who encouraged me to make difference in this world. To my parents, Abdul Hamid Malla and Mimoona Akhter for always being supportive and understanding. And to my sweet sisters Abida Hamid and Uzma Hamid, who have tolerated me and my orders for all these years. I am humbled and graciously thankful for all your smiles, care, and love. And my faithful gratitude to the Almighty, through Whose grace I exist.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to many people without whom I would have never been able to complete my thesis. First and foremost I would like to thank my adviser, Dr. Tracy Hammond for her constant support, encouragement, and enthusiasm. I would also like to thank my committee members Dr. Joanna Lahey and Dr. James Caverlee for their perspectives and reviews on my work.

Also, I would like to thank whole Sketch Recognition Lab members for their support, feedback, reviews, and encouragement for this work. In particular, I would like to thank Vijay Rajanna for his guidance and Seth Polsley, Josh Cherian for their guidance and insights throughout the process. I would also like to thank the people who participated in the user studies of my thesis work.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professors Dr. Tracy Hammond (advisor) and Dr. James Caverlee of the Department of Computer Science & Engineering and Professor Dr. Joanna Lahey of the Department of Public Service and Administration.

The initial research and development for this thesis was done in collaboration with Vijay Rajanna and Rahul Bhagat. Vijay Rajanna helped in setting up the foundations of using the gaze based authentication based on shape transition which is presented in Section 3.2.2 and 3.3.3. The user studies for gaze based authentication which is present in Section 4.2 and 5.3 was done by Vijay.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

Graduate study was supported by a Research Assistantship from the Department of Computer Science & Engineering at Texas A&M University.

NOMENCLATURE

PDA	Personal Digital Assistants
CCD	Charge Coupled Device
SDK	Software Development Kit
WPF	Windows Presentation Foundation
RMS	Root Mean Square
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
FAR	False Accept Rate
TAR	True Accept Rate
FRR	False Reject Rate
TRR	True Reject Rate
EER	Equal Error Rate
WEKA	Waikato Environment for Knowledge Analysis
NIST	National Institute of Standards and Technology

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION

Authentication systems play an important role in the present-day technological world. All secure systems which include the banking application, websites, mobile applications, mobile devices, etc. are controlled one of authentication, authorization and identification system. Whether you are trying to login to your mobile device or trying to access a lab in your department, or if you are trying to transact using a banking website, all these trivial tasks in this present world are secured by an authentication or identification/authorization system. While your phone may be just an authentication system, logging in to the lab premises is considered as an identification as well as authorization process. The system should be able to know who entered the premises and whether the person unlocking the mobile phone is genuine user. In nutshell, all technology we interact with involves an authentication system in some way or other. Thus a secured authentication [6] is very important in most technological systems, making the security of the passwords to become a domain of paramount interest to researchers.

1.1 Authentications Systems

Authentication is a process of knowing whether the credentials given by the user matches to information stored by the authentication system for that particular user. Different types of authentication systems are available to protect secured systems. The pin or password-based authentication [7] was one of the first authentication systems to be introduced. These authentication systems will take pin or password input and authenticate the user by matching pin or password with one stored in the system [8]. These traditional passwords can be easily shoulder surfed or even guessed by the intruders [9, 10, 11].

A second class of authentication systems is biometrics authentication systems, which consist of some physiological or behavioral trait that is unique to the person [12] [13]. These can include fingerprint [14] [15], face [16], iris [17], palm print, finger vein pattern [18], gait [19], etc. Authentication procedure comprises of two stages. First, where a user enrolls himself using the pin or biometric trait which is stored by the system for that particular user. Second, the user uses the authentication system to get authenticated by using the passwords or biometric trait given during the enrollment. The basic framework/system consists of an acquisition hardware/sensor, which takes these physiological traits as input, extractor which extracts the unique information and a matcher which matches two inputs as extracted features for comparison. Although these authentication systems are highly reliable, they can be easily compromised [20].

1.2 Attacks on Authentication

The number of unauthorized accesses to an authentication system is growing day by day [21]. Securing a software system or application thus requires a secure authentication process. Making sure the authentication process is itself is highly secure is an important and vital task. Successfully bypassing authentication system can lead to severe consequences both in terms of loss of capital as well as private data. Attackers can not only get the access to private information, but can also change, delete, or corrupt important data. Moreover, an attacker can steal their victim's identity, resulting in identity theft and private information being leaked. These attacks has resulted in significant interest in making authentications systems as secure as possible. Such attacks to break into the system can be categorized [22] into various types:

1. Bypass Attacks

2. Brute Force Attacks

3. Session Eavesdropping

4. Shoulder Surfing and Replay Attack

5. Verifier Impersonation

6. KeyLogger Attack

7. Server Side Attack

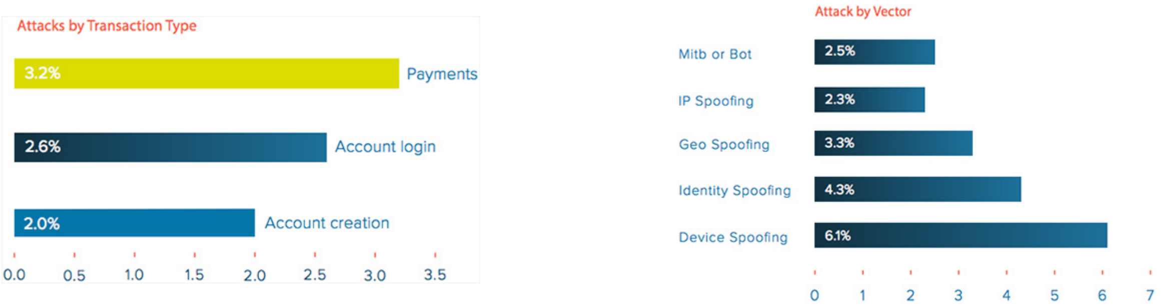
8. Spoofing Attacks

Method	Examples	Properties
What you know?	UserID Password PIN	Many Passwords easy to guess Forgotten
What you have?	Cards Badges Keys	Shared Can be duplicated Lost of Stolen
What you have and what you know?	ATM + PIN	Shared PIN Weak Link
Something unique about user?	Face Fingerprint Iris Voice Finger-Vein	Not Possible to Share Forging Possible (Now) Cannot be lost Stolen (Yes)

Table 1.1: Existing User Authentication Techniques adapted from [5].

1.2.1 Bypass Attacks

Bypass attacks are easiest to carry out for an attacker. These types of attacks occur when the authentication system or device is broken due to some failure or some bug in the code. The general authentication systems which are customized by the security organizations are the favorites for attackers as they have a lot of threat surfaces due to the weak architecture and customization.



(a) Attacks by Transaction Type

(b) Attacks by Vector Type

Figure 1.1: Different attack classification for Q4 2015 and Q1 2016 - Total of 6 billion transactions adapted from [1]

1.2.2 Brute Force

Brute force attacks are an unintelligent way of gaining unauthorized access to a system as they use a trial and error methods to guess the right password. This is a time-consuming process and demands patience. However, brute force attacks have a higher rate of success when the lifetime of the password has no constraints. Since brute force attacks try all possible permutations and combinations of the potential password, the search tree is usually large and can not be solved in linear time.

1.2.3 Session Eavesdropping

Session Eavesdropping occurs when attackers attempt to capture the packets or the signal during the authentication process. The method includes the session takeover attacks where an attacker captures the authentication token and reuses it in the new request for the authentication.

1.2.4 Shoulder Surfing

Shoulder surfing is a direct observation technique in which an attacker looks over the victim's shoulder to get their confidential information [23]. Shoulder-surfing attacks are prevalent in crowded places as the victim is typically unable to recognize a potential attacker. In most of cases, an attacker tries to get the victim's personal identification information to gain unauthorized access to the system. In addition, the attackers can use devices such as long-range binoculars, video cameras, and thermal cameras, etc. to support their attacks.

1.2.5 Spoofing Attacks

Spoofing attacks are one of the more prevalent attacks these days due to the advancement of technology. These type of attacks are more sophisticated and are typically used to attack biometrics authentication systems [24]. Spoofing attacks capture the biometric trait of the authentic user and then replicating the biometric on some other material and use that for authentication. The most common examples of biometric spoofing [25] [26] [27] is fingerprint, Iris, Face, Voice etc. The ease of getting biometric sample of the user makes it more susceptible to spoofing attacks. Moreover, if the biometric trait is lost then the user won't be able to use that modality for authentication as the biometric traits remain constant for the entire lifetime. Also, the biometric modalities can be easily lifted or obtained, like a fingerprint, face, iris, etc which is most common biometric trait used in authentication in the present world [28].

1.2.6 Other Attacks

Verifier impersonation is a basically a man-in-the-middle attack, where an attacker uses information about the authentication system to create a convincing replica in which the make the user puts all his details. These are also called phishing attacks sometimes if the authentication system is over the internet using a web page [29].

KeyLogger attacks [30] a well-known methodology of attack on an authentication system. In this system, a Trojan is used to capture the key pressing details of the keyboard. Server-side [31] attacks are related to attacks on the servers instead of at client side. This is similar to man-in-the-middle attack which exploits internet protocols to get the user's authentication information.

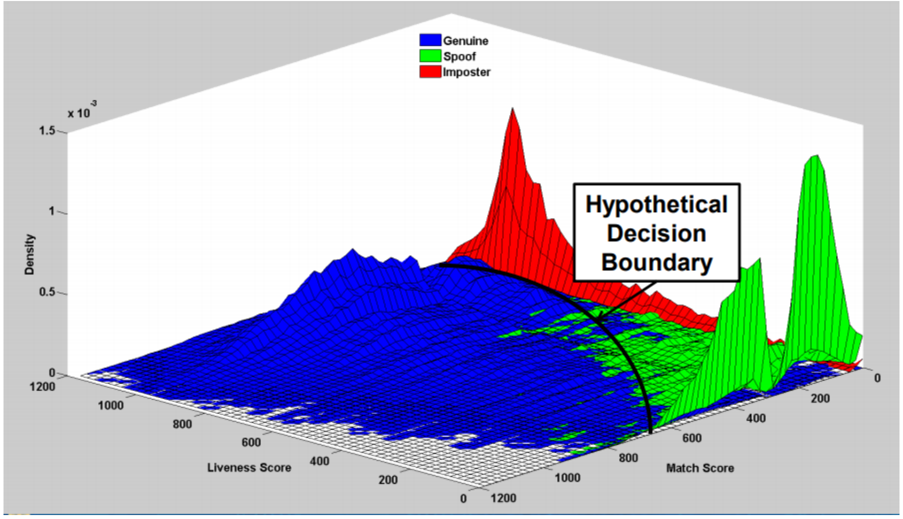


Figure 1.2: Joint distribution of the match and liveness scores for fingerprint authentication on presentation attack dataset by CiTeR, adapted from [2]

Thus, it is clear that any successful authentication system must handle two major problems: it must be able to detect when an impostor is trying to login to the system and when an intruder is making a spoofing attempt. In Figure 1.2, we can clearly see that we have to tackle these problems in an authentication system simultaneously. There have been many attempts to address the various attacks on the authentication systems. The National Institute of Standards and Technology (NIST) funds a number of projects to tackle the problem associated with biometric authentication by conducting the competitions for spoof detection for fingerprint, iris as well as face [32] [33] [34]. Many algorithms have been implemented as well to cope with spoofing attacks on biometrics as well as general authentication systems.

Eye tracking has been used in a plethora of domains like computer science, psychology, marketing, neuroscience, etc [35]. Eye tracking is basically the method/algorithm of tracking an eye using the sensors. This interaction is captured to explore the areas in psychology to check the scene perception, reading, visual search etc. In neuroscience, eye tracking is used to explore the attentional neuroscience, and brain functioning while tracking the eye fixations. In computer science, the eye tracking is used in wide applications [36]. One of the first and foremost uses of eye tracking in computer science is selective systems. Eye tracking can be used as an input system like a mouse pointer to select the menus, text and navigate the applications and websites. Eye tracking has also been used for typing [37], authentication [38, 39], computer-human interactions [40, 41, 42, 43]. The eye tracking methodology provides unique benefits of being fast, reactive, and user convenience [44].

In this work, we address the problem of shoulder surfing, video analysis attacks, spoofing attacks and the problem of compromised passwords and biometric traits through the development of a gaze-based authentication system. We are also exploring the use of eye tracking technology for user authentication and identification as a soft biometric authentication system.

The proposed system seeks to address the problem of shoulder surfing, video analysis attacks, spoofing attacks and the common problem of compromised passwords (intruder detection). In gaze-based authentication, the user will have to follow colored shapes on the screen or move their eye over particular places without giving much information to the intruder. These objects have a randomized position on the screen and follow a randomized path during an animation. A user chooses different colors or positions as a password and authenticates himself/herself by following the path of the shape colors or the particular places. For a successful authentication, the user's gaze path should match the path traced by the colored shape on a given frame. To match the user's gaze path to the path traced by the shape (which acts as the template) we use template matching algorithms. The template matching algorithm [45] matches the predefined template that is at the nearest distance to the path drawn by the gaze. Since all the paths and position of the shapes are randomized there is no way of knowing which shape the user is following in a series of the steps. Moreover, since there is no feedback given by the interface to the user, intruders can't detect or see what users did to authenticate themselves. To handle spoofing attacks, where an imposter knows the pattern of the password or combination of the colors, we will try to recognize that it is an impostor by checking the saccade and eye movement level features.

The organization of the rest of this thesis is as follows. Chapter 2 discusses existing authentication systems and their limitations. Chapter 3 describes the system architecture of the authentication system. Moreover, it will cover the authentication interface as well as the authentication engine. Chapter 4 outlines user studies and what data was collected. Chapter 5 will discuss and analyze the results obtained for various authentication interfaces and their pros and cons. The results will be presented in chronological order of improvement to the system. Chapter 6 describes intruder detection using the gaze features. Chapter 7 will analyze the results obtained for the intrusion detection. Chapter 8 will present the possible future work based on our work. Finally, Chapter 9 presents the conclusion of this thesis work.

2. RELATED WORK

Research on making the authentication systems foolproof from different kinds of attacks has always been of paramount interest to researchers from different domains like biometrics, cryptography, network security, financial institutions, etc. This interest is justified by the potential of these authentication systems which handle the security and integrity of the processes that range from financial transactions to the personalized login to the mobile devices [46, 47, 48, 49, 50]. The widespread adoption of the mobile phone devices and tablets/PDA's has increased the threat surfaces of the authentication systems. The high usage of mobile devices to access private and critical applications in public settings has also increased the demand to have a foolproof system of authentication which cannot be bypassed easily [51, 52, 53]. The increased use of financial application [54] on mobile devices has increased this threat to a new level and provides a new set of challenges to the researchers.

The three broad categorized attacks faced by the authentication systems can be classified as follows:

2.1 Shoulder Surfing Attacks

The significant problem in password-based user authentication is shoulder surfing [55]. There have been several approaches previously suggested beyond the conventional way of using a keyboard and mouse-based input. In this section, we discuss few of the gaze-based solutions for shoulder-surfing.

Kumar et al. [56] provided motivation for using eye tracking to thwart shoulder surfing. Gaze-based password entry makes it even harder to detect user's password if someone shoulder surfs. This paper [56] provides two designs of keyboards (QWERTY and alphabetically ordered) to input password where instead of touching the screen or typing a key, the user looks at each region or character in sequence. The system uses two methods: 1) Gaze + Dwell, and 2) Gaze + Trigger. In the dwell based method, the users gaze on a character for a short time (450 ms - 900 ms). The trigger approach captures the user's gaze only when a dedicated trigger key is pressed. The result of the study conducted shows that Gaze + Dwell method is more appropriate for triggering the key press as it has a lower error rate. Also, the trigger-based approach requires the user to make a coordination of eye and trigger keypress which is difficult to synchronize. The QWERTY keyboard outperforms Alpha-numeric keyboard.

In PassShapes [57], for the authentication procedure, a user draws a shape of the character in a predefined order. The representation of the shapes of characters is represented by characters which can be easily hashed. The drawback of PassShapes is that it does not mitigate the shoulder surfing attacks. Thus an intruder can easily get the password used by a user.

In EyePin [58] the password of a user is still a PIN only the way the PIN or password is inputted in the system is different. EyePin uses gaze to input the numbers in the authentication system. The gaze gestures were used to input the PIN in this system which increased the cognitive load on users.

De Luca et al. [59] presented Eye-PassShapes that uses the eye-tracking based authentication method which improves authentication by combining the above two methods: 1) PassShapes [57], and 2) EyePin[58]. It uses the strokes from PassShapes and EyePIN's eye-tracking approach. A key needs to be kept on pressed while inputting the PassShape. The eye movements of users are analyzed and compared with that of stored PassShape in an enrolled database for authentication. The paper presents various hypotheses to identify the user-interaction speed, error rate and ease-of-use of this method.

ColorPIN [60] is an authentication mechanism that enhances the PIN entry. It uses a combination of digits and colors. The combination of digits and colors help the user to input the password. It presents an effective and easy way of reducing the shoulder surfing attacks by an intruder. Moreover, it also addresses the issue of an intruder using a camera for capturing the interaction with the authentication interface.

Vibrainput [61] uses vibrations as cognitive means to provide security. In Vibrainput, a symbol is mapped to a vibration pattern. Vibration starts as the user touches the screen and stops as user releases. The paper shows 2 prototypes of Vibrainput: 1) Wheel Based- The system shows a graphical wheel comprised of a set of ten symbols and ten different digits. Users can move a symbol to other position by rotating the wheel. The contents of the graphical wheel changes after each selection. 2) A bar based - The system displays ten digits and two vertical bars. The movement of the symbols is done by scrolling the bar. When the user drags the bar, only the symbols change their positions. Vibrainput is generally slow as compared to a 4-digit PIN entry but it is easy to learn and can be easily improved.

Zakaria, Nur Haryani, et al introduced 3 defense techniques on recall of graphical passwords [62] such as Draw-A-Secret [63] and Background Draw-A-Secret [64]. In the Draw-A-Secret system, a password is a picture drawn on N*N grid. The grid is represented by rectangular coordinates (x, y). The user's password is the cells that users cross while moving their gaze on the picture. The 3 techniques are 1) Decoy Strokes 2) Disappearing Strokes 3) Line Snaking Strokes.

Bulling et al. [65] presented a gaze-based authentication system. This authentication system uses the graphical passwords on a single image. The algorithm leverages a computational model that masks the areas of the image that are obvious spots of visual attention. The authors showed that their algorithm is more secure than a standard image-based authentication and gazebased 4-digit pin entry. Specifically, the authors use a bottomup computational model of visual saliency aimed at estimating the parts of the visual scene that are most likely to attract visual attention. To set a password, the authors present to the user an image with all parts of the image masked that are more likely to draw the user's attention.

Luca et al. [58] evaluate three different eye gaze interaction based methods for PIN entry, specifically designed to be immune to the shoulder surfing attacks. The authors also investigated a new approach to use gaze gestures and compare it to the well known classical gaze-interaction based methods. The three modes of gaze interactions presented are 1) Dwell time-based input, 2) Look and shoot, and 3) Gaze gestures.

Roth et al. [66], presented an alternative way of PIN entry method. They called it Cognitive Trapdoor Games. This PIN entry method makes it extremely hard to break into the system. In addition, the authors also introduce an idea of using probabilistic cognitive based trapdoor games. This method fails to attempt to recover the PIN by recording a PIN entry procedure using a camera. The PIN entry process is designed like a gameplay, involving 3 participants: a machine interrogator, a human observer, and a human oracle. Both the interrogator and the Oracle receive a key from a dealer. The human observer will observe the oracle as they try to log-in. The oracle authenticates herself to the system by answering questions presented by the system. At each step, the Oracle is presented with two partitions of colored black and white and must input in which partition the current PIN digit is in. Various other works has been done to thwart the shoulder surfing attacks that include the work by Rajanna et al. [38]

Some of the common issues among all these solutions are that they are inaccurate, induce cognitive load, and can't be adapted to various scenarios. Hence, a system that is accurate, induce no cognitive load, and can easily be adapted to multiple scenarios still remain unsolved.

2.2 Spoofing Authentication Systems

A spoofing attack is a situation in which an intruder or malware successfully pretends as a legitimate/genuine user by falsifying data required for authentication, thereby gaining an illegitimate access to the system secured by the authentication system. There has been a lot of effort towards addressing the spoofing attacks on various kinds of authentication systems [67, 68, 69, 70] in general and biometric authentication and identification systems in specific [25, 71, 72]. The biometric authentication is more susceptible to spoofing attacks as compared to the other authentication devices due to material nature of the items used for authentication. These physiological traits can be falsified, like fingerprint [73, 32], iris [33], face [74, 75] or finger-vein etc. A password cannot be falsified, as an exact password is needed to crack the system by the intruder, which can be easily obtained by doing the brute force attacks or shoulder surfing attacks. There are plethora of the techniques to classify the spoof fingerprint [76, 77, 78, 79, 34], iris [78, 80, 81] or face [82, 83, 84] from the live ones. The sophistication of the spoofing techniques is enhancing day by day with the advancement of the technology, so constant research efforts are needed for making the authentication system foolproof to the spoofing attacks [85, 86, 87, 88, 89].

Since these problems are still not addressed completely, we need an authentication system that users can easily use without having the possibility of shoulder surfing attack possible in working condition and the forging or spoofing should be reduced to the minimal. Our system tries to address all these problems viz. shoulder surfing, video analysis attacks and spoofing attacks. Our system works on the principle of gaze movements to do the authentication. Moreover, we use the movement features to detect the spoofing of the eye movements by some other person. Furthermore, we want to show this authentication system is not limited to the controlled setting but can be used in real-time systems.

3. THE GAZE BASED AUTHENTICATION SYSTEM: BACKGROUND

Gaze-based authentication is not a new field, but as was discussed in chapter 2.1 gaze-based authentication systems face many problems including inaccuracy, inducing cognitive load, and being inflexible to various real-life scenarios. The terms "gaze" and "eye movements" mean the same and will be used interchangeably in the text, although we prefer to use gaze more.

Our system consists of three subsystems:

1. Gaze Tracking Module or Eye Tracker.

2. The User Interface.

3. The Authentication Engine.

A full working model of the system emulated on the computer screen is shown in figure 3.1.

3.1 The Gaze Tracking Modules

The Gaze tracking modules are basically the sensors used to perceive and detect the eye movements of a user [90, 91, 92]. The modern Gaze trackers are based on the Infra-Red light source based sensors [93]. The tracker has the light source which illuminates the eyes and the CCD camera to capture the illuminated eyes to detect the pupil of the eye. The localization of the pupil is done using the Daugman's Method [94, 95]. Once both eyes or a single eye is detected, the system translates the distance of the person and angle to detect the exact point on the screen where the user is looking [96]. These gaze trackers use a simple calibration for making sure the support for displays of different sizes and resolutions.

There are various types of gaze tracker sensors available in the market. We chose one which resembles the basic commodity level sensor which will eventually find a place in real systems, consumer electronics, homes, offices and other electronic devices. Moreover, the placement and the technology of these sensors also vary. Some are head mounted, while others come with chin

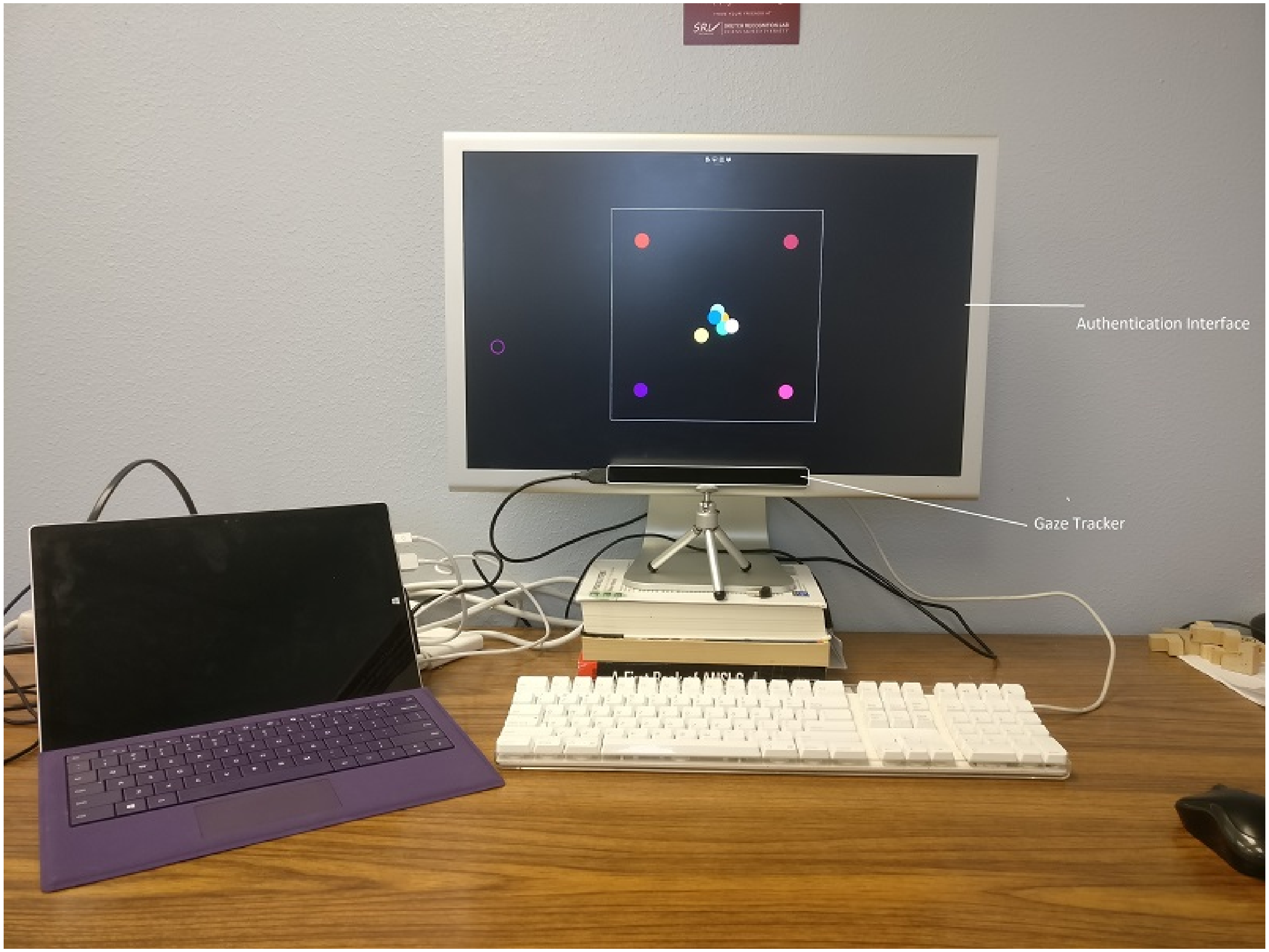


Figure 3.1: Our Gaze-based authentication system showing the authentication interface and gaze tracking module.

rests to make the environment very restricted [35]. We chose the Eye-Tribe tracker [1] for its wide support in terms of SDK, no chin rest, and being economical. The Eye-Tribe tracker is a table mounted eye/gaze tracking sensor that provides the location of the user's gaze on the screen. In addition to this, it also includes the pupil size of each eye. Figure 3.1 shows the table mounted Eye-Tribe Gaze tracking device we used in our study.

3.2 The User Interface

As mentioned earlier, cognitive load, inaccuracy, and adapting to real-life scenarios make gaze tracking-based authentication systems limited in their capabilities. Our research wanted to address these issues and make general authentication systems resilient to shoulder surfing attacks as well as put less cognitive load on the user using the authentication system based on gaze. We researched many aspects of gaze based authentication systems and decided to make the authentication system highly randomized to counter the shoulder surfing attacks and also, accurate and easy to use. We present different gaze based authentication interfaces, and a user can choose an interface based on requirements. The different interfaces are trade-offs between the total authentication time needed by the system and level of authentication security necessary. The two are inversely proportional to each other. Further, we will discuss the dynamics of each interface system and the trade-off of choices made.

3.2.1 Randomized Simple Shape Moving Interface

Randomized simple shape interface is the first interface we designed after a literature review. Since the gaze-based pin faced the shoulder surfing attacks, we came up a concept where we will use circular shapes of different colors to represent a unit of the password. Each colored circle will have a distinct location on the interface that will be randomized for each authentication instance i.e. when a user starts to authenticate using our system. As you can see from the Figure 3.2 below, the five colored shapes represent the units of the password, and the one on top is a circle for gaze movement.

[1] theeyetribe.com [last accessed: 1st Feb 2018]

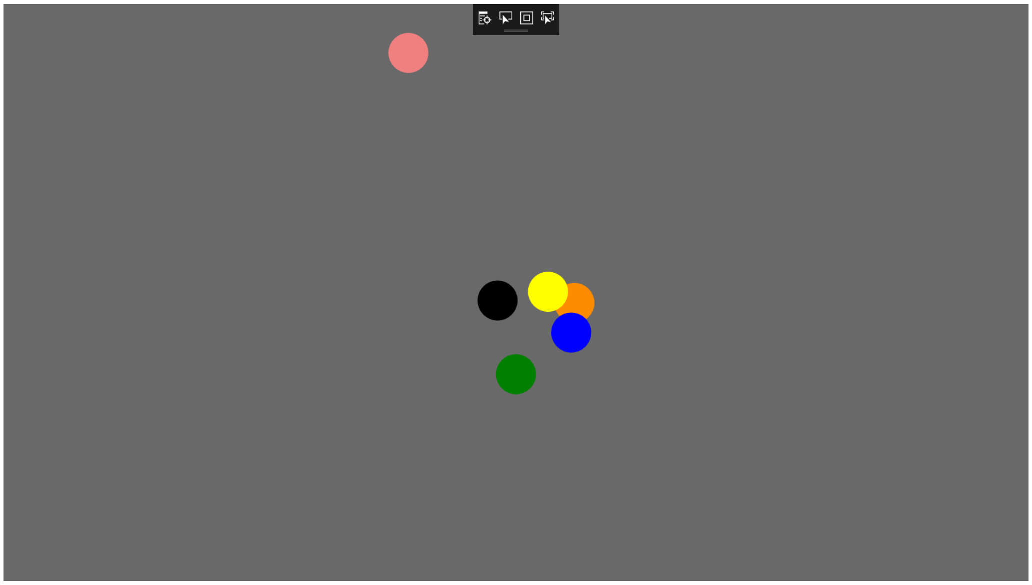
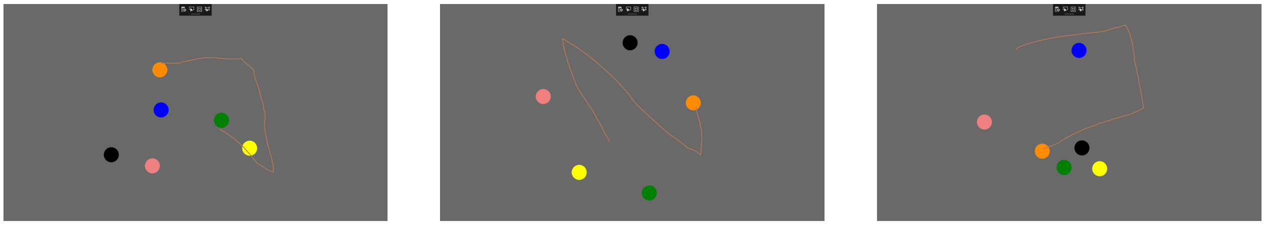


Figure 3.2: A simple randomized interface with 5 colored circles and gaze pointer.

The users can select the combination of the colored circles as password and then during the authentication phase the user has to follow colored shapes selected during the enrollment phase. In this interface, the user can select up to five colored shapes as the password. The following figure 3.3 shows the three animations when a user selects three colored shapes as their password.



(a) Password animation frame 1 - Orange followed

(b) Password animation frame 2 - Yellow followed

(c) Password animation frame 3- Blue followed

Figure 3.3: A full authentication procedure where the password is three colored circles. Orange, Yellow and Blue

This interface is simplistic in terms of the authentication procedure which we will talk about in 4.1. The number of the colored circles shown on the interface can be increased to any number with different colors. Having said that, as we increase the number of colored circles, the memorability of the password becomes difficult. Also, since all of the colored circles move, it can be tough to stay concentrated on the movement of the colored shapes which the user has to follow.

3.2.2 Static-Dynamic Interface

To make the authentication system highly dynamic and compare its authentication accuracy as well as resilience to shoulder surfing attacks, we made the number of the colored circles present on the interface fixed as 10, equal to the number of the digits on the keypad of an ATM or general pin based password keypad. The randomness factor of both the initial location as well as the traversing the path during the animation is also highly randomized using a randomization algorithm which we will discuss in section 3.3.1. We enhanced the randomized interface to develop the static-dynamic interface for two reasons.

1. Although the dynamic interface with ten (10) moving colored circles introduces enough randomness to prevent both shoulder surfing and video analysis attacks, a few users were overwhelmed by the visual cluttering of the interface.

2. Second, since all the 10 circles move within a space of 800×800 pixels, two random paths might be similar which again leads to recognition failures that negatively affect the accuracy. The reason for choosing the 800×800 pixels will be presented in section 3.2.3.

Considering these factors, we designed the static-dynamic authentication interface in such a way that only half of the actual colored circles were dynamic and rest half were static. However, a dynamic (moving) circle on the current animation can continue to be a dynamic circle or become a static circle on the subsequent animation; the same is true for a static circle. Hence while authenticating, the user mostly ends up following a few dynamic circles and focusing on a few static circles. Since random placement of the static circles may bring two circles closer and result in recognition errors, we fixed the locations for static circles as shown in Figure 3.4 along the virtual circular boundary at angles 45°, 135°, 225°, 315°, and at the center of the rectangle.

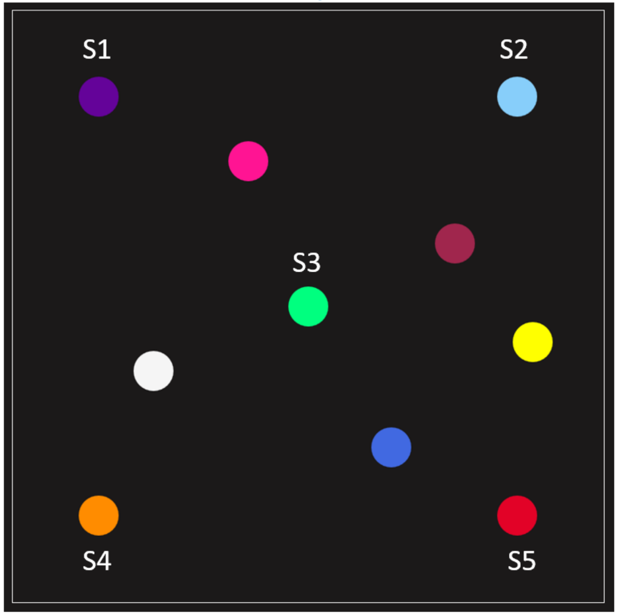


Figure 3.4: 10 colored circles in static-dynamic authentication interface.

3.2.3 Enrollment Interface

The interface is used by the user for the enrollment procedure. The user inputs the name and the password selection which depends on the authentication interface used. If a user wants to use the Static-Dynamic Interface or the Dynamic Interface, then the user will be prompted to select four colors for four different animations which will be used for the authentication process. As shown in figure 3.5, the user for the enrollment procedure has to provide the details like UserID and the four color selection, which will act as the password for the Static-Dynamic or Dynamic Interface. For the case of the shape-based interface the four rows seen in figure 3.5, are replaced by the shapes present in the system. The selection of the shapes is done similar to that of colored circles i.e. one per animation from each row.

The authentication interface selection is the direct function of the screen size as well as the security level needed except the last interface which is based on shapes instead of just colors. Moreover, the different authentication interfaces provide the flexibility to the user to select from different choices. The interface dimensions were chosen to be 800 × 800 pixels to resemble the

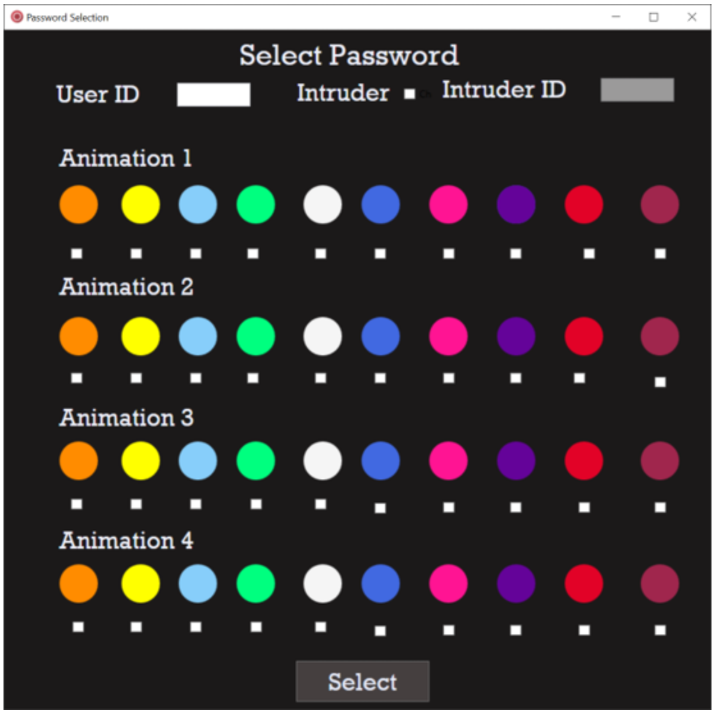


Figure 3.5: Enrollment interface for static-dynamic and dynamic authentication procedures.

average screens of the target devices where we will want to use this authentication system and procedure. We did some research regarding the average sizes of ATM Screens, kiosks at airports, laptop screens and mobile and PDA devices in general with average to small screens. From the survey of ATM machines and websites [97], we came to know the general dimensions of these screens is in the range of 8 inches to 15 inches. We took the median of the screen size and came up with the number 11.5 inches, which roughly translates to 800×800 pixels on a screen with resolution 1920×1200 pixels.

3.3 Authentication Engine

The authentication engine is the backbone of our proposed gaze based authentication system. In our research project, we used the authentication engine based in C# code with windows WPF application to have easy integration of the gaze tracking module and the interface for proof of concept and full system emulation. The authentication handles interaction with both of the above-mentioned components of the authentication systems. Moreover, the authentication engine handles all of the algorithms for data storage, enrollment procedure, template generation, template matching, user authentication, and user identification. The intruder detection is implemented offline based on the data obtained using the Static-Dynamic Interface system. The data is pre-processed using the python routine using pandas and other libraries. The analysis and model building for each user is done using WEKA Data Mining Toolkit [98] and scikit-learn (Python library). The main components of the authentication engine will be described in the below sections. The randomness in the system is generated by the random point generation algorithm and the matching is done using a traditional template-matching algorithm.

3.3.1 Random Point Generation Algorithm

This part of the thesis is the novel component of the work on top of the already present gaze-based authentication system. As we have seen in 2, most gaze-based authentication systems have predefined shapes and paths for the animation, so every time a user starts the authentication procedure the shapes and other components on the screen remain at the same location and makes same movements, thus making the possibility of video analysis attacks very obvious. To overcome this problem of shoulder surfing attacks we developed a randomized positioning algorithm which randomizes the position of each colored circle or shape on the authentication interface. The obvious answer is to randomize the location of the shapes on the interface i.e. is within the region of interest. But a common problem faced by these randomization algorithms is that there is a high collision rate of the points generated in the subsequent trials which reduces the effectiveness of having the shapes highly randomized. To avoid this situation we introduce the concept of uniform distribution of the points on the interface surface. This algorithm makes the possibility of the collision very low, making the point distribution highly randomized. We generate a n number of points within the authentication interface area using the famous method described by Leon-Garcia et al. [99]. We evaluate the joint probability distribution of the points in the given area described by the circle of radius R_c and the random variables $\mathbf{X}$ and $\mathbf{Y}$ representing the point distribution in the given area. The points which show the uniform joint probability density function(pdf) is given by equation 3.1.

$$f_{X,Y}(x,y) = \begin{cases} \frac{1}{A} = \frac{1}{\pi R_c^2} & x^2 + y^2 \leq R_c^2 \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

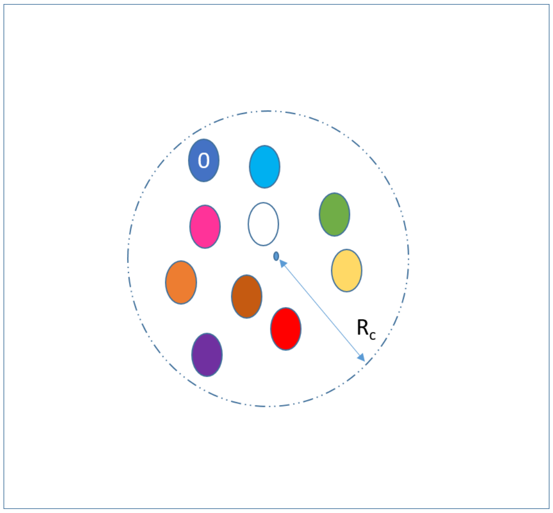


Figure 3.6: The distribution of the points using equation 3.1 for initial positioning and path points.

This algorithm is translated in the code by the number of points required as input and the radius of the circle. The output is the n number of points which are uniformly distributed in the circle of radius given as input. In the given figure 3.6 the randomized point algorithm is used for two purposes.

3.3.1.1 Starting Point Generation Algorithm

Starting point generation algorithm makes sure that the initial position of the colored circles or shapes is randomized at each authentication attempt by the user. As we can see in figure 3.6, where we are just showing one colored circle, the initial placement of the colored circle/shape is placed within a virtual circle of radius equal to one-third of the width of the authentication interface. The reason behind having this kind of setup is to make sure all the colored circles or shapes are initially placed near each other so that video analysis attacks won't help any hacker to pinpoint the location on the screen where the user is looking at the start of the authentication process. Since all the circles will be near each other, the attacker will be confused by the density of the circles initially placed on the interface.

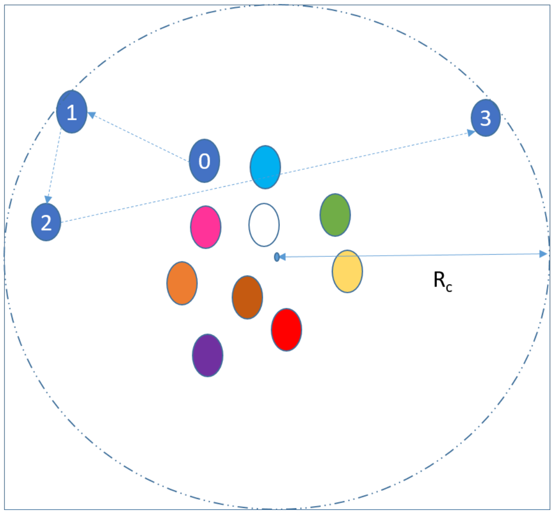


Figure 3.7: The distribution of the points using equation 3.1 for path generation.

3.3.2 Animation Path and Template Generation Algorithm

The above point generation algorithm is applied to get a $n = 10$ number of points for the initial positions. Once the initial positions are obtained, we need to get the animation path of these colored circles or shapes. We use the same point generation algorithm to generate a set of points for each shape. Here, the number of the points are three for each shape and the value of the radius is increased to half the width of the authentication interface ($R_c = 400$ pixels) to make the circle cover almost all the portions of the authentication interface. Based on the authentication interface dimensions, duration of the animation, and the frequency of the gaze tracker, we have established empirically that the template path should be made up of 300 points that are equally distributed along its path relative to the length of each line segment. The template path generation for each colored circle or shape is generated using a curve fitting algorithm. While there are various curve fitting algorithms [100], only two work for our application, Beizer Curve Fitting [101] and

Straight Line fitting [102]. These algorithms were used to make the intermediate points to fit the curve. After applying one of these algorithms we came up with the 300 intermediate points to cover the animations as discussed earlier.

3.3.3 Template Matching Algorithm

The matching algorithm we use in our system is based on template matching algorithms [103]. We match the template path and the users scan-path obtained during the authentication animations. If the user followed green and during the enrollment, the user has selected green colored circle then the first unit of the password is correct. The template matching algorithm is based on the root mean square distance between the scan-path of the user and all the template paths of the shapes. The template path with which the RMS distance is lowest is considered the entered unit (color/shape) of the password from the user. The basic crux of the matching algorithm is taken from the $1 Algorithm [104].

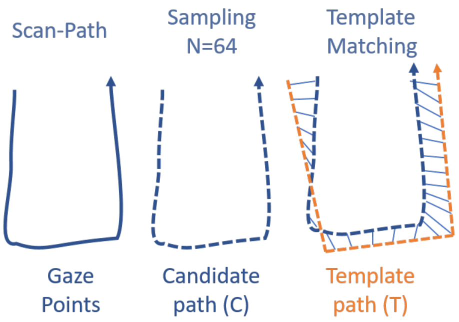


Figure 3.8: The actual and sampled scan path and matching the sampled scan path to the template of a colored circle/shape, adapted from [3].

The two phases of the template matching are Sampling and Actual Matching. In sampling, we downsize the template path as well as the candidate scan path. Sampling makes sure irregular spikes or jittery eye movements are removed before matching the two templates. Moreover, the sampling also helps in reducing the computational complexity and the latency of the matching algorithm. The matching algorithm is used to compute the score which is averaged root mean square distance between the template paths and the user scan path. An equation 3.2 is used to calculate the average RMS distance of the template path from the user scan path when down sampled to N. The matching accuracy is inversely proportional to the score obtained. That is the higher the score, the less it matches. Hence, we select the path of the colored circle or shape, with which the value of score is minimum.

$$\Delta DT_{Color/Shape} = \frac{\sum\limits_{p=1}^{N} \sqrt{(C[p]_x - T[p]_x)^2 + (C[p]_y - T[p]_y)^2}}{N} \tag{3.2}$$

where p is a point on path, C - candidate path, T - template path, and $\Delta DT_{Color/Shape}$ - average distance to template of particular color/shape.

To detect the colored circle or shape the user followed we use equation 3.3 to get the color or shape the user has followed.

$$MatchedColor/Shape = \min_{\forall s \in SColors} \Delta DT_s \tag{3.3}$$

where $SColors$ is a set of all shapes/colors on authentication interface and $\Delta DT_{Color/Shape}$ - average distance to template of particular color/shape.

4. DATA COLLECTION

To test our proposed gaze based authentication system for recognition accuracy, data needed to be collected from users using our authentication systems for both enrollment as well as authentication. The user study was conducted for each authentication interface. This section describes the procedure followed for data collection for each authentication interface separately. Section 5 will talk about the results, evaluation and other analysis of the authentication and recognition accuracy.Moreover there will be a section to analyze the hacking studies done on our system to prove the resilience to shoulder-surfing and video analysis attacks.

The user studies were based on human users using gaze based authentication system proposed in this work. Texas A&M University needs to get the consent from the users and the study should be authorized by Institutional Review Board (IRB) of university for conducting the human user study. This research project was authorized to do human user study under the IRB number IRB2015-0529D.

4.1 Randomized Simple Shape Moving Interface Based Authentication User Study

In this user study, 15 participants (11 male, 4 female), the ages ranging from 18 to 25 (mean age=21.65 years) were asked to use our authentication system. The users were asked to use the enrollment interface to select the three colored circle based password out of five as shown in figure 3.2. After the password selection, the users were asked to use the authentication interface and gaze tracking module to get authenticated. The authentication procedure is simple, the user has three unit (colored circle based) password. On each animation, the user has to follow the colored shape which has same color they have selected during the enrollment phase according to sequence. Once the user followed all the shapes, the template matching is done using the algorithm mentioned in section 3.3.3. The template matching algorithm provides the actual colored circles/shapes followed by a user and the authentication engine matches it with the password selected by that user during enrollment phase. If they match correctly then the user is authenticated and authorized otherwise the user is denied the access. Although the interface is extensible to 12 colored circles, we wanted to see how our algorithms worked and how good was template matching on the dynamic moving colored shapes. The results will be discussed in the later section 5.1.

4.2 Dynamic Authentication Interface based Authentication User Study

After maturing the user interface of proposed gaze based authentication system with static-dynamic interface, we wanted to compare the authentication accuracy and recognition accuracy of algorithm with that of gaze based pin authentication system and previous proposed simple interface. In this study, data were collected from 20 participants (16 male and 4 female). The mean age of the participants was 23.15 years. The participants were asked to enter gaze based passwords for both the dynamic as well as static-dynamic interfaces. The interface shown to the participant was randomized between dynamic and static-dynamic interface to make the presentation of the interfaces balanced to the participants. The dynamic and static-dynamic interface were evaluated under two animation speeds. The animation time of each animation in the two interfaces were differed to check the authentication accuracy and the cognitive load on the participants. Both the cases of system's authentication rates were evaluated by making the users to try the actual passwords as well as false passwords. The two cases helped in achieving the true accept rate and false accept rate.

Participants were also asked about the ease of use of interface and authentication system as well as the amount of the cognitive load they experienced while doing the user study. Moreover, the participants were asked about the password memorability and any other issues faced by them during the use of the authentication interface in specific and authentication system based on the gaze tracking in general.

5. AUTHENTICATION RESULTS BASED ON TEMPLATE MATCHING

Having collected the data for each authentication interface, we analyzed both the recognition as well as the authentication accuracy. In this section, we describe all the results associated with the accuracy of different authentication interfaces. We will describe all results and analyze those results. The recognition algorithm is already mentioned in section 3.3.3. The enrollment uses the interface mentioned in section 3.2.3. Moreover, the authentication, as well as recognition, is done on the fly and we note the results of each recognition attempt. We will be reporting the accuracy [105, 106], F-measure [107], and other error rates [108]. True Accept Rate (TAR) is the percentage of the time our authentication system correctly verifies a true claim of identity. E.g, If I am Adil and I claim to be Adil, then the system verifies my claim. True Reject Rate (TRR) is the percentage of times our authentication system correctly rejects a false claim of identity. E.g, If I am Adil and claim to be Dr. Hammond and the system rejects the claim as it should. False Accept Rate (FAR) is the percentage of the time our authentication system incorrectly verifies a false claim of identity. E.g,, If I am Adil and I claim to be Dr. Hammond and system incorrectly verifies my claim. False Reject Rate (FRR) is the percentage of the time our authentication system incorrectly rejects the true claim of identity. E.g,, If I am Adil and I claim to be Adil and system incorrectly rejects the claim.

5.1 Simple Shape Moving Interface

The authentication results were obtained for both the true passwords and false passwords. True passwords yielded us the TAR and FAR and false passwords helped us to get the FRR and TRR.

	Authorized	**Not Authorized**	Total
Authorized User	30	0	30
Un-Authorized User	0	28	18
Total	30	28	58

Table 5.1: Confusion matrix - user based authentication results.

Table 5.1 describes the confusion matrix of the authentication process as defined in section 4.1. This confusion matrix only shows whether the user followed all the colored circles for all the animations correctly. The user selected these colored circles during the enrollment phase. As in the case of the 15 participants with who participated in user studies, all the participants were recognized correctly, and the unauthorized person was denied access to the system. Hence using the formula for accuracy and using the values of TP = 20, FP = 0, TN = 18, and FN =0 from the table 5.1, we get the accuracy using Equation 5.1 as 100% for this authentication interface.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

$$F - measure = \frac{2TP}{2TP + FP + FN} \tag{5.2}$$

The F-measure obtained using Equation 5.2 is 1.0. Moreover the FAR = 0% , FRR = 0% , TAR= 100% , TRR= 100% and misclassification rate is 0 % .

Also, we tested individual recognition of all the templates with animations to see the accuracy of our template matching algorithm. This result helped us to understand how much jitter the algorithm can tolerate and whether or not sampling helps our cause to increase the accuracy and reduce the computation time.

Actual/Predicted	**Orange**	**Yellow**	**Blue**	**Green**	**Black**
Orange	**1.0**	0.0	0.0	0.0	0.0
Yellow	0.0	**0.98**	0.0	0.2	0.0
Blue	0.0	0.0	**0.99**	0.0	0.01
Green	0.0	0.0	0.0	**1.00**	0.0
Black	0.0	0.0	0.01	0.0	**0.99**

Table 5.2: Confusion matrix - template matching algorithm, the colors represent the colored circle as showed in Figure 3.2 and Figure 3.3.

Table 5.2 shows the confusion matrix of the template matching algorithm on each of the template paths generated by matching the colored circle with the user's scan path. We can see the accuracy of the template matching algorithm is 98.8%. Now the logical question would be why the template matching algorithm is less accurate than the total accuracy of the authentication system. The answer is due to the different behaviors of authorized and unauthorized users. The password is considered correct if the user follows all of the shapes in the password. So in this scenario, all the cases were truly authenticated and classified correctly by the template matching algorithm as the users knew which colored circle to follow and made fewer wandering gazes to other colors. But in the case where unauthorized users were trying to get into the system, the users jumped from one colored shape to another as they didn't have a particular shape in mind which happens in the case of an authorized user who knows and remembers his/her password. The template matching algorithm would mis-recognize these cases. Table 5.2 shows that blue colored circle was misclassified as black, and the green colored circle was misclassified as yellow in a few cases.

Moreover, this result showed that the accuracy obtained by the template matching algorithms is quite high as compared to some solutions we discussed in Section 2. The resilience of the template matching algorithm to jerky eye/gaze movements is more naturalistic to a human eye. Moreover, after looking at the gaze movements and scan path of the users, we wanted to dive more into the individuality of the eye movement to distinguish between genuine and intruder users, which will be discussed in the Chapter 6. The results from this interface motivated us to go forward with gaze-based authentication which will be more complex and support larger password spaces. Even though the system was extensible to 12 colored circles, we needed a better approach to handle the highly dynamic authentication interface and also try to reduce the cognitive load on the user. Since this interface was built to demonstrate applicability of the gaze-based authentication with template matching for higher accuracy, which it successfully proved, our next goal was to address the shoulder surfing and video analysis attacks. The only downside of this interface was the password space ($5^3 = 125$) and time of authentication, which was around 12 seconds. The authentication time, as well as password space, will be addressed in subsequent interfaces.

5.2 Dynamic Authentication Interface

In this section, we will talk about increasing the password space of the gaze-based authentication system and also how this system helped us to handle shoulder surfing and video analysis attacks. More details about the hacking studies will be discussed in Section 5.3. As we talked in Section 3.2.2 we made the password comparable to a four digit pin so that we can have the comparison with that of pin-based authentication. In this interface, authentication is done when the user follows four colored circles correctly. The number of possible colors are 10 which is again the same as that of the number of digits possible (i.e. 0 through 9). We have two different authentication interfaces and the results will be discussed separately. The error rates, F-measure, and accuracy will be calculated using the Equations 5.1 and 5.2 in Section 5.1.

To address these issues which lead to low accuracy of the system, we enhanced the interface to have only five moving colored circles. The rest will be static during the animation. This interface change enhanced the path-following capability of the user and led to fewer conflicts between two colored circles. Moreover, due to fewer moving colored circles, the participants mentioned that the cognitive load was less as compared to that of the fully randomized interface system. Table 5.3 shows the confusion matrix of the 3-second animation based static-dynamic interface.

Actual/Predicted	**Actual True**	**Actual False**	**Accuracy**	**F-Measure**
True Prediction	97.5%	2.5%	**98.75%**	**0.99**
False Prediction	0%	100%		

Table 5.3: Static-dynamic interface 3 second animations confusion matrix adapted from [3].

During the post-study survey we came to know a few limitations of our system. One, the system is still slow as compared to other real time authentication based systems. Second, the people who are color blind found it hard to use our authentication system. The first limitation is basically the trade off between speed and the randomness of the system which makes it resilient to shoulder surfing attacks completely. In the next section we will talk about the shoulder surfing and video

Edit Distance	**3 Seconds**
0 Error	**97.5%** (39/40)
1 Error	**2.5%** (1/40)

Table 5.4: Static-dynamic interface: recognition error based on the edit distance (higher 0 error is better) adapted from [3].

analysis attacks and how our system outperformed basic as well as gaze-based pin authentication system. We addressed the problem of people having color blindness by including the shapes instead of the colors in our authentication system.

5.3 Hacking Studies

We evaluated our gaze-based authentication system for both shoulder surfing as well as video analysis attacks. The threats were emulated using two separate studies.

1. **Shoulder Surfing attacks**: This attack was emulated by using a single camera which records chain of actions and movements of eyes when a user is inputting the password to the gaze-based authentication system. The camera acts as if someone is shoulder surfing you to check what you are inputting on the authentication interface. Since just eye movements are captured by the web cam of our system, it is difficult to identify what is happening in the system.

2. **Video Analysis Attack**: This attack is more enhanced. The intruder tries to capture all the details of a user's activities while using the authentication system. We emulated this setup by using two cameras to capture both the user's eye movement as well as the authentication interface.

In this user study, we choose the 2-second animation static-dynamic interface. As discussed in the last section, this interface selection is justified by lower latency as well as the accuracy of authentication. We evaluated our system with that of a gaze-based pin authentication system, where a user will use gaze to input the password on a number pad as the authentication screen.

12 participants performed the hacking study. Each participant was asked to use the two studies as mentioned above to evaluate the pin-based authentication system and our gaze-based authentication system. Each user was given three chances to crack the password, which is analogous to the number of tries a user can attempt without blocking the authentication system for the user.

In the first case, where the user was provided with just the eye movements of the user, the results for pin-based authentication system and 5.5 for our proposed gaze-based authentication system. We see that for the case of pin password-based authentication systems, hackers were easily able to crack the system. The reason being the static and localized position of all the numbers on the interface. For twelve people, each was given two passwords to guess given the front camera videos. For the case of the pin-based password, a large number of passwords were hacked, around 80% of them. Obviously, this is too high and needs to be addressed. There was no need to use the double-camera system in this case as the single-camera easily resulted in 80% of passwords being hacked.

Video	**1st Try**	**2nd Try**	**3rd Try**	**Total**
Single	0	0	0	0%
Dual	4 (16.7%)	0	0	16.7%

Table 5.5: 2-second animation static-dynamic authentication interface hacking: The number of passwords hacked in each try for both single and double camera attacks adapted from [3].

The results of our gaze-based authentication system are in Table 5.5. It clearly shows that none of the hackers were successful in cracking our authentication system given one front camera feed only. This makes our system resilient to shoulder surfing attacks. In the case of the dual video analysis, we see that around four passwords were cracked by three hackers, one each by two, and two by one hacker. The password cracking was possible using the deep video analysis given two feeds, that is front and back camera feeds. When the hackers were asked how they cracked the password, they replied that they were able to synchronize precisely both front and back camera video feeds. Moreover, in all the cracked password cases, the users waited a long time before starting the new animations and also gave a clue to the camera by making some movement when they started the subsequent animations. The Figure 3.2 shows the setup used for the hacking studies.

As the results in the Table 5.5 show that even when we use the dynamic interface based gaze based authentication, the intruders can guess the password using the advance video analysis attacks. To handle these cases where the intruder can detect the password using the video analysis attacks or generally using the guess or some kind of inferences, we would want a system to handle these intruders. The next section of this thesis handles these attacks and generates a predictive model using gaze features that helps us to avoid the intruders from getting the access to the system even though they know the password. Chapter 6 describes the methodology, the features and the algorithms used to do the classification of intruder from that of genuine user.

6. INTRUDER DETECTION BASED ON GAZE FEATURES

As we have seen, even after having highly dynamic interfaces, the hackers who guessed the password can get authorized. This kind of attack is possible with almost all types of authentication systems except for biometric authentication systems, which handle this kind of threat using the spoof detection. Since the pin or password inputted by the authorized user or intruder is the same unless the underlying system is using the typing pattern of the user, which again is behavioral biometrics where a unique way/behavior of user distinguishes him from other users. So unless we are using some system which identifies the genuine user there are chances that the intruder/attacker can get authenticated.

In this section, we will explore the possibility of using the uniqueness of the gaze movement of users to distinguish the intruder from genuine user. We want to reduce the number of unauthorized accesses to the system by employing an intruder detection algorithm using the gaze movement of the user. We will use the same authentication interface used in the static-dynamic interface mentioned in Section 3.2.2 and the layout given in Figure 3.4. We hypothesize that the gaze pattern of a user while following a colored circle path is different from the gaze pattern of other users. Moreover, we also try to show that the features obtained from the eye movement data can help us differentiate between different kind of users based on the movement of the eyes.

6.1 User Study for Data Collection

To demonstrate the individuality of gaze movements when following a path we tested our algorithm on real users. We collected data from 16 participants (mean age of 23.6) in a two-phase user study.

1. We collected gaze data from authorized users as they used the authentication system. We collected data for 30 authorized accesses for each user to have data for training and testing the algorithms as well as to make sure the data were not jittery.

2. Second the unauthorized users were given the password sequence of the authorized user and

were asked to use the authentication system. We collected the gaze data of the unauthorized user to help us to determine the accuracy of the system. Each user was asked to input every other user's password sequence 4 times.

We used the enrollment interface shown in Figure 3.5 for both the phases of the study. The user data were collected by inputting the userId in the text box, while the password was statistically generated by us so that a variety of password combination were used. After the password sequence was given to the user, gaze data were collected by running the authentication interface for this password thirty times. These data were collected to get the user's baseline. The intruder's data were collected by giving a different user the password sequence and asking them to use the authentication system. The intruder's gaze data are different from that of actual authorized user as depicted in Figures 6.1, 6.2, and 6.3.

The users were asked to come different times to make ensure that a wide variety of the data is obtained. This variety of data involves different lightning conditions, different calibrations with the eye tracking and seating positions. We originally used 64 features to detect the difference between the actual user and an intruder trying to get into the system using user's password.

6.2 Features used for Intruder Detection

In order to detect intruders reliably and accurately we extracted 44 features from the data. In this section we will describe these features. These features can be broken down into several subgroups and are discussed in detail in the following sections.

6.2.1 Spatial and Frequency Domain Features

The spatial features include the average (X, Y, Pupil radius) equation 6.1, standard deviation (X,Y,Pupil radius) equation 6.2, median(X,Y, Pupil radius), peak(X,Y, Pupil radius).

$$\mu_x = \frac{\sum_{i=1}^{n} x_i}{n} \tag{6.1}$$

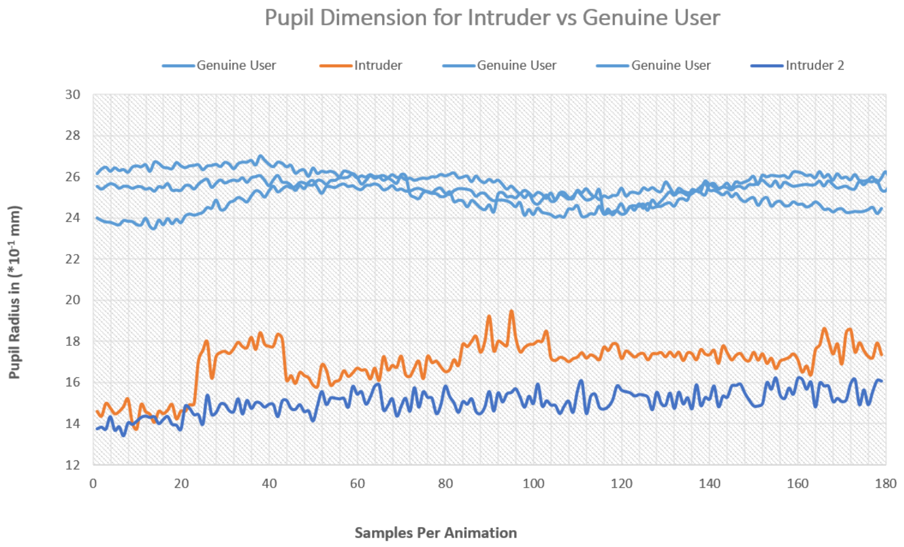


Figure 6.1: A plot showing the variation in the pupil dimensions of an intruder and genuine user.

$$\sigma_x = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i - \mu_x} \tag{6.2}$$

These features helped us to understand how the data differ in the users on an aggregate level. We also calculate the frequency domain features which include entropy and energy from the gaze data. The Equation 6.4 is used to calculate the entropy of a random variable given its probability. In our context, entropy is used as a measure of the jerkiness of the data in terms of movement of gaze in X-Y coordinate system. We also calculated the entropy of the Pupil radius to understand the dynamics of the pupil dilation during the authentication.

$$energy = \sum_{j=1}^{n} \frac{a_j^2 + b_j^2}{n} \tag{6.3}$$

$$entropy = \sum_{j=1}^{n} p_j * \log(p_j) \tag{6.4}$$

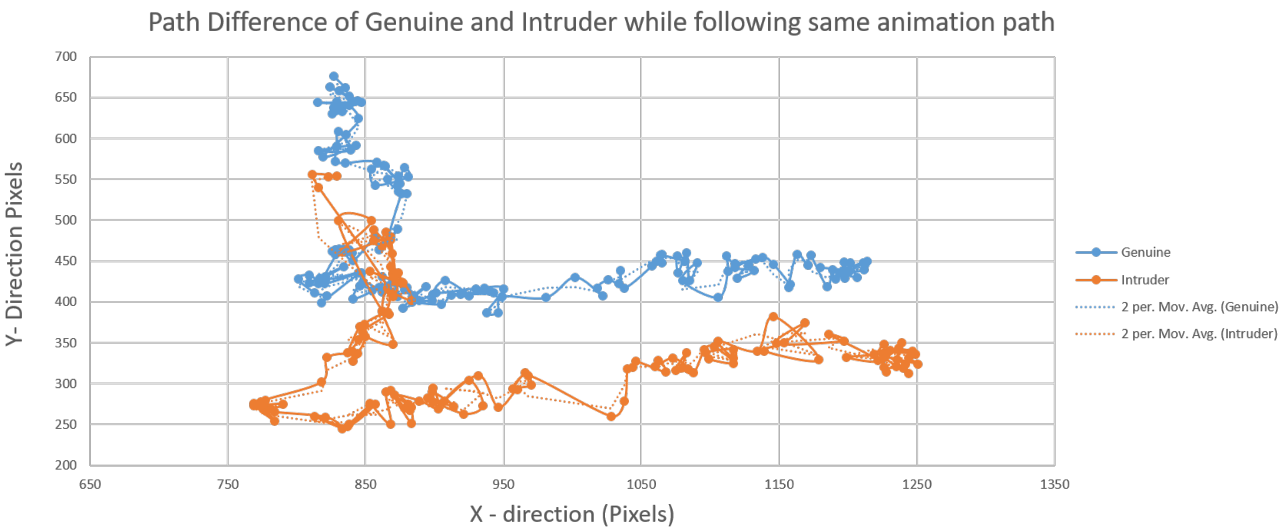


Figure 6.2: A plot showing the variation in the fixations and saccades of an intruder and genuine user.

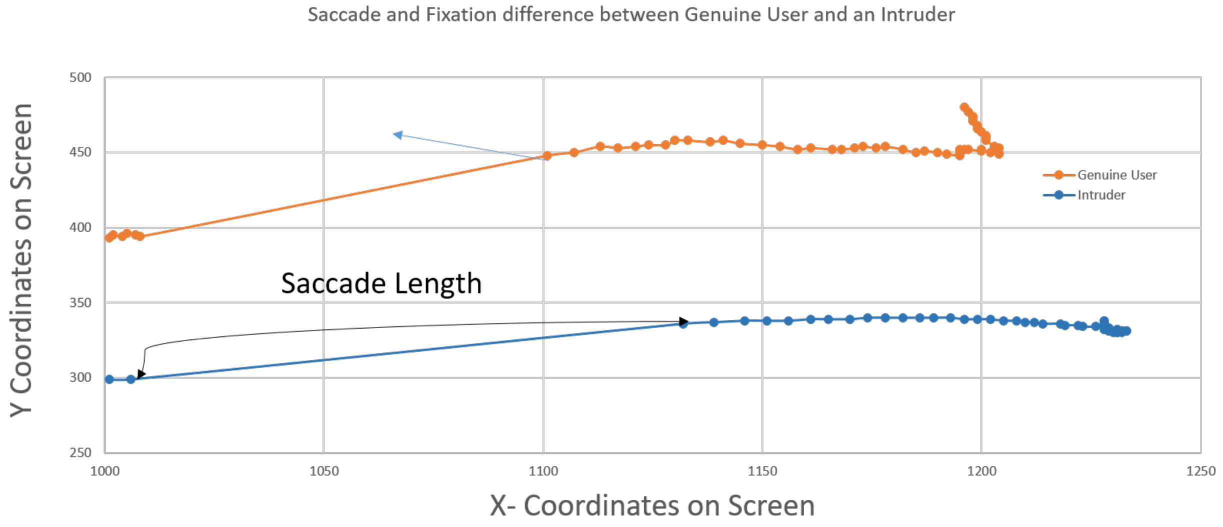


Figure 6.3: The distinction of saccade length and distribution of saccades for different users.

$$p_j = \frac{\sqrt{a_j^2 + b_j^2}}{\sum_{k=1}^{n} \sqrt{a_k^2 + b_k^2}} \tag{6.5}$$

In the equation 6.3, a and b represent the real and imaginary part of the data point when it has been converted into the frequency domain.

We calculate the Fourier domain of each data point using the Fast Fourier Transform (FFT). The formula for the Fourier transform conversion is given in Equation 6.6. In the frequency domain the jerkiness is reflected by the number of the peaks, if the peaks are small and large in number then the scan path of the user is highly jerky, whereas few number of peaks and high amplitude means that the scan path is less jerky [109]. We are handling the jerkiness in our authentication system using the sampling to reduce the number of points going out of the scope.

$$x[\mathbf{k}] = \sum_{j=0}^{n-1} e^{(-2\pi j/N)\mathbf{k_n}} x[n] \tag{6.6}$$

In the equation, the x[n] corresponds to the data point in spatial domain which is discrete and the x[k] describes the equivalent data point in the frequency domain.

6.2.2 Eye Gaze Domain based features

Spatial and Fourier domain features only cover some aspects of the data, we also wanted to include the features which are unique based on the user at eye movement level. This led to the exploration of more features based on the gaze pattern. We explored certain features limited to the capability of the gaze tracking module we have. The features include the number of fixations, duration of the fixation, gaze velocity. These features helped to distinguish people based on the how the eye movement and the muscles play a role in saccades and fixations. We explore the saccades when the colored circle is moving and the fixations when the user is constantly looking at a static colored circle. Fixations are typically the gaze points that are voluntary and average duration of gaze is around 100–300 ms. Saccades are rapid eye movements from one fixation to other. Average duration of a saccade is usually 20–50ms. Figure 6.4 shows a representation of fixations and saccades.

6.2.3 Sketch Based Features

As we have seen in Figure 3.3 the stroke made by the gaze data on the interface when traced makes a sketch-like drawing. There are many sketch based algorithms to distinguish between two

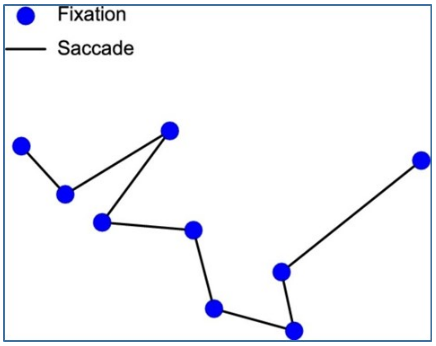


Figure 6.4: The typical fixations and saccades.

sketches and as such we can model them in various ways. Geometry-based algorithms [110] such as Tahuti [111], PaleoSketch [112, 113, 114], and Ladder [115, 116, 117, 118, 119, 120, 121, 122, 123, 124] are used as low-level recognizer to detect primitive shapes in strokes. The use of sketch based features enhance the distinction between the user's and intruder's gaze path.

The features we used include Rubine's [125, 63, 126] 10 features, leaving three features based on the time as the speed and the time is constant for all the participants. We additionally included the extended features from Long's [127] features also help in the gesture-based classification such as in our case. Moreover, since we are distinguishing between two gestures or sketches, we also use the Hausdorff's distance [128] to measure the difference between two sketches or scan paths of different users. The Hausdorff measure between two vectors is given by D_A and D_B where each vector is calculated by Equation 6.7,

$$D_A = \min_{b \in P_B} |a - b|, a \in P_A \tag{6.7}$$

where the P_A and P_B are vector of points in A and B respectively and N_A represent the number of points in each vector. The two sided measure for the Hausdorff modified version is calculated using equation 6.8, And the final measure is calculated using the equation 6.9

$$h(A, B) = \frac{\sum(D_A)}{N_A} \tag{6.8}$$

$$H(A, B) = \max(h(A, B), h(B, A)) \tag{6.9}$$

To describe the all the Rubine and Long's features as aggregated in the book [129] we need to define few terms and what they mean. The following enumeration gives the terminology for the items we will be using to calculate the Rubine and Long's feature. Some of Rubine's features are represented in the Figure 6.5.

1. n the total number of data points in a gaze path
2. p_0 the first data point in a gaze path
3. p_i the i^{th} data point in a gaze path
4. p_{n-1} the last data point in a gaze path
5. (x_0, y_0, t_0) the x, y, and $time$ value for the first data point in a gaze path
6. (x_{n1}, y_{n1}, t_{n1}) the x, y, time value for the last data point in a gaze path
7. (x_i, y_i, t_i) the x, y, time value for the i^{th} data point in a gaze path
8. x_{min} the minimum x value of the gaze path (identical to the minimum x value of the bounding box)
9. x_{max} the maximum x value of the gaze path (identical to the maximum x value of the bounding box)
10. y_{min} the minimum y value of the gaze path (identical to the minimum y value of the bounding box)
11. y_{max} the maximum y value of the gaze path (identical to the maximum y value of the bounding box)
12. α the starting angle of the gaze path

13. β the angle between the first (p_0) and last (p_{n-1}) point

14. θ_i the angle of the line between the i^{th} and the $(i+k)^{th}$ point, for some constant offset k

15. d the length of the bounding box encapsulating the entire gaze path

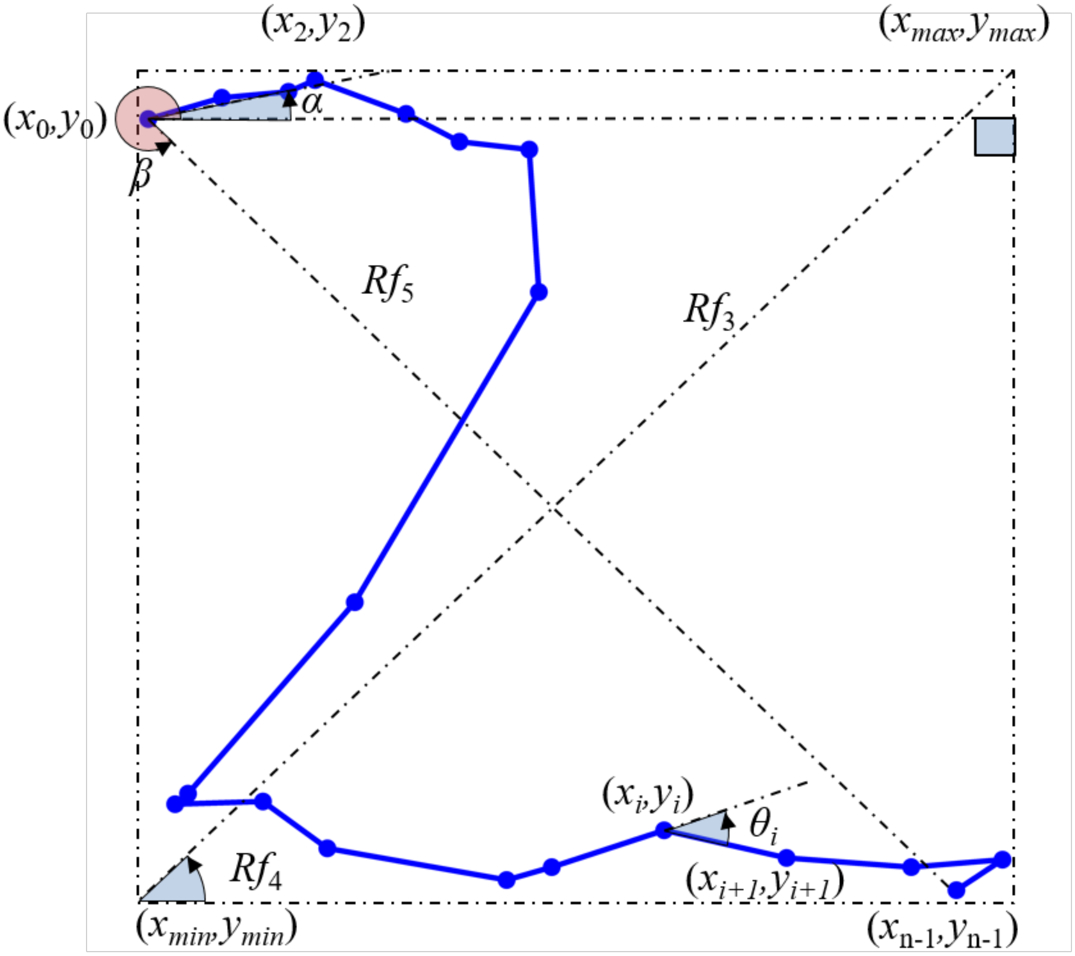


Figure 6.5: A visual reference for the terms defined in section 6.2.3 and Rubine features as discussed in equations from [4].

The calculation of the feature will be done using the above terms and the visual figure will be used to reference what the feature means for few of them. Moreover, we will use some of the features from Rubine and Long's Feature which helped algorithm to distinguish between the intruder and genuine user are:

$$f_8 \ = \ \sum_{i=1}^{n-1} \sqrt{\Delta x_i^2 + \Delta y_i^2} \tag{6.10}$$

where $\Delta x_i \ = \ x_i - x_{i-1} \ and \ \Delta y_i \ = \ y_i - y_{i-1}$.

1. Feature *f8* Equation 6.10: This feature measures the total length of the path formed by the gaze. This feature distinguishes between an intruder and genuine person using the total path length formed by saccades. It also helps in differentiating between the saccades of two users with similar bounding boxes and same animations of the shapes.

2. Feature *f15* Equation 6.11: This feature represents the curviness of gaze path. Since the path formed by the gaze generated while following a path is different for different people, this feature helps to quantify that change in terms of the curviness of the path.

3. Feature *f17* Equation 6.13: This feature represents the density metric of gaze path corresponding to the total length between start and end point.

4. Feature *f18* Equation 6.12: This feature represents the density metric of gaze path length. It is ratio of feature *f8* to the length of the diagonal of bounding box of the gaze path.

5. Feature *f27* Equation 6.14: This feature quantifies the differentiation between two shapes. The equation computes the Normalized distance between the direction extremes (NDDE). NDDE calculates the difference between the point of highest direction value and the lowest value normalized by the total gaze path (*f8*). Higher NDDE values mean curved shapes whereas lower values mean a poly-line.

6. Feature *f28* Equation 6.15: This feature also measures the curviness of the gaze path. It calculates the Direction Change Ratio (DCR) which computes the ration of maximum change in direction to the average change. Poly-lines have high DCR whereas the curved lines have a lower value.

Rest of the feature definition for Rubine and Long's features are mention in Appendix A.

$$f_{15} \;=\; \sum_{i=1}^{n-2} \begin{cases} |\theta_i|, & \text{if } \theta_i < 19^\circ \\ 0, & \text{otherwise} \end{cases} \tag{6.11}$$

$$f_{18} \;=\; \frac{\displaystyle\sum_{i=1}^{n-1} \sqrt{\Delta x_i^2 + \Delta y_i^2}}{\sqrt{[(y_{max} - y_{min})^2 + (x_{max} - x_{min})^2]}} \tag{6.12}$$

where $\Delta x_i \;=\; x_i - x_{i-1} \; and \; \Delta y_i \;=\; y_i - y_{i-1}$.

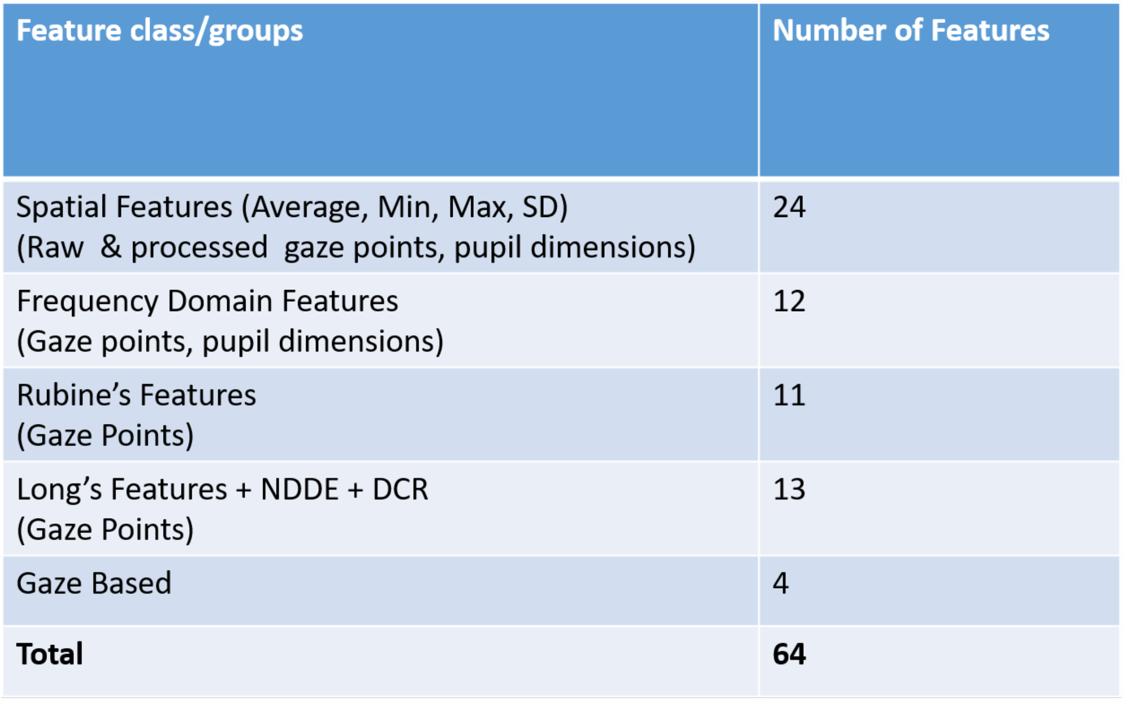
$$f_{17} \;=\; \frac{\displaystyle\sum_{i=1}^{n-1} \sqrt{\Delta x_i^2 + \Delta y_i^2}}{\sqrt{[(x_{n-1} - x_0)^2 + (y_{n-1} - y_0)^2]}} \tag{6.13}$$

$$f_{27} \;=\; \frac{\max\limits_{i=1}^{n-2} \frac{\Delta y_i}{\Delta x_i} - \min\limits_{i=1}^{n-2} \frac{\Delta y_i}{\Delta x_i}}{\sum_{i=1}^{n-1} \sqrt{\Delta y_i^2 + \Delta x_i^2}} \tag{6.14}$$

where $\Delta x_i \;=\; x_i - x_{i-1} \; and \; \Delta y_i \;=\; y_i - y_{i-1}$.

$$f_{28} \;=\; \frac{\max\limits_{i=1}^{n-2} \frac{\Delta y_i}{\Delta x_i}}{\displaystyle\sum_{i=1}^{n-2} \left[\frac{\Delta y_i}{\Delta x_i}\right] / n - 2} \tag{6.15}$$

where $\Delta x_i \;=\; x_i - x_{i-1} \; and \; \Delta y_i \;=\; y_i - y_{i-1}$.



Feature class/groups	Number of Features
Spatial Features (Average, Min, Max, SD) (Raw & processed gaze points, pupil dimensions)	24
Frequency Domain Features (Gaze points, pupil dimensions)	12
Rubine’s Features (Gaze Points)	11
Long’s Features + NDDE + DCR (Gaze Points)	13
Gaze Based	4
Total	**64**

Figure 6.6: All the features we used in our system.

7. INTRUDER DETECTION FEATURE SELECTION AND CLASSIFICATION

The user studies were conducted according to the protocol described in Section 6.1 and the feature extraction module was run over the data collected to get the features described in Section 6.2. In this section, we discuss feature selection, our implementation of intruder detection system, user wise model for classification and results of our proposed intruder detection system.

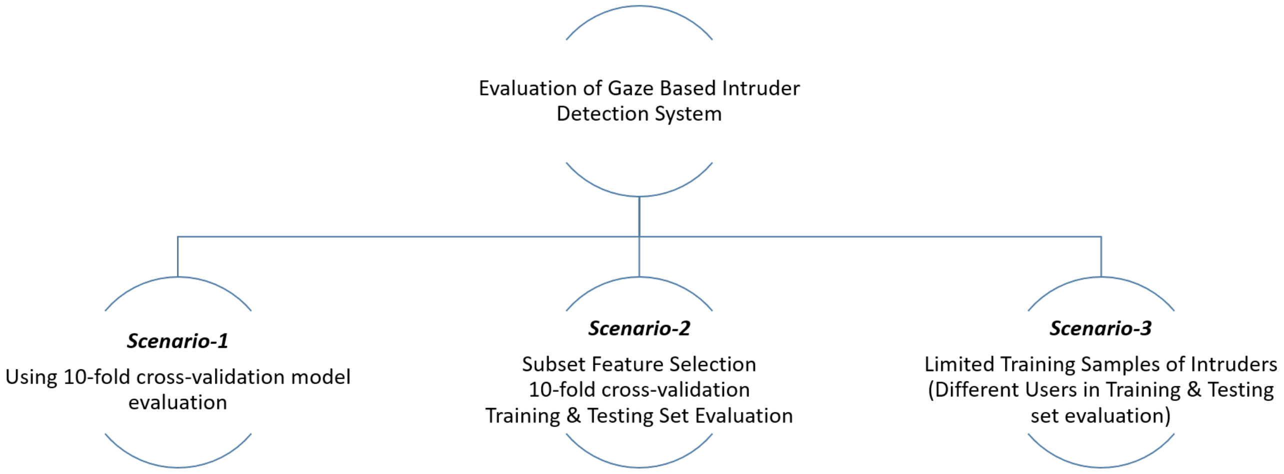

Evaluation of Gaze Based Intruder Detection System
Scenario-1
Using 10-fold cross-validation model evaluation
Scenario-2
Subset Feature Selection
10-fold cross-validation
Training & Testing Set Evaluation
Scenario-3
Limited Training Samples of Intruders
(Different Users in Training & Testing set evaluation)

Figure 7.1: The three evaluation scenarios for gaze intruder detection.

7.1 Feature Subset Selection

To determine the features which were optimal for intruder detection, we performed subset selection on the 64 extracted features. The process of subset selection is critical for any machine learning algorithm. Although a large number of features make the classification more accurate, it can adversely make the process of generating model computationally intensive thereby leading to high latency. Moreover, features selected during initial phase can have high correlation making the generated model an approximation and inadequate for classification. Also, irrelevant features can cause overfitting of machine learning algorithms like ANN [130], making the generated model sub-optimal and faulty. In more technical terms feature subset selection evaluates the worth of adding a new feature to the subset using Equation 7.1, where $\overline{r_{zc}}$ is the worthiness of the subset of features, $\overline{r_{zi}}$ is the average of the correlations between the features and the classification results, k is the number of features, and r_{ii} is the average of the correlations between the features.

$$r_{zc} = \frac{k\overline{r_{zi}}}{\sqrt{k + k(k-1)\overline{r_{ii}}}} \tag{7.1}$$

The BestFirst method of uses hill climbing to find the optimal solution. One of the other effective feature selection techniques is Sequential Floating Search Methods (SFSM) [131, 132]. There are two categories of floating search algorithms: Sequential Floating Forward Selection (SFFS) and Sequential Floating Backward Selection (SFBS). In the SFFS, the algorithm starts with a null set of features and at each step, and the best feature satisfying a defined criterion function is included in current feature set. This algorithm also confirms the likelihood of improvement of the criterion if a feature is excluded from the current set. In this case, the worst feature which leads to low accuracy (criterion function) is eliminated from the set. Therefore, SFFS keeps on increasing and decreasing the number of features until the desired set of features are obtained. The backward search (SFBS) works analogously, starting with a full feature set and executing the search until all the features tested and the corresponding accuracy as a criterion.

7.2 Recognition Algorithms

The features selected after the subset selection were tested on five different classifiers: Random Forest, Multilayer Perceptron, J48 (C4.5), Naive Bayes and Random Tree. We evaluated the recognition accuracy of our proposed system using all popular machine learning algorithms present in WEKA Toolkit. However, we only selected five of these machine learning algorithms according to the characteristics of features. The data we obtained were labeled, so we choose the supervised learning algorithms. We will talk more about these machine learning algorithms we selected.

7.2.1 Decision Trees

The first class of algorithms is Decision Trees. Decision trees are one of the most commonly used machine learning algorithms for a plethora of reasons. The model generated by these algorithms are easy to understand. These algorithms try to interpret the latent factors or relationships in the data. Moreover, these algorithms implicitly determine the optimal subset feature selection while building the model.

In our proposed system, we explored three of these algorithms, J48, Random Tree, and Random Forest. Random Tree is one of the simplest among these algorithms where each node represents the random subset of features. This algorithm doesn't prune the data. The J48 algorithm is one of the popular decision tree algorithms widely used. It does pruning of the data to form 5 decision trees. Both decision tree and J48 algorithms are susceptible to overfitting in case of noisy data. To handle overfitting, a superior machine algorithm Random Forest was used. It constructs a series of random trees to make the classification decisions thus avoiding the overfitting. All the decision tree based algorithms can handle missing data and can be used to nominal as well as numeric data.

7.2.2 Artificial Neural Network

Artificial Neural Networks are known machine learning algorithm that model the non-linearity of the data. The class of ANN we are using in our proposed solution is Multilayer Perceptron. It is a feed-forward ANN that uses back-propagation or errors to train its model. All the nodes are sigmoid as we need binary classification. ANN handles the numeric data well and finds the nonlinear relationship among the data to calculate the weights to each node in hidden layers. Since the algorithm involves a lot of back-propagation and error correction, it is very slow as compared to function based machine learning algorithms. It is known to be used for gesture recognition thus serving our purpose.

7.2.3 Bayesian Networks

To model using the probabilistic theory, we used the Naive Bayes algorithm. Naive Bayes is a simplistic probability based machine learning algorithm that handles numeric data and uses conditional probability to build the Bayesian network. The algorithm uses probability decisions at each node of the network to generate a full probabilistic graphic model.

Table 7.1: Performance of the classifiers for each user using all 64 features, where A is Accuracy in percentage (%) and B is F- measure.

User/Classifier	**C4.5(J48)**		**Naive Bayes**		**Multilayer Perceptron**		**Random Tree**		**Random Forest**	
	A (%)	**B**	**A (%)**	**B**	**A (%)**	**B**	**A (%)**	**B**	**A (%)**	**B**
User 1	**91.30**	**0.91**	81.52	0.81	89.13	0.88	82.60	0.82	**92.39**	**0.92**
User 2	**93.47**	**0.93**	89.13	0.89	**93.47**	**0.93**	92.39	0.924	**97.82**	**0.97**
User 3	**92.55**	**0.92**	90.42	0.90	92.80	0.91	**93.60**	**0.93**	**93.61**	**0.95**
User 4	93.47	0.93	**94.56**	**0.94**	**97.81**	**0.97**	90.20	0.90	**96.73**	**0.96**
User 5	86.36	0.86	81.67	0.81	**93.18**	**0.93**	85.22	0.85	**94.31**	**0.94**
User 6	86.51	0.86	85.39	0.85	88.76	0.88	**93.25**	**0.93**	**92.13**	**0.92**
User 7	93.18	0.93	87.50	0.87	**95.45**	**0.95**	85.22	0.85	**94.31**	**0.94**
User 8	**87.50**	**0.87**	84.09	0.84	87.50	0.87	87.50	0.87	**92.04**	**0.92**
User 9	**96.59**	**0.96**	96.59	0.96	**98.86**	**0.98**	95.45	0.95	**99.90**	**0.99**
User 10	94.31	0.94	**96.59**	**0.96**	**97.70**	**0.97**	93.18	0.93	**97.72**	**0.97**
User 11	88.09	0.88	85.71	0.85	90.10	0.90	81.82	0.81	**92.85**	**0.92**
User 12	94.04	0.94	**96.42**	**0.96**	94.04	0.94	94.04	0.94	**98.80**	**0.98**
User 13	93.18	0.93	87.50	0.87	93.25	0.92	**95.45**	**0.95**	**96.59**	**0.96**
User 14	85.33	0.85	**89.88**	**0.90**	88.76	0.88	85.39	0.85	**94.55**	**0.95**
User 15	**96.55**	**0.96**	**96.59**	**0.96**	94.57	0.94	98.85	0.98	**98.99**	**0.99**
User 16	**93.61**	**0.93**	92.05	0.92	89.72	0.89	89.63	0.86	**96.54**	**0.96**

7.3 Results and Analysis

The evaluation of these algorithms was done using the classification accuracy and F-measure. The accuracy can be misleading at times when the data is class skewed. Skewed data has more instances of one class to that of other classes. Data collected in the user study were divided into skewed and non-skewed classes. For the skewed data, the baseline of the classification/recognition accuracy is calculated using the majority classifier. ZeroR in WEKA Data Mining Toolkit is a majority classifier which classifies every instance of the data as an instance of majority class. We use the F-measure to overcome this problem of comparison between the baseline accuracy and accuracy obtained by the classifier.

Table 7.2: Cumulative accuracy and F-measure of classifiers across the users.

Classifier	**Overall Accuracy (%)**	**Overall F-Measure**
Naive Bayes	89.72	0.89
Random Tree	90.23	0.89
C4.5 J48	91.62	0.91
Random Forest	**95.58**	**0.95**
Multilayer Perceptron	92.81	0.92

7.3.1 Intruder Detection using User Based Models

We generated the model for each user who took part in the user study. We got 16 models for each user. These models were all the binary classifiers predicting whether the new instance supplied to the model is a genuine user or an intruder. We evaluated two different training and testing cases.

The results obtained before the subset selection per user can be seen in Table 7.1. All 64 features were selected for generating the results given in Table 7.1. As you can see, all the features we discussed earlier are more than sufficient to reliably classify an intruder from a genuine user. The results were obtained by modeling each user separately. Each user has a set of positive instances (30), where the user itself is using the authentication system and negative instances $(15 * 4 = 60)$ where an intruder is trying to imitate the genuine user's password by following the same pattern of colored circles. The majority classification accuracy for all the user is around 66% but as we can see that all the classifiers for each user can easily classify the genuine vs intruder with greater than 85% accuracy. This shows that while following a pattern the gaze movement of a user is distinct compared to any other user. Moreover, results in Table 7.1 also show that the F-measure value for all the classifiers is higher than 0.80, meaning that the classifier is able to distinguish between the real and intruder users easily. One important detail we noticed while training the model is all the classifiers took an average of 0.5 seconds except Multilayer Perceptron which took around 45

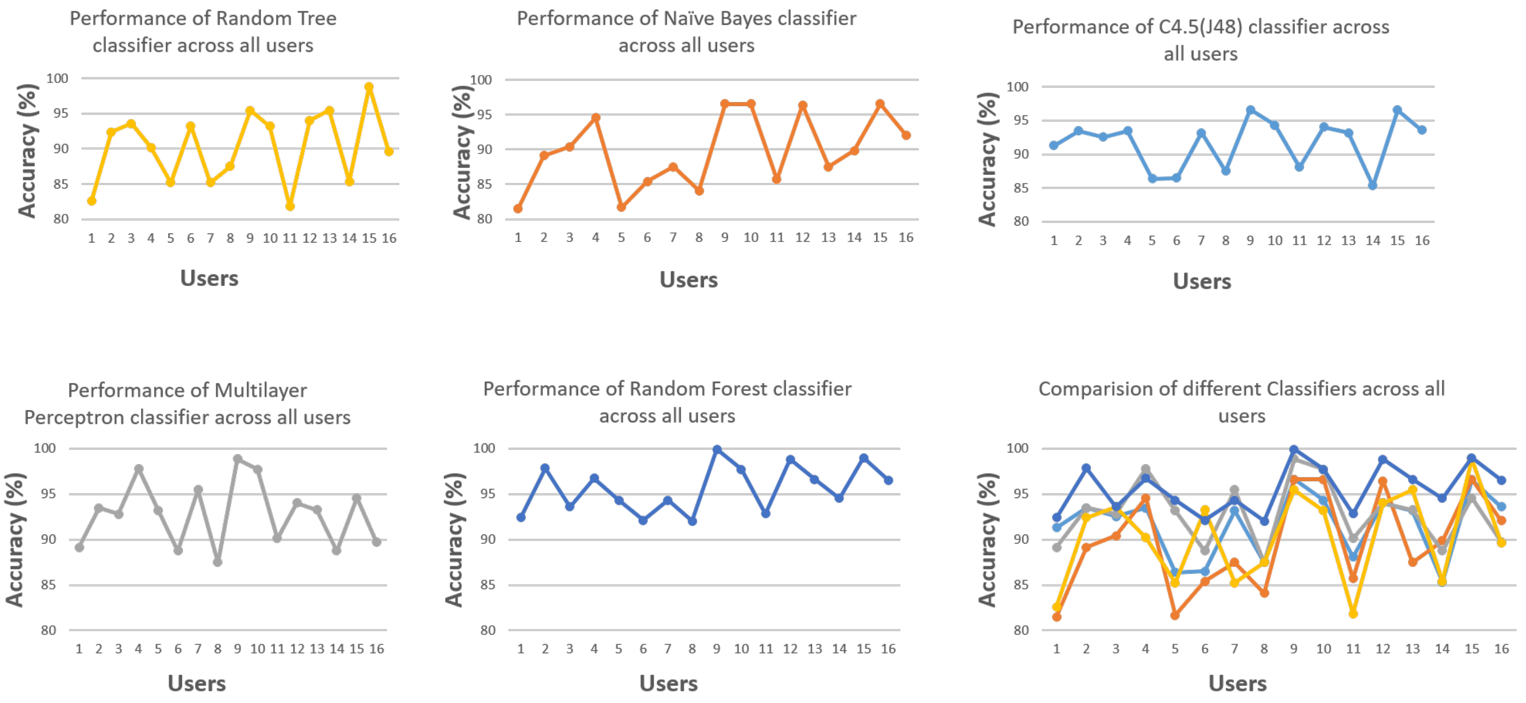


Figure 7.2: The graphs showing the performance of the five classifiers on different user models
a) Performance of Random Tree classifier b) Performance of Naive Bayes classifier c) Performance of j48 classifier d) Performance of Multilayer Perceptron e) Performance of Random Forest classifier f) All the classifiers combined in one graph.
The classifiers are run on all the user models (16) separately.

Table 7.3: Cumulative confusion matrix for C4.5 Decision Tree classifier.

User	Classified As	
	Genuine User	Intruder
Genuine User	**0.88**	0.12
Intruder	0.06	**0.94**

Table 7.4: Cumulative confusion matrix for Naive Bayes classifier.

User	Classified As	
	Genuine User	Intruder
Genuine User	**0.87**	0.13
Intruder	0.07	**0.93**

seconds for each user. Table 7.2 represents the cumulative accuracy and F-measure for all the users for different classifiers.

Tables 7.3, 7.4, 7.5, 7.6, and 7.7 show the cumulative confusion matrix for all the user for five different classifiers used.

7.3.2 Classification Results after Subset Selection

The features selected by using the two methods of subset selection were same. The reason for this behavior is both the subset selection algorithms make a greedy choice while selecting the feature which increases the classification accuracy. Moreover, both algorithms avoid selecting a feature which lowers the classification accuracy. After running the subset feature selection algorithm on our features we got the features mentioned in Figure 7.3.

The results in Table 7.8 shows the accuracy and F-measure for each user after the subset selec-

Table 7.5: Cumulative confusion matrix for Mutlilayer Perceptron classifier.

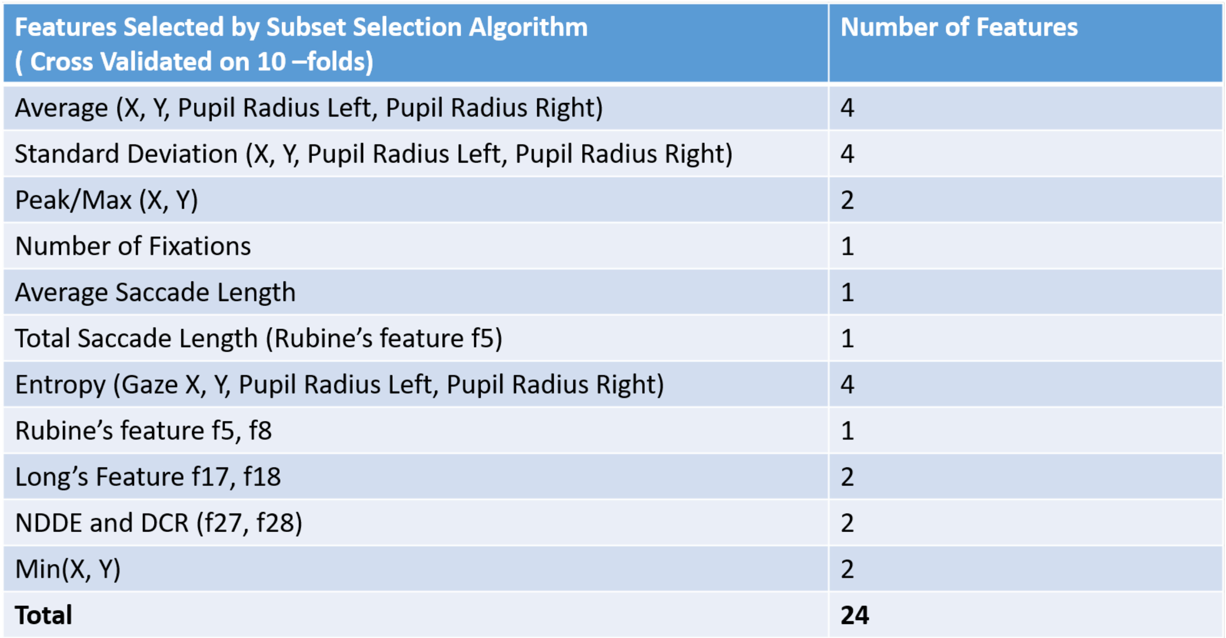
User	Classified As	
	Genuine User	Intruder
Genuine User	**0.94**	0.06
Intruder	0.06	**0.94**

Table 7.6: Cumulative confusion matrix for Random Tree classifier.

User	Classified As	
	Genuine User	Intruder
Genuine User	**0.87**	0.13
Intruder	0.07	**0.93**

Table 7.7: Cumulative confusion matrix for Random Forrest classifier.

User	Classified As	
	Genuine User	Intruder
Genuine User	**0.95**	0.05
Intruder	0.03	**0.97**



Features Selected by Subset Selection Algorithm (Cross Validated on 10 –folds)	Number of Features
Average (X, Y, Pupil Radius Left, Pupil Radius Right)	4
Standard Deviation (X, Y, Pupil Radius Left, Pupil Radius Right)	4
Peak/Max (X, Y)	2
Number of Fixations	1
Average Saccade Length	1
Total Saccade Length (Rubine’s feature f5)	1
Entropy (Gaze X, Y, Pupil Radius Left, Pupil Radius Right)	4
Rubine’s feature f5, f8	1
Long’s Feature f17, f18	2
NDDE and DCR (f27, f28)	2
Min(X, Y)	2
Total	**24**

Figure 7.3: Features selected by the subset selection algorithm by running the algorithm using the 10-fold cross validation and selecting the features that existed in more than 6 folds and also appeared in all the users subset selection.

tion of features. Tables 7.9, 7.10, 7.11, 7.12, and 7.13 shows the cumulative confusion matrix of all the classifiers used in our study. Comparing the results from the previous section, we saw the decrease in accuracy and F-measure. The reason for this behavior is when we selected the subset

Table 7.8: Cumulative accuracy and F-measure of classifiers across the users after subset selection algorithm: using the features from Figure 7.3.

Classifier	**Overall Accuracy (%)**	**Overall F-Measure**
Naive Bayes	87.72	0.87
Random Tree	89.24	0.89
C4.5 J48	89.66	0.89
Random Forest	**92.08**	**0.91**
Multilayer Perceptron	90.81	0.90

Table 7.9: Cumulative confusion matrix for C4.5 Decision Tree classifier.

User	Classified As	
	Genuine User	Intruder
Genuine User	**0.88**	0.12
Intruder	0.09	**0.91**

of features, we selected subset features that appeared in all resultant sets after running subset selection algorithm on all users. Also, this helped in making sure that a user's model is not overfitting. The classification accuracy and F-measure of the Random Forest algorithm are high for before and after subset selection algorithm due to the pruning of the dataset, thereby removing the outliers which cause the failure.

7.3.3 Classification Results For Skewed Data

We also explored the possibility of using the fewer negative instances i.e. intruder trying to use authentication system. Instead of having 10 users' intruder data for training we used 4 to balance the positive samples. This reflects the practical scenario for the model making as the number of

Table 7.10: Cumulative confusion matrix for Naive Bayes classifier.

User	Classified As	
	Genuine User	Intruder
Genuine User	**0.87**	0.13
Intruder	0.11	**0.89**

Table 7.11: Cumulative confusion matrix for Mutlilayer Perceptron classifier.

User	Classified As	
	Genuine User	Intruder
Genuine User	**0.90**	0.10
Intruder	0.09	**0.91**

Table 7.12: Cumulative confusion matrix for Random Tree classifier.

User	Classified As	
	Genuine User	Intruder
Genuine User	**0.89**	0.11
Intruder	0.11	**0.89**

intruders training data will always be limited. We also tried without using any negative instance of the sample data, but the model didn't understand what is considered as negative and classified everything as positive. We will leave the model generation based on the no negative instance for future work. The results obtained using these limited intruder training instances showed that intruder detection is possible using gaze. The results in Table 7.14 show the accuracy and F-measure of the system using limited negative training instances.

7.3.4 Clustering based Results

We explored the possibility of using just the features to detect the intruder vs genuine user without using the training. For achieving this we tried to use the clustering method to detect if the genuine users' cluster will be separable from that of intruder users. We used two clustering methods for our data Expectation Maximization and Simple K means. We ran k-means clustering

Table 7.13: Cumulative confusion matrix for Random Forrest classifier.

User	Classified As	
	Genuine User	Intruder
Genuine User	**0.91**	0.09
Intruder	0.06	**0.94**

Table 7.14: Cumulative acccuracy and F-measure of classifiers across the users after subset selection algorithm: using the features from Figure 7.3.

Classifier	Overall Accuracy (%)	Overall F-Measure
Naive Bayes	84.72	0.84
Random Tree	86.23	0.86
C4.5 J48	86.68	0.86
Random Forest	**90.08**	**0.89**
Multilayer Perceptron	88.80	0.88

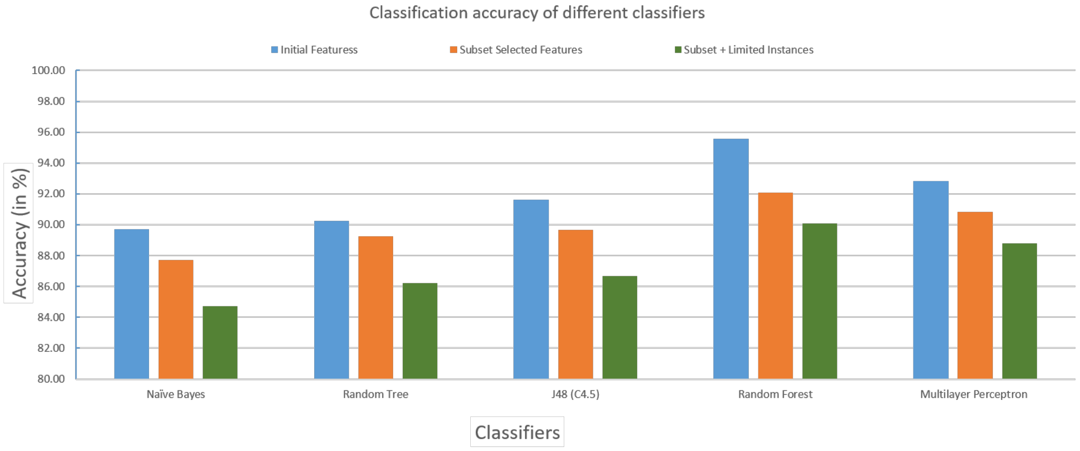


Figure 7.4: Classification accuracy of different classifiers.

algorithm on each user data and distance function used by the algorithm was euclidean distance. The maximum iteration for the clustering algorithm was set to 500 and number of clusters were 2 (since we are classifying the data into two clusters). Expectation Maximization was set to use two distance based evaluations i.e. Euclidean distance based and Manhattan distance based. The number of iteration were set to 500 and the clusters were set again 2 for binary clusters. Since the Euclidean distance and Manhattan distance doesn't respect the non-linearity of the features and calculated the clusters with a mix of genuine and intruder users. The results obtained by the clustering algorithm was very nominal and couldn't provide conclusive results due to distance measures used. The cumulative results over all the users for both the clustering algorithms is shown in Table 7.15.

Table 7.15: Clustering accuracy of cumulative user models.

Clustering Algorithm	Cluster Accuracy (%)
Expectation Maximization	71.36
k-Means	72.06

7.4 Algorithm Accuracy

The results showed that classification of an intruder from a genuine user using the feature set given was better than the majority classifier. The basis for this difference is evaluated using the features we used. The classification accuracy of classifiers on an average was 89%. The classifiers used by our system were trained on features from 15 samples of a genuine user and 16 samples from intruders. The testing set consisted of 15 samples of a genuine user and 16 samples from intruders (instances from 13 remaining users selected at random). The random selection of instances made sure model did not overfit the data. This process was repeated 10 times to make sure all the instances were covered for testing. Figure 7.4 shows the comparison of the classification accuracy across various scenarios and classifiers. The classification accuracy for the initial set of 64 features is better for all the classifiers as compared to other scenarios. This behavior is the result of having independent users models using different features for modeling. Hence the models were user specific and feature driven. In other scenarios, we chose a constant set of features using the subset selection algorithm over all the users. This process helped us to select unique and repeatable features that can be used to classify intruders in a generic case.

After selecting common features by subset selection algorithm, the models were generated and classification results are shown in Table 7.8 and Figure 7.4. The results show a reduction in accuracy results when compared with previous results. The reason was discussed in the previous paragraph of this section.

In case of limited intruder training, the drop in the accuracy was expected. The user model misclassified the negative instances as the model was not trained enough on variances of negative instances.

7.4.1 Evaluation of different classifiers

The intruder detection classification was done using 5 classifiers. The prime reason for having a variety of classifiers was to make sure the models are generic and don't overfit the data. The classifiers were chosen from various categories of machine learning algorithms. The first classifier we chose was Naive Bayes. The foundation of this classifier is based on conditional probability. We wanted to have a system that uses probabilistic approach learning while modeling the intrusion detection system. The second classifier was chosen from the function based machine learning algorithm. We chose Multilayer Perceptron for its error propagation and sigmoid function. The Support Vector Machine was not selected due to its complexity in modeling the non-linear systems. Finally, we selected Random Forest, Random Tree, and C4.5 (J48) classifier from decision tree based machine learning algorithms because data from real life are never divided into pure classes, we need a modeling mechanism to handle the impurity of class division and decision trees are one of the best algorithms to address these issues.

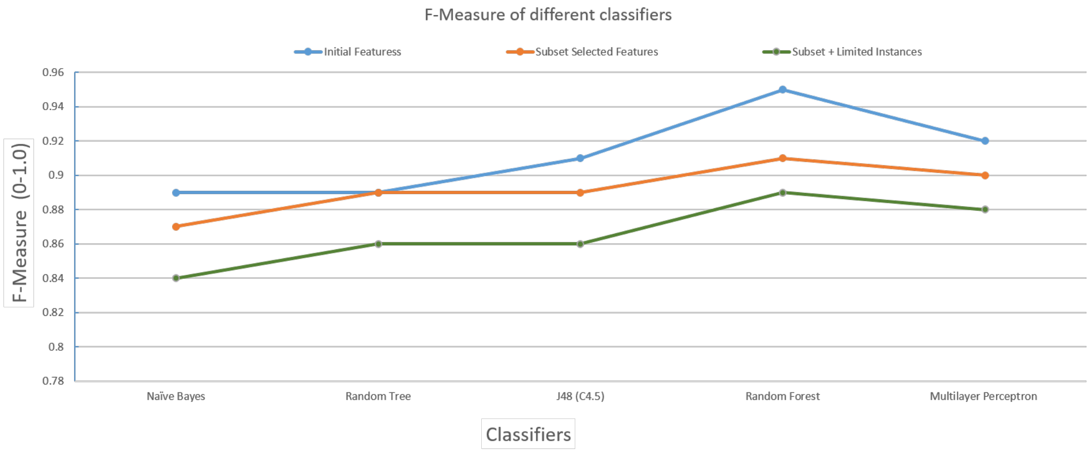


Figure 7.5: F-measure of different classifiers for three scenarios.

Figure 7.5 shows the F-measure of 5 classifiers we used to model the system. We saw that when using all features, classifiers are performing better as compared to other scenarios. Moreover, an interesting observation revealed that Random Tree performed the same in case of when using 64 features and subset of features. The reason for this behavior is that decision tree based algorithms choose features that make the decision about the class easy and doesn't consider the feature that introduces impurity in classification.

As we can see from Figure 7.4 and 7.5 that among different decision tree based algorithms the Random Forest performs best. Random Forest prunes the decision tree to reduce the complexity of the final classifier, and hence improves predictive accuracy by the reduction of overfitting.

Moreover, the Multilayer Perceptron performs on par with that of the decision tree based algorithms because of back propagation of error to update the weights of nodes. Neural networks (Multilayer Perceptron) are known for overfitting to the data so we used regularization built in WEKA Data Mining Toolkit to avoid it. Even though we used regularization, we still have the error propagation that makes the model more dependent on data.

7.4.2 Misclassification

In our system, classifiers made two types of mistakes. One, an intruder was misclassified as a genuine user. Other, a genuine user was classified as an intruder. All the cases that resulted in the misclassification of a genuine user as an intruder were the result of fatigue to the user while participating in the user study. All users provided thirty samples of positive instances i.e. using their password for authentication. The repetition of doing the same authentication resulted in fatigue and jitter while following the colored circles/shapes. Moreover, the saccades got longer, the number of fixations changed. Since the possibility of giving continuous samples will not be the regular case while using an authentication system, the number of false negatives will decrease. Moreover, the continuous exposure to the screen resulted in pupil dimension changes.

For the case of false positives, which is a serious affair as compared to false negatives in an authentication system, our system had around 8% of instances overall coming from selected users who were new to the eye tracking and couldn't follow the eye tracking protocol. We have user 5 and user 6, who found it hard to concentrate while performing the studies. This resulted in a decrease of accuracy as reflected in Table 7.1. Moreover, the experienced users who have performed the eye tracking based user study had fewer false positives. Finally, the clustering results can be improved by using the non-linear distance based measure to put the data into clusters. The manifold distance can be used as distance measure in clustering algorithms. The clustering algorithms with manifold distance has been used primarily for non-linear based data.

7.4.3 Usability of the Authentication Interface and Intruder Detection

All the users who participated in the user studies for intruder detection were given the survey to give feedback about our system and were asked questions about authentication systems as shown in figs. 7.6 to 7.13.

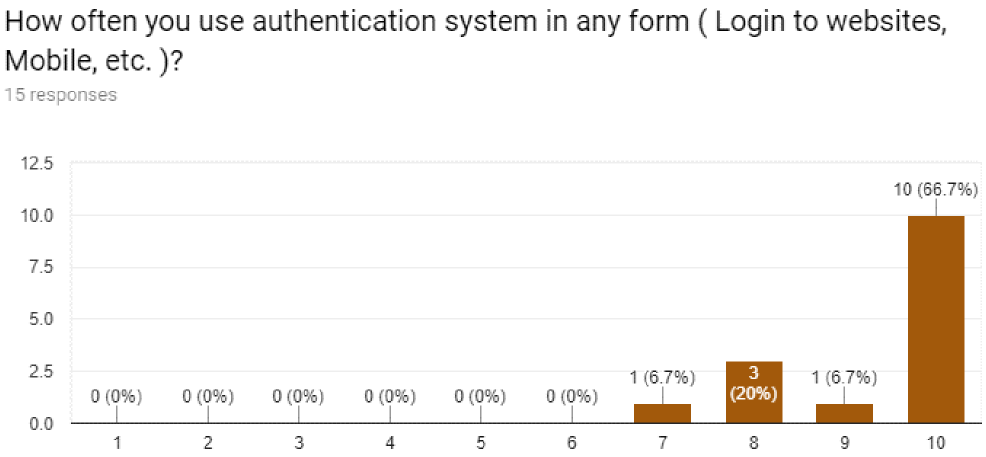


Figure 7.6: Figure showing how often people use authentication systems. (1-Never, 10-Everyday)

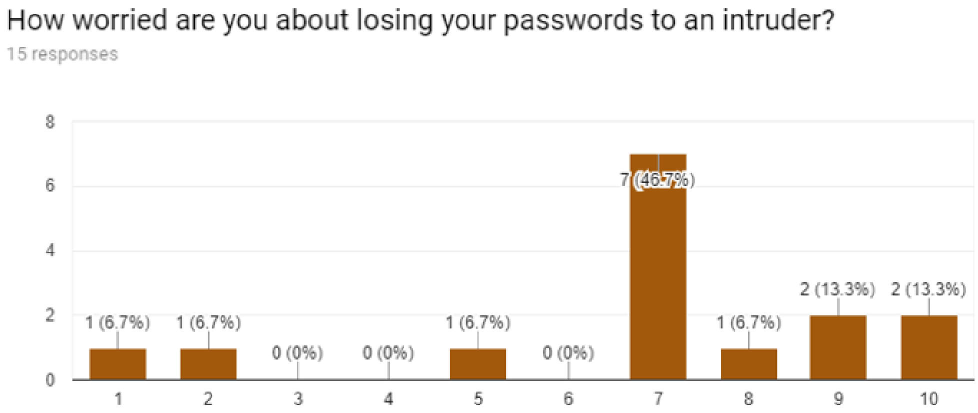


Figure 7.7: Chart showing the distribution of how serious the people are about losing their passwords. (1-Don't Care, 10-Freak Out)

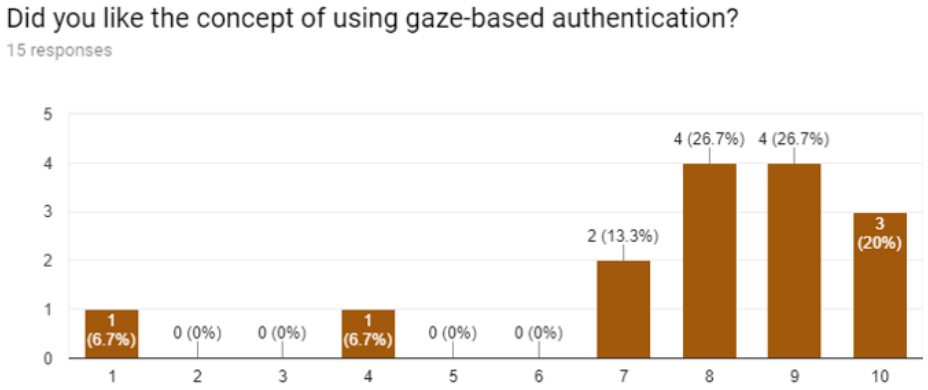


Figure 7.8: Chart showing the distribution of how participants liked the concept of using the gaze based authentication. (1-Didn't like it, 10-Loved it)

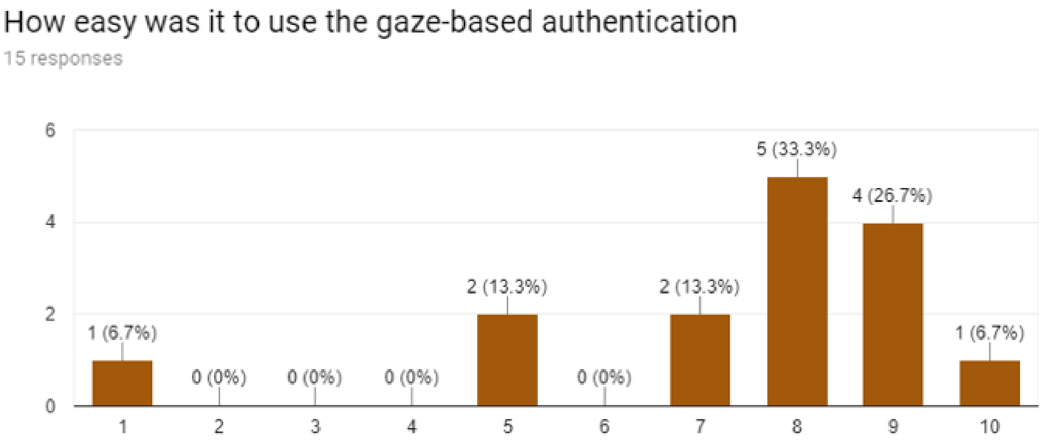


Figure 7.9: Ease of use of gaze-based authentication system. (1-Didn't like it, 10-Loved it)

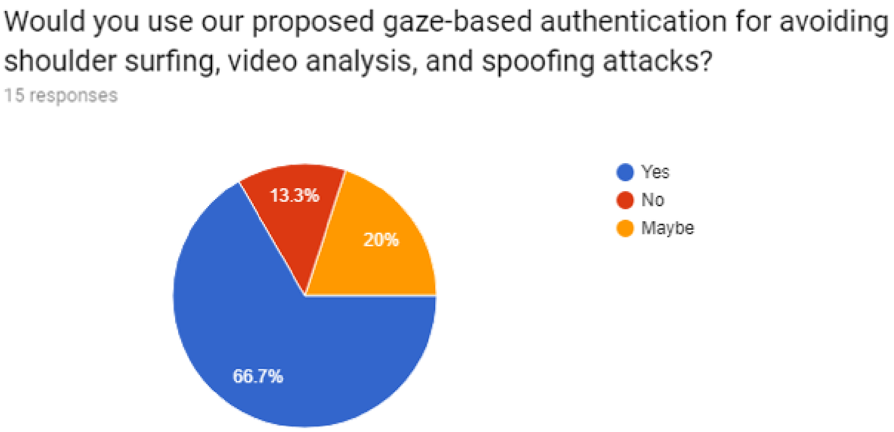


Figure 7.10: Ease of use of gaze-based authentication system. (1-Didn't like it, 10-Loved it)

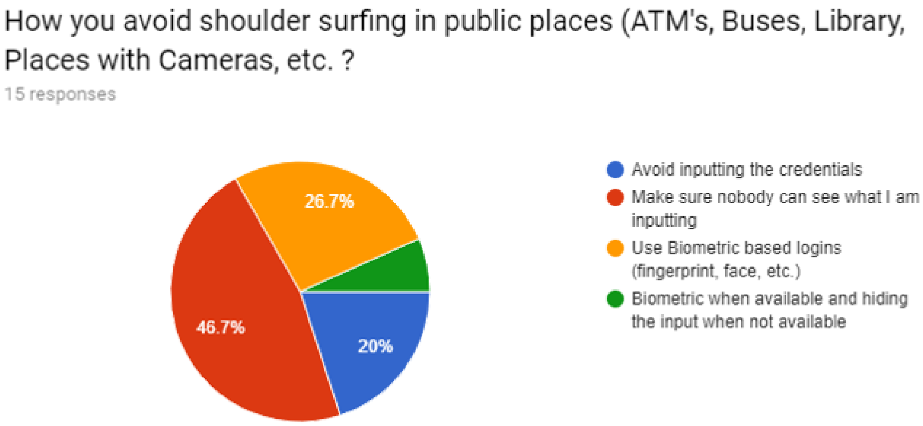


Figure 7.11: How would you avoid the shoulder surfing attacks in public places.

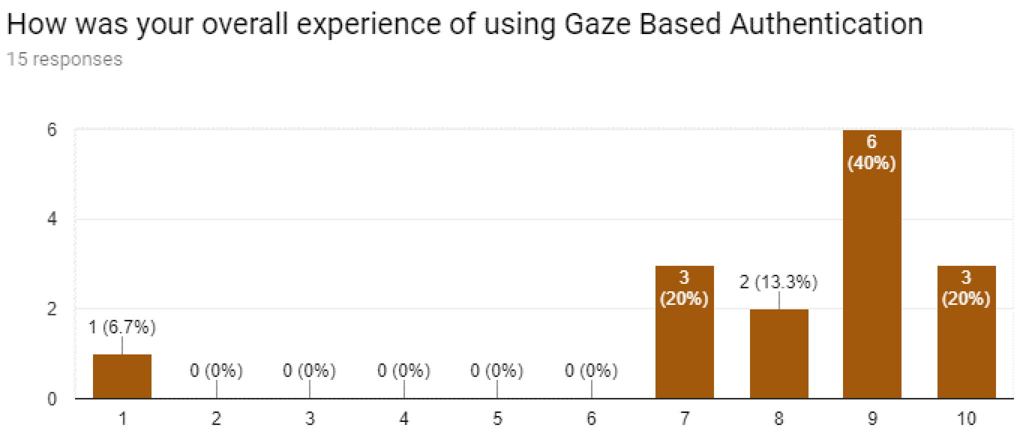


Figure 7.12: Overall experience of the participants with gaze based authentication system.

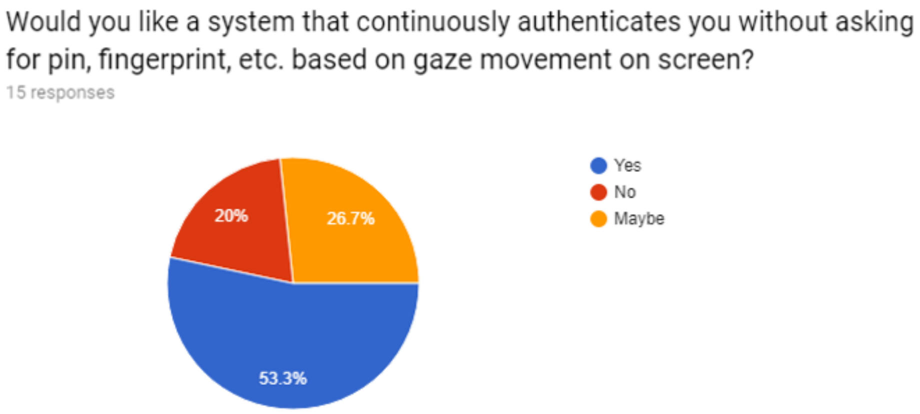


Figure 7.13: Prospects of having continuous authentication systems based on gaze.

8. FUTURE WORK

Our work suggests that the gaze-based authentication is a possible solution to handle shoulder surfing, video analysis attacks, and intruder detection. Moreover, we can explore the possibility of using the gaze-based authentication as a full biometric system. Furthermore, we believe that gaze-based authentication system can be made available to the mobile as well as small PDA's.

8.1 Generalized Biometric System

With the advancement of technology and hardware, it is possible to research on an avenue of making gaze-based authentication as general biometric authentication. High-frequency data coming from the gaze trackers can help us get more minute information of the gaze data and other features of eye movements like microsaccades, jitters, etc. Also, the replication/spoofing of muscle movement of eyes is very hard and nearly impossible. Furthermore, the pupillary light reflex to the light can't be modeled easily hence making gaze-based authentication resilient to the spoofing attacks. Finally, the gaze-based authentication can be used as a separate biometric-based authentication system without having an animation interface. A window of eye movement can be captured and classified as if it belongs to an individual or not. But it would need a large testing and detailed research.

8.2 Authentication System: Real Time

While our work evaluated the intruder detection using features and collected data, we believe that the system can be implemented for real-time recognition. The unique problem is the latency and resources for computation. The amount of the data coming from the gaze tracker is huge that is why we downsample the data for template matching algorithm to detect the colored circle/shape followed by the user. For doing the real-time intruder detection, we need to process the data to get all the features and run the recognition algorithm, which needs more research and improvement. Furthermore, the authentication latency needs to be decreased. More simplified yet secure interfaces can help in overcoming this problem.

8.3 Continuous Authentication Systems

Continuous authentication is a buzz word these days and industries are spending a lot on finding new ways to make sure the user is continuously authenticated while using the system. Gaze-based authentication is a non-intrusive way of continuous authentication. While a user is using a system, they need to look at it. This way the continuous authentication is non-intrusive and natural. Whenever a user looks away the system can be locked and needs to get authenticated while using the system.

9. CONCLUSION

In this work, we developed an authentication system based on gaze tracking. We were able to authenticate users with an accuracy of 97.5% and F-measure of 0.97. Moreover, we also developed a system to classify an intruder from that of a genuine user with an accuracy of 89.5% and F-measure of 0.89.

There are three main contributions of this research work. First, developing a framework for making the gaze-based authentication system resilient to the shoulder surfing, and video analysis attacks. This was achieved by using the randomization framework for initial placement and movement of colored circles/shapes on the interface.

Second, a non-intrusive authentication interface that is clutter free and simplified in contrast to other gaze-based authentication was developed. Moreover, the animation of the colored shapes and path generation was made generic to include the transition in different types of paths, like straight lines, curves, etc. We made the system generic so it can be updated to handle the changes with different applications. Also, we provided a solution to select the security level of gaze-based authentication. The more secure system will lead to increase in latency.

Finally, an intruder detection system was developed, which used 24 features to classify an intruder from a genuine user. The features were tested and evaluated to make sure the features represent the uniqueness of the user. Although our system had several errors, we discussed their probable cause and way to handle them by using high-frequency gaze tracker and features which were not feasible using the present eye tracker. Through this research work, we demonstrated a solution to shoulder surfing, video analysis, and intrusion detection using the gaze-based authentication system. Through the advancement of the trackers and technology, the gaze-based authentication can be used as stand-alone biometric authentication system and presents a non-intrusive way to achieve continuous authentication.

REFERENCES

[1] M. Zorz, "Data breaches lead to surge of spoofing attacks - Help Net Security," February 2018. https://www.helpnetsecurity.com/2015/05/13/data-breaches-lead-to-surge-of-spoofing-attacks/.

[2] P. Johnson and S. Schuckers, "Evaluation of Presentation Attack Detection: An Example," *https://www.nist.gov/*, 2016.

[3] V. Rajanna, A. H. Malla, R. A. Bhagat, and T. Hammond, "Dygazepass: A gaze gesture-based dynamic authentication system to counter shoulder surfing and video analysis attacks," in *2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, pp. 1–8, Jan 2018.

[4] F. Alamudun, "Analysis of visuo-cognitive behavior in screening mammography," PhD Doctoral Dissertation, Texas A&M University (TAMU), College Station, TX, USA, May 2016. Advisor: Tracy Hammond, ISBN: ORCID id: 0000-0002-0803-4542, `http://hdl.handle.net/1969.1/157040`.

[5] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Systems Journal*, vol. 40, no. 3, pp. 614–634, 2001.

[6] R. Morris and K. Thompson, "Password security: A case history," *Communications of the ACM*, vol. 22, no. 11, pp. 594–597, 1979.

[7] M. L. Das, A. Saxena, and V. P. Gulati, "A dynamic id-based remote user authentication scheme," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 629–631, 2004.

[8] L. O'Gorman, "Comparing passwords, tokens, and biometrics for user authentication," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2021–2040, 2003.

[9] K. Bagchi and G. Udo, "An analysis of the growth of computer and internet security breaches," *Communications of the Association for Information Systems*, vol. 12, no. 1, p. 46,

2003.

[10] J. Long, *No tech hacking: A guide to social engineering, dumpster diving, and shoulder surfing*. Syngress, 2011.

[11] M. Eiband, M. Khamis, E. von Zezschwitz, H. Hussmann, and F. Alt, "Understanding shoulder surfing in the wild: Stories from users and observers," in *Proceedings of the 35th Annual ACM Conference on Human Factors in Computing Systems*, CHI '17, ACM, 2017.

[12] J. Wayman, A. Jain, D. Maltoni, and D. Maio, "An introduction to biometric authentication systems," *Biometric Systems*, pp. 1–20, 2005.

[13] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on circuits and systems for video technology*, vol. 14, no. 1, pp. 4–20, 2004.

[14] A. K. Jain, L. Hong, S. Pankanti, and R. Bolle, "An identity-authentication system using fingerprints," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1365–1388, 1997.

[15] S. Mathur, A. Vjay, J. Shah, S. Das, and A. Malla, "Methodology for partial fingerprint enrollment and authentication on mobile devices," in *2016 International Conference on Biometrics (ICB)*, pp. 1–8, June 2016.

[16] W. Zhao, R. Chellappa, and A. Krishnaswamy, "Discriminant analysis of principal components for face recognition," in *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pp. 336–341, IEEE, 1998.

[17] R. P. Wildes, "Iris recognition: an emerging biometric technology," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1348–1363, 1997.

[18] N. Miura, A. Nagasaka, and T. Miyatake, "Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification," *Machine Vision and Applications*, vol. 15, no. 4, pp. 194–203, 2004.

[19] D. Gafurov, K. Helkala, and T. Søndrol, "Biometric gait authentication using accelerometer sensor.," *JCP*, vol. 1, no. 7, pp. 51–59, 2006.

[20] S. M. Furnell, P. Dowland, H. Illingworth, and P. L. Reynolds, "Authentication and supervision: A survey of user attitudes," *Computers & Security*, vol. 19, no. 6, pp. 529–539, 2000.

[21] P. Institute, "Global Visual Hacking Experimental Study: Analysis," *http://multimedia.3m.com/mws/media/1254232O/global-visual-hacking-experiment-study-summary.pdf*, 2016.

[22] "Cyber Attacks Explained: Authentication Attacks | | Valency Networks Blog."

[23] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget, "Design and evaluation of a shoulder-surfing resistant graphical password scheme," in *Proceedings of the working conference on Advanced visual interfaces*, pp. 177–184, ACM, 2006.

[24] A. K. Jain and A. Ross, "Multibiometric systems," *Communications of the ACM*, vol. 47, no. 1, pp. 34–40, 2004.

[25] B. Biggio, Z. Akhtar, G. Fumera, G. L. Marcialis, and F. Roli, "Security evaluation of biometric authentication systems under real spoofing attacks," *IET biometrics*, vol. 1, no. 1, pp. 11–24, 2012.

[26] C. Roberts, "Biometric attack vectors and defences," *Computers & Security*, vol. 26, no. 1, pp. 14–25, 2007.

[27] M. K. Khan and J. Zhang, "Improving the security of a flexible biometrics remote user authentication scheme," *Computer Standards & Interfaces*, vol. 29, no. 1, pp. 82–85, 2007.

[28] A. K. Jain and K. Nandakumar, "Biometric authentication: System security and user privacy.," *IEEE Computer*, vol. 45, no. 11, pp. 87–92, 2012.

[29] M. Wu, R. C. Miller, and S. L. Garfinkel, "Do security toolbars actually prevent phishing attacks?," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 601–610, ACM, 2006.

[30] T. Holz, M. Engelberth, and F. Freiling, “Learning more about the underground economy: A case-study of keyloggers and dropzones,” in *European Symposium on Research in Computer Security*, pp. 1–18, Springer, 2009.

[31] J. G. Steiner, B. C. Neuman, and J. I. Schiller, “Kerberos: An authentication service for open network systems.,” in *Usenix Winter*, pp. 191–202, 1988.

[32] L. Ghiani, D. Yambay, V. Mura, S. Tocco, G. L. Marcialis, F. Roli, and S. Schuckcrs, “Livdet 2013 fingerprint liveness detection competition 2013,” in *Biometrics (ICB), 2013 International Conference on*, pp. 1–6, IEEE, 2013.

[33] D. Yambay, J. S. Doyle, K. W. Bowyer, A. Czajka, and S. Schuckers, “Livdet-iris 2013-iris liveness detection competition 2013,” in *Biometrics (IJCB), 2014 IEEE International Joint Conference on*, pp. 1–8, IEEE, 2014.

[34] G. L. Marcialis, A. Lewicke, B. Tan, P. Coli, D. Grimberg, A. Congiu, A. Tidu, F. Roli, and S. Schuckers, “First international fingerprint liveness detection competition-livdet 2009,” in *International Conference on Image Analysis and Processing*, pp. 12–23, Springer, 2009.

[35] A. T. Duchowski, “Eye tracking methodology,” *Theory and practice*, vol. 328, 2007.

[36] A. H. Malla and T. Hammond, “Eye tracking- single technology to handle multiple domains,” in *23rd International Conference on Intelligent User Interfaces*, IUI ’18, (New York, NY, USA), pp. 677–678, ACM, 2018.

[37] V. Rajanna, “Gaze typing through foot-operated wearable device,” in *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS ’16, (New York, NY, USA), pp. 345–346, ACM, 2016.

[38] V. Rajanna, S. Polsley, P. Taele, and T. Hammond, “A gaze gesture-based user authentication system to counter shoulder-surfing attacks,” in *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA ’17, (New York, NY, USA), pp. 1978–1986, ACM, 2017.

[39] V. Rajanna and T. Hammond, “Gaze-Assisted User Authentication to Counter Shoulder-surfing Attacks,” *ArXiv e-prints*, Mar. 2018.

[40] V. Rajanna and T. Hammond, “Gawschi: Gaze-augmented, wearable-supplemented computer-human interaction,” in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research and Applications*, ETRA ’16, (New York, NY, USA), pp. 233–236, ACM, 2016.

[41] P. Kaul, V. Rajanna, and T. Hammond, “Exploring users’ perceived activities in a sketch-based intelligent tutoring system through eye movement data,” in *Proceedings of the ACM Symposium on Applied Perception*, SAP ’16, (New York, NY, USA), pp. 134–134, ACM, 2016.

[42] V. D. Rajanna, “Gaze and foot input: Toward a rich and assistive interaction modality,” in *Companion Publication of the 21st International Conference on Intelligent User Interfaces*, IUI ’16 Companion, (New York, NY, USA), pp. 126–129, ACM, 2016.

[43] B. Bauman, R. Gunhouse, A. Jones, W. Da Silva, S. Sharar, V. Rajanna, J. Cherian, J. I. Koh, and T. Hammond, “Visualeyeze: A web-based solution for receiving feedback on artwork through eye tracking,” in *IUI 2018 Workshop on Web Intelligence and Interaction*, 2018.

[44] V. Rajanna and T. Hammond, “A Gaze-Assisted Multimodal Approach to Rich and Accessible Human-Computer Interaction,” *ArXiv e-prints*, Mar. 2018.

[45] J. P. Lewis, “Fast template matching,” in *Vision interface*, vol. 95, pp. 15–19, 1995.

[46] G. J. Simmons, “A survey of information authentication,” *Proceedings of the IEEE*, vol. 76, no. 5, pp. 603–620, 1988.

[47] C. J. Colson, “Verification and authentication systems and methods,” Aug. 7 2012. US Patent 8,239,677.

[48] N. Asghari-Kamrani and K. Asghari-Kamrani, “Direct authentication and authorization system and method for trusted network of financial institutions,” Oct. 28 2008. US Patent 7,444,676.

[49] J. French and J. Wilder, “System and method for authentication of network users,” Dec. 17 2002. US Patent 6,496,936.

[50] R. V. Yampolskiy and V. Govindaraju, “Behavioural biometrics: a survey and classification,” *International Journal of Biometrics*, vol. 1, no. 1, pp. 81–113, 2008.

[51] U. Varshney and R. Vetter, “Mobile commerce: framework, applications and networking support,” *Mobile networks and Applications*, vol. 7, no. 3, pp. 185–198, 2002.

[52] E. W. Ngai and A. Gunasekaran, “A review for mobile commerce research and applications,” *Decision support systems*, vol. 43, no. 1, pp. 3–15, 2007.

[53] N. Leavitt, “Mobile phones: the next frontier for hackers?,” *Computer*, vol. 38, no. 4, pp. 20–23, 2005.

[54] K. Siau, L. Ee-Peng, and Z. Shen, “Mobile commerce: promises, challenges, and research agenda,” *Journal of Database management*, vol. 12, no. 3, p. 4, 2001.

[55] A. H. Lashkari, S. Farmand, D. Zakaria, O. Bin, D. Saleh, *et al.*, “Shoulder surfing attack in graphical password authentication,” *arXiv preprint arXiv:0912.0951*, 2009.

[56] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd, “Reducing shoulder-surfing by using gaze-based password entry,” in *Proceedings of the 3rd Symposium on Usable Privacy and Security*, SOUPS ’07, (New York, NY, USA), pp. 13–19, ACM, 2007.

[57] R. Weiss and A. De Luca, “Passshapes: utilizing stroke based authentication to increase password memorability,” in *Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges*, pp. 383–392, ACM, 2008.

[58] A. De Luca, R. Weiss, and H. Drewes, “Evaluation of eye-gaze interaction methods for security enhanced pin-entry,” in *Proceedings of the 19th Australasian Conference on Computer-Human Interaction: Entertaining User Interfaces*, OZCHI ’07, (New York, NY, USA), pp. 199–202, ACM, 2007.

[59] A. De Luca, M. Denzel, and H. Hussmann, "Look into my eyes!: Can you guess my password?," in *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, (New York, NY, USA), pp. 7:1–7:12, ACM, 2009.

[60] A. De Luca, K. Hertzschuch, and H. Hussmann, "Colorpin: securing pin entry through indirect input," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1103–1106, ACM, 2010.

[61] T. Kuribara, B. Shizuki, and J. Tanaka, "Vibrainput: two-step pin entry system based on vibration and visual information," in *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pp. 2473–2478, ACM, 2014.

[62] N. H. Zakaria, D. Griffiths, S. Brostoff, and J. Yan, "Shoulder surfing defence for recall-based graphical passwords," in *Proceedings of the Seventh Symposium on Usable Privacy and Security*, p. 6, ACM, 2011.

[63] A. D. Rubin, I. Jermyn, A. Mayer, F. Monrose, and M. K. Reiter, "The design and analysis of graphical passwords," in *8th USENIX Security Symposium*, Citeseer, 1999.

[64] P. Dunphy and J. Yan, "Do background images improve draw a secret graphical passwords?," in *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 36–47, ACM, 2007.

[65] A. Bulling, F. Alt, and A. Schmidt, "Increasing the security of gaze-based cued-recall graphical passwords using saliency masks," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, (New York, NY, USA), pp. 3011–3020, ACM, 2012.

[66] V. Roth, K. Richter, and R. Freidinger, "A pin-entry method resilient against shoulder surfing," in *Proceedings of the 11th ACM Conference on Computer and Communications Security*, CCS '04, (New York, NY, USA), pp. 236–245, ACM, 2004.

[67] R. W. Szeto, N. Jain, R. Suresh, and P. Kwan, "System and method for source ip anti-spoofing security," Sept. 10 2013. US Patent 8,533,823.

[68] T. M. Olkin, J. C. Olkin, and J. Moreh, “System for detecting spoofed hyperlinks,” Dec. 2 2008. US Patent 7,461,257.

[69] A. da Silva Pinto, H. Pedrini, W. Schwartz, and A. Rocha, “Video-based face spoofing detection through visual rhythm analysis,” in *Graphics, Patterns and Images (SIBGRAPI), 2012 25th SIBGRAPI Conference on*, pp. 221–228, IEEE, 2012.

[70] N. Erdogmus and S. Marcel, “Spoofing in 2d face recognition with 3d masks and anti-spoofing with kinect,” in *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, pp. 1–6, IEEE, 2013.

[71] D. Gafurov, E. Snekkenes, and P. Bours, “Spoof attacks on gait authentication system,” *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 491–502, 2007.

[72] S. M. Bellovin and M. Merritt, “Limitations of the kerberos authentication system,” *ACM SIGCOMM Computer Communication Review*, vol. 20, no. 5, pp. 119–132, 1990.

[73] J. Galbally, *Anti-spoofing: Fingerprint Databases*, pp. 1–9. Boston, MA: Springer US, 2009.

[74] G. Pan, L. Sun, Z. Wu, and S. Lao, “Eyeblink-based anti-spoofing in face recognition from a generic webcamera,” in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[75] A. Anjos, I. Chingovska, and S. Marcel, *Anti-spoofing: Face Databases*, pp. 1–13. Boston, MA: Springer US, 2009.

[76] A. Abhyankar and S. Schuckers, “Fingerprint liveness detection using local ridge frequencies and multiresolution texture analysis techniques,” in *Image Processing, 2006 IEEE International Conference on*, pp. 321–324, IEEE, 2006.

[77] A. Russo, “System for and method of securing fingerprint biometric systems against fake-finger spoofing,” Mar. 17 2009. US Patent 7,505,613.

[78] J. Galbally, S. Marcel, and J. Fierrez, "Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition," *IEEE transactions on image processing*, vol. 23, no. 2, pp. 710–724, 2014.

[79] X. Jia, X. Yang, K. Cao, Y. Zang, N. Zhang, R. Dai, X. Zhu, and J. Tian, "Multi-scale local binary pattern with filters for spoof fingerprint detection," *Information Sciences*, vol. 268, pp. 91–102, 2014.

[80] Z. Wei, X. Qiu, Z. Sun, and T. Tan, "Counterfeit iris detection based on texture analysis," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pp. 1–4, IEEE, 2008.

[81] X. He, Y. Lu, and P. Shi, "A new fake iris detection method," in *International Conference on Biometrics*, pp. 1132–1139, Springer, 2009.

[82] R.-L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 696–706, 2002.

[83] E. Osuna, R. Freund, and F. Girosit, "Training support vector machines: an application to face detection," in *Computer vision and pattern recognition, 1997. Proceedings., 1997 IEEE computer society conference on*, pp. 130–136, IEEE, 1997.

[84] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5325–5334, 2015.

[85] A. Rattani, W. J. Scheirer, and A. Ross, "Open set fingerprint spoof detection across novel fabrication materials," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2447–2460, 2015.

[86] J. Yamagishi, T. H. Kinnunen, N. Evans, P. De Leon, and I. Trancoso, "Introduction to the issue on spoofing and countermeasures for automatic speaker verification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 585–587, 2017.

[87] I. Chingovska, N. Erdogmus, A. Anjos, and S. Marcel, “Face recognition systems under spoofing attacks,” in *Face Recognition Across the Imaging Spectrum*, pp. 165–194, Springer, 2016.

[88] Z. Akhtar, “Biometric spoofing and anti-spoofing,” *Developing Next-Generation Countermeasures for Homeland Security Threat Prevention*, p. 121, 2016.

[89] Z. Boulkenafet, J. Komulainen, and A. Hadid, “Face spoofing detection using colour texture analysis,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1818–1830, 2016.

[90] J. H. Aughey, M. V. Rohr, S. D. Swaine, and C. J. Vorst, “Gaze tracking system, eye-tracking assembly and an associated method of calibration,” Sept. 13 2005. US Patent 6,943,754.

[91] T. Ohno and N. Mukawa, “A free-head, simple calibration, gaze tracking system that enables gaze-based interaction,” in *Proceedings of the 2004 symposium on Eye tracking research & applications*, pp. 115–122, ACM, 2004.

[92] V. V. Tengshe, H. V. Tengshe, and V. G. Tengshe, “Gaze tracking system and method,” Oct. 28 2003. US Patent 6,637,883.

[93] D. H. Yoo, J. H. Kim, B. R. Lee, and M. J. Chung, “Non-contact eye gaze tracking system by mapping of corneal reflections,” in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pp. 101–106, IEEE, 2002.

[94] J. Daugman, “New methods in iris recognition,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 5, pp. 1167–1175, 2007.

[95] J. Daugman, “How iris recognition works,” in *The essential guide to image processing*, pp. 715–739, Elsevier, 2009.

[96] O. Yamaguchi, Y. Okamoto, and K. Fukui, “Gaze detection apparatus and its method as well as information display apparatus,” June 15 1999. US Patent 5,912,721.

[97] S. Cluckey, "ATM / Automated Teller Machine business news, research, more | ATM Marketplace," February 2018. https://www.atmmarketplace.com/.

[98] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[99] A. Leon-Garcia and A. Leon-Garcia, *Probability, statistics, and random processes for electrical engineering*. Pearson/Prentice Hall 3rd ed. Upper Saddle River, NJ, 2008.

[100] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 21, no. 5, pp. 476–480, 1999.

[101] L. Shao and H. Zhou, "Curve fitting with bezier cubics," *Graphical models and image processing*, vol. 58, no. 3, pp. 223–232, 1996.

[102] H. Akima, "A new method of interpolation and smooth curve fitting based on local procedures," *Journal of the ACM (JACM)*, vol. 17, no. 4, pp. 589–602, 1970.

[103] H. Chui and A. Rangarajan, "A new algorithm for non-rigid point matching," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, pp. 44–51, IEEE, 2000.

[104] J. O. Wobbrock, A. D. Wilson, and Y. Li, "Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes," in *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pp. 159–168, ACM, 2007.

[105] M. El-Abed, C. Charrier, and C. Rosenberger, "Evaluation of Biometric Systems," *Evaluation of Biometric Systems*, 2012.

[106] B. Martin, "Biometric Identification: Metrics & Models," *NIST*, 2010.

[107] C. Goutte and E. Gaussier, “A probabilistic interpretation of precision, recall and f-score, with implication for evaluation,” in *European Conference on Information Retrieval*, pp. 345–359, Springer, 2005.

[108] T. Inc., “Error Rates,” Feb 2018. http://www.technovelgy.com/ct/Technology-Article.asp?ArtNum=93.

[109] J. Cherian, V. Rajanna, D. Goldberg, and T. Hammond, “Did you remember to brush? : A noninvasive wearable approach to recognizing brushing teeth for elderly care,” in *11th EAI International Conference on Pervasive Computing Technologies for Healthcare*, ICDC, 2017.

[110] B. Paulson, P. Rajan, P. Davalos, R. Gutierrez-Osuna, and T. Hammond, “What!?! no rubine features?: Using geometric-based features to produce normalized confidence values for sketch recognition,” in *HCC Workshop: Sketch Tools for Diagramming (VL/HCC)*, (Herrsching am Ammersee, Germany), pp. 57–63, VL/HCC, 9 2008.

[111] T. Hammond and R. Davis, “Tahuti: A geometrical sketch recognition system for uml class diagrams,” in *Technical Report SS-02-08: Papers from the 2002 Association for the Advancement of Artificial Intelligence (AAAI) Spring Symposium on Sketch Understanding*, (Menlo Park, CA), AAAI, 7 2002. 8 pages.

[112] B. Paulson and T. Hammond, “Paleosketch: accurate primitive sketch recognition and beautification,” in *Proceedings of the 13th international conference on Intelligent user interfaces*, pp. 1–10, ACM, 2008.

[113] T. Hammond and B. Paulson, “Recognizing sketched multistroke primitives,” *ACM Trans. Interact. Intell. Syst.*, vol. 1, pp. 4:1–4:34, Oct. 2011.

[114] B. Paulson and T. Hammond, “A system for recognizing and beautifying low-level sketch shapes using ndde and dcr,” in *ACM Symposium on User Interface Software and Technology (UIST)*, (Newport Rhode Island), ACM, 10 2007. 2 pages.

[115] T. Hammond and R. Davis, “Creating the perception-based ladder sketch recognition language,” in *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, DIS ’10, (New York, NY, USA), pp. 141–150, ACM, 2010.

[116] T. A. Hammond and R. Davis, “Recognizing interspersed sketches quickly,” in *Proceedings of Graphics Interface 2009*, GI ’09, (Toronto, Ont., Canada, Canada), pp. 157–166, Canadian Information Processing Society, 2009.

[117] T. Hammond and R. Davis, “Ladder: A language to describe drawing, display, and editing in sketch recognition,” in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI’03, (San Francisco, CA, USA), pp. 461–467, Morgan Kaufmann Publishers Inc., 2003.

[118] T. Hammond and R. Davis, “Shady: A shape description debugger for use in sketch recognition,” in *AAAI Fall Symposium on Making Pen-Based Interaction Intelligent and Natural (AAAI)*, (Arlington, VA), AAAI, 10 2004. 7 pages.

[119] T. Hammond and R. Davis, “Automatically transforming symbolic shape descriptions for use in sketch recognition,” in *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI’04, pp. 450–456, AAAI Press, 2004.

[120] T. Hammond and R. Davis, “Ladder, a sketching language for user interface developers,” *Computers & Graphics*, vol. 29, no. 4, pp. 518–532, 2005.

[121] T. Hammond and R. Davis, “Interactive learning of structural shape descriptions from automatically generated near-miss examples,” in *Proceedings of the 11th International Conference on Intelligent User Interfaces*, IUI ’06, (New York, NY, USA), pp. 210–217, ACM, 2006.

[122] T. A. Hammond, *Ladder: A Perceptually-based Language to Simplify Sketch Recognition User Interface Development*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2007. AAI0818371.

[123] T. Hammond, “Enabling instructors to develop sketch recognition applications for the classroom,” in *Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE ’07. 37th Annual*, pp. S3J–11–S3J–16, Oct 2007.

[124] T. Hammond, “Simplifying sketch recognition UI development,” in *Grace Hopper Celebration of Women in Computing*, (Orlando, FL), GHC, 10 2007. 5 pages.

[125] D. Rubine, “Specifying gestures by example,” in *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’91, (New York, NY, USA), pp. 329–337, ACM, 1991.

[126] D. Rubine, *Specifying gestures by example*, vol. 25(4). ACM, 1991.

[127] A. C. Long Jr, J. A. Landay, L. A. Rowe, and J. Michiels, “Visual similarity of pen gestures,” in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 360–367, ACM, 2000.

[128] M.-P. Dubuisson and A. K. Jain, “A modified hausdorff distance for object matching,” in *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, vol. 1, pp. 566–568, IEEE, 1994.

[129] T. Hammond, *Sketch Recognition: Algorithms and Applications*. Cambridge University Press, 2017. draft from March 1, 2016, publication forthcoming.

[130] S. Haykin and N. Network, “A comprehensive foundation,” *Neural networks*, vol. 2, no. 2004, p. 41, 2004.

[131] A. Jain and D. Zongker, “Feature selection: Evaluation, application, and small sample performance,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 19, no. 2, pp. 153–158, 1997.

[132] P. Pudil, J. Novovičová, and J. Kittler, “Floating search methods in feature selection,” *Pattern recognition letters*, vol. 15, no. 11, pp. 1119–1125, 1994.

APPENDIX A

RUBINE AND LONG'S FEATURES EQUATIONS

$$f_1 = \cos(\alpha) = \frac{(x_2 - x_0)}{\sqrt{[(y_2 - y_0)^2 + (x_2 - x_0)^2]}} \tag{A.1}$$

$$f_2 = \sin(\alpha) = \frac{(y_2 - y_0)}{\sqrt{[(y_2 - y_0)^2 + (x_2 - x_0)^2]}} \tag{A.2}$$

$$f_3 = \sqrt{[(y_{max} - y_{min})^2 + (x_{max} - x_{min})^2]} \tag{A.3}$$

$$f_4 = \arctan \frac{\left[(y_{max} - y_{min})\right.}{\left.(x_{max} - x_{min})\right]} \tag{A.4}$$

$$f_5 = \sqrt{(x_{n-1} - x_0)^2 + (y_{n-1} - y_0)^2} \tag{A.5}$$

$$f_6 = \cos(\beta) = \frac{(x_{n-1} - x_0)}{f_5} \tag{A.6}$$

$$f_7 = \sin(\beta) = \frac{(y_{n-1} - x_0)}{f_5} \tag{A.7}$$

$$f_8 = \sum_{i=1}^{n-1} \sqrt{\Delta x_i^2 + \Delta y_i^2} \tag{A.8}$$

where $\Delta x_i = x_i - x_{i-1}$ *and* $\Delta y_i = y_i - y_{i-1}$.

$$\theta_i = \arctan\left(\frac{\Delta x_i \Delta y_{i-1} - \Delta x_{i-1} \Delta y_i}{\Delta x_i \Delta x_{i-1} + \Delta y_i \Delta y_{i-1}}\right) \tag{A.9}$$

$$f_9 \;=\; \sum_{i=1}^{n-2} \theta_i \tag{A.10}$$

$$f_{10} \;=\; \sum_{i=1}^{n-2} |\theta_i| \tag{A.11}$$

$$f_{11} \;=\; \sum_{i=1}^{n-2} |\theta_i|^2 \tag{A.12}$$

$$f_{12} \;=\; \max_{i=1}^{n-1} \left[\frac{\Delta x_i^2 + \Delta y_i^2}{\Delta t_i^2} \right] \tag{A.13}$$

$$f_{13} \;=\; t_{n-1} - t_0 \tag{A.14}$$

$$f_{14} \;=\; |45^\circ - \arctan\left[\frac{(y_{max} - y_{min})}{(x_{max} - x_{min})} \right]| \tag{A.15}$$

$$f_{15} \;=\; \sum_{i=1}^{n-2} \begin{cases} |\theta_i|, & \text{if } \theta_i < 19^\circ \\ 0, & \text{otherwise} \end{cases} \tag{A.16}$$

$$f_{17} \;=\; \frac{\sum_{i=1}^{n-1} \sqrt{\Delta x_i^2 + \Delta y_i^2}}{\sqrt{[(x_{n-1} - x_0)^2 + (y_{n-1} - y_0)^2]}} \tag{A.17}$$

$$f_{18} \;=\; \frac{\sum_{i=1}^{n-1} \sqrt{\Delta x_i^2 + \Delta y_i^2}}{\sqrt{[(y_{max} - y_{min})^2 + (x_{max} - x_{min})^2]}} \tag{A.18}$$

where $\Delta x_i \;=\; x_i - x_{i-1} \; and \; \Delta y_i \;=\; y_i - y_{i-1}$.

$$f_{19} \;=\; \frac{\sqrt{[(x_{n-1} - x_0)^2 + (y_{n-1} - y_0)^2]}}{\sqrt{[(y_{max} - y_{min})^2 + (x_{max} - x_{min})^2]}} \tag{A.19}$$

$$f_{22} = \frac{\sum_{i=1}^{n-2} \theta_i}{\sum_{i=1}^{n-2} |\theta_i|} \tag{A.20}$$

$$f_{23} = \log[\sum_{i=1}^{n-1} \sqrt{\Delta x_i^2 + \Delta y_i^2}] \tag{A.21}$$

where $\Delta x_i = x_i - x_{i-1}$ *and* $\Delta y_i = y_i - y_{i-1}$.

$$f_{24} = \log[|45^\circ - \arctan\left[\frac{(y_{max} - y_{min})}{(x_{max} - x_{min})}\right]|] \tag{A.22}$$

$$f_{27} = \frac{\max_{i=1}^{n-2} \frac{\Delta y_i}{\Delta x_i} - \min_{i=1}^{n-2} \frac{\Delta y_i}{\Delta x_i}}{\sum_{i=1}^{n-1} \sqrt{\Delta y_i^2 + \Delta x_i^2}} \tag{A.23}$$

where $\Delta x_i = x_i - x_{i-1}$ *and* $\Delta y_i = y_i - y_{i-1}$.

$$f_{28} = \frac{\max_{i=1}^{n-2} \frac{\Delta y_i}{\Delta x_i}}{\sum_{i=1}^{n-2} \left[\frac{\Delta y_i}{\Delta x_i}\right] / n - 2} \tag{A.24}$$

where $\Delta x_i = x_i - x_{i-1}$ *and* $\Delta y_i = y_i - y_{i-1}$.